MICROCOMPUTER-ASSISTED DIAGNOSIS

OF

INHERITED DISORDERS

OF

THE SKELETON

A thesis submitted for the degree of Doctor of Medicine of the

University of Cape Town

Francois van Greunen MB ChB FCP(SA) Dip Datametrics(SA)

1987

DECLARATION.

This thesis is submitted for the degree of Doctor of Medicine of the University of Cape Town. The study was performed by the author under supervision of Professor P Beighton of the Department of Human Genetics and Professor K MacGregor of the Department of Computer Science.

I, Francois van Greunen, hereby declare that the work on which this thesis is based is original (except where acknowledgements indicate otherwise) and that neither the whole work nor any part of it has been, is being, or is to be submitted for another degree in this or any other University.

ACKNOWLEDGEMENTS.

The author wishes to express his sincere thanks and appreciation to:

Professor Peter Beighton of the department of Human Genetics for his enthusiasm, assistance and guidance. I can say with certainty that, without his constant encouragement, this study would not have come to fruition.

Professor Ken MacGregor of the Computer Science department for help and direction with the data processing aspects of this thesis as well as his willingness to comment on the manuscript.

The colleagues in the department of Paediatrics of the Johannes Gutenberg University in Mainz, Federal Republic of Germany. Professor Jurgen Spranger was and will remain a great source of inspiration. His courtesy and attention during my two visits to Mainz and his valuable contributions by correspondence, are acknowledged with gratitude. Sincere thanks to Dr Hartmut Menger and Dr Ralf Huth for their cordial co-operation.

Dr Colin Wallis of the department of Human Genetics for clinical material provided during the testing phase of this project. Dr Ingrid Winship for her invaluable advice on the intricacies of thesis preparation.

The management of the Cape Technikon for generously providing research time. My thanks to Mr Tobie Crause, head of the department of computer data processing, for his constant interest and encouragement.

Financial support for the research facilities which formed the background for this project was provided by the Mauerberger Foundation, the University of Cape Town Staff Research Fund, the Harry Crossley Foundation, and the Medical Research Council. These contributions are gratefully acknowledged.

To my wife and children

ABSTRACT.

Several hundred inherited disorders of the skeleton have been delineated. Individually these conditions are rare, but as a group they cause much crippling and hardship.

Several factors, including the rarity and complexity of the manifestations of these conditions, as well as semantic overlap, impede the accurate diagnosis which is essential for effective treatment. In this regard, the adoption of microcomputers warrants evaluation as a high technology aid.

Microcomputers have developed tremendous capabilities during recent years. The state of the art has become such that a diagnostic aid facility on such a device has been demonstrated in various disciplines of medicine and may also be feasible in the area of inherited skeletal disorders. The study which forms the basis of this thesis, concerns the investigation of this feasibility and has led to the development of an effective working model which sets the basis for microcomputer-aided diagnosis.

The design features followed in this project are similar to those conventionally employed for "Expert systems" on mainframe computers. A comprehensive knowledge base consisting of over 200 skeletal disorders and 700 radiographic and clinical manifestations, has resulted. Furthermore, the application is capable of "learning", although inference as employed by the inference engines of real expert systems, is not employed. In this context learning implies that the knowledge base, with the passage of time, improves considerably when used by experts. Serendipitous findings in this regard are:

- 1) Considerable improvement of existing profile descriptions can occur without any increased demands on computer memory and storage space;
- 2) Growth of the knowledge base in the form of additional disease profiles can be effected with very modest inroads on memory and storage resources.

The computerized diagnostic aid which resulted from this thesis, has been demonstrated to be successful in both the Department of Human Genetics of the University of Cape Town and the Department of Paediatrics of the Johannes Guten-

berg University in Mainz. Evaluated both in terms of efficiency and utility, the system provides an enhancement to the specialist genetic diagnostician.

These achievements have been effected by means of a unique newly developed application of compressed bit-mapping, attained by writing the applicable programs in Turbo Pascal and 8086- assembler languages. Calculations indicate that much larger data bases may possibly be implemented on present-day microcomputers by means of the methods developed in this project.

TABLE OF CONTENTS

SECTION ONE.

INTRODUCTION.

CHAPTER 1.1
THE MICROCOMPUTER.

In 1972 a remarkable electronic device appeared on the scene in the Unites States of America, and a new term was coined which would soon become a household word all over the world. The Intel Corporation had introduced the 4004 microprocessor, a small electronic component capable of all the basic processing activities normally encountered in the processing units of the bigger mini and mainframe computers of the day (Huggins, 1979).

To understand the significance of this development, it is necessary to briefly review the situation that had prevailed before the advent of the new device. The electronic computer was, at that stage about twenty years old and had consisted until then of discrete electronic parts, initially in the form of vacuum tubes (valves), and later of transistors. The implementation of the latter was a great improvement over the inefficient thermionic heat-generating tubes. Although significantly smaller, the transistors, by the sheer numbers needed, still resulted in bulky machines which, as a consequence of the construction process, continued to manifest a high failure rate.

In the nineteen sixties, the process of integration of electronic circuitry had begun and it became possible to combine many discrete circuits in one tiny housing. Thus Small Scale Integration (SSI) packages were developed, each containing up to about ten electronic logic circuits (gates). Gradually more adventurous devices appeared giving rise to:

MSI (Medium Scale Integration) with 10 to 100 gates per package;

LSI (Large Scale Integration) with 100 to a few thousand gates;

VLSI (Very large scale Integration) with several thousand gates.

This new technology paved the way for the development of microprocessors. The latter may be described as electronic devices, each consisting of a multitude of electronic circuits capable of data processing, and which could all be accommodated in one confined housing or "chip", a term which has become a household word. From the process of electronic integration stems the term Integrated circuit (IC). The microprocessors form one subset of many different types of IC's.

Currently the microprocessors are in the VLSI range with the so- called 32-bit devices very much the vogue. The earliest pioneer processors like the 4004 had 4-bit capability. By this is meant that they could switch four binary digits (bits) on or off, providing sixteen different combinations of binary states. Each of these combinations could be used to act as a discrete in-

struction which would incite the processor to initiate a certain task such as, for example, adding two numbers or inverting a number. As the microprocessors developed in sophistication to the present 16 and even 32-bit capabilities, and with technological advances resulting in ever increasing speed of circuit switching, it became possible to construct small computers which had central processing units (CPU's) consisting of a single microprocessor accompanied by a few support IC's. This situation stands in stark contrast to the mini- and mainframe machines which operate with CPU's consisting of thousands of LSI and even VLSI devices.

1.1.1 Definition of a microcomputer.

A microcomputer may be defined as a computer, not necessarily small, whose CPU is based on a single microprocessor integrated circuit. A practical definition of a computer, in turn, is that of an "automatic, electronic, 'stored program', reprogrammable digital device for data processing" (Zorkoczy et al, 1975).

The effects of the microcomputer evolution can only be described as staggering. It is beyond the scope of this study to enumerate all the consequences of this revolutionary development but the expansion of computing power and the increase in computer literacy are particularly relevant.

1.1.2 Computing Power expands.

In the pre-micro era the privilege of computation was limited to a small specialised subset of society. With the advent of the new affordable small machine, computing facilities are reaching an expanding section of the population. Micros are appearing in laboratories, offices, schools and homes. This allows individuals and small groups to gain access to ever-improving processing possibi lities, unbounded by constraints such as limited CPU time, rigid time slots and exasperatingly long turnaround times that had characterised computation on larger machines in the past.

1.1.3 Computer Literacy increases.

Computer literacy is another hackneyed term that has resulted from the small computer explosion. It is, however, a fact that more and more people are becoming conversant with these micromachines and are applying them in their various job situations. Authors, for instance, find the word processing capabilities offered by affordable small micros a boon to their creative efforts. Mana gers employing microcomputers in the so called Management Informa tion Systems (MIS) environment, find them invaluable in the deci sion making process (McLeod, 1983). Small expert systems have become reality and are helping in such diverse fields of endeavour as mineral prospecting, medical diagnosis and molecular chemistry (Hayes-Roth et al, 1983).

The microprocessor is one of the greatest technological discove ries of the century. The consequent development of the microcom puter will eventually affect all walks of life. What role it will eventually play as a utility in the wide field of Medicine, cannot at present be foreseen.

CHAPTER 1.2
THE MICROCOMPUTER AND MEDICAL DIAGNOSIS.

1.2.1 Mainframe Applications.

Several medical applications on mainframe computers have met with success. Some examples will be given in this chapter.

1.2.1.1 MYCIN.

MYCIN is an expert system which was developed at Stanford Univer sity by Shortliffe in 1976 (Shortliffe, 1976; Hayes-Roth et al, 1983). It gives advice on the diagnosis and therapy of infectious diseases, the medical knowledge being represented in terms of pro duction rules involving certainty factors which help accommodate probabilistic reasoning. Its knowledge base consists of about 400 rules relating possible conditions to associated interpretations. MYCIN tests a rule's conditions against available data or requests data from the doctor. A team of experts tested and compared the performance of medical experts, interns and MYCIN. The latter's performance was judged as good as, or superior to that of all the others (Hayes-Roth et al 1983).

1.2.1.2 CADUCEUS.

Another interesting mainframe application in the medical diagnos tic field is CADUCEUS, developed at Carnegie-Mellon University in Pittsburg in 1977 under the guidance of Pople and coworkers (Brachman et al 1983). This program displays expert performance in about 85 percent of the internal medicine domain, and its knowledge base is one of the largest known. In 1982 the system contained about 100 000 associations and employed very sophisti cated strategies to distinguish multiple diseases (Hayes-Roth et al 1983).

1.2.2 Minicomputer Applications.

The mere magnitude of knowledge bases such as the above precludes their implementation on microcomputers. Nevertheless medical diagnostic programs had been tried on small computers before the advent of the micro. In 1972 a team in Leeds developed a program to diagnose acute abdominal pain (Horrocks et al 1972, De Dombal et al 1972). The system was small and slow, taking 30 seconds to 15 minutes (Horrocks et al 1972) to make a diagnosis from a small set of abdominal conditions but, nevertheless, was remarkable for the era. What was astonishing was the fact that the computing system's diagnostic accuracy (91.8%) was significantly higher than that of the most senior member of the clinical team (79.6%). In a follow-up study reported in 1974, the same team (De Dombal et al 1974) reported the following significant observations:

The clinicians' diagnostic performance improved markedly during the period of the trial;

The proportion of appendices which ruptured before operation fell from 36% to 4%;

The negative laparotomy rate dropped sharply.

After the trial had closed, however, the figures reverted back to the pretrial levels with the number of perforations and negative operations increasing again. This may be regarded as an indica tion that computers can be beneficial in a clinical environment.

1.2.3 Microcomputer Applications.

1.2.3.1 The program described in 1.2.2 and whose decision making was based on Bayesian probability theory, was subsequently adapted to execute on a Commodore CBM 8050 microcomputer. Whether this micro-application displayed the same prowess as the larger one, is not known.

1.2.3.2 BDIS.

A notable development in the microcomputer field is the micro BDIS (Birth Defects Information System) as developed by Buyse and Edwards from a mainframe environment for use on IBM-XT compatible microcomputers. Micro BDIS was written in the C programming language by Langer and Brown, using the Lattice C package under the MSDOS version 2.1 operating system (Edwards, 1985).

Micro BDIS is an interactive program accepting patient information from a microcomputer terminal. It presents the operator with tables of Signs, Symptoms and Test results (SSTs) from which selections can be made as pertaining to cases under investigation. Clinical Case Profiles (Criteria) are made up which in turn are compared with Birth Defect Models (Candidates) and a number of potential diagnoses are produced for differential consideration.

Written entirely in a high level language, the system as implemen ted on an Olivetti M24 (IBM-XT compatible) with a fixed disk, tended to be somewhat cumbersome and verbose and with a slow response time. The repertoire of defects, however, is remarkably comprehensive and the program is a valuable tool for storing unknown or undiagnosed profiles which may at any later stage be matched with "candidates".

Possible uses for the microcomputer in a medical environment are protean. Apart from its success as a word processor for medical documentation, it has also been used to host highly normalised relational databases (Van Greunen et al 1986). The aim of this thesis was to investigate the use of a microcomputer system as an adjuvant in the diagnosis of inherited disorders of the skeleton and, if possible, to develop a working model. This intention will be formally stated in the next section.

SECTION TWO.

THE PROBLEM STATEMENT

CHAPTER 2.1
AIM AND RATIONALE OF THE STUDY.

The aim of the study which forms the subject of this thesis was to design, develop, establish and test a microcomputer facility which could assist in the diagnosis of inherited skeletal disorders.

Several hundred such conditions of the skeleton have been delineated. These disorders, which are individually rare but collectively not uncommon, represent an important cause of crippling and physical handicap. It is widely accepted that, for effective clinical and genetic management, diagnostic accuracy of these conditions is crucial. In view of the large number of skeletal dysplasias and the complexity of their manifestations, some form of computer assisted diagnostic aid warrants evaluation.

As was indicated in section one, microcomputers have undergone rapid development over the last few years. The limitation imposed by restricted memory capabilities in the past, is being alleviated at a fast rate and the serious application of these devices in the field of assisted medical diagnosis, has become feasible.

The Medical Resarch Council (MRC) unit for inherited disorders of the skeleton has been established in the department of human genetics at the University of Cape Town. A microcomputer facility exists within the unit which, in conjunction with the author's own equipment, has been used to develop the schemas and programs for the implementation and evaluation of this project.

Implementation will be discussed in section 5. The important aspect of evaluation, which will test the microcomputer's performance when presented with the profiles of both known, diagnosed cases, and undiagnosed problem cases, will be elaborated in sec tion 7.

Chapter 2.2 The Computer Equipment.

2.2.1 The Hardware.

Three microcomputers have been employed during the development of this project:

2.2.1.1 Olivetti M24:

This apparatus, which is compatible with the IBM-XT type of micro computers, is situated in the department of human genetics. The system consists of a central processing unit (CPU) with 640 kilo bytes (Kb) of random access memory (RAM), one 360 Kb flexible disk drive and one 20 megabyte (Mb) Everex fixed disk drive. The operating system (OS) is the Micro-Soft Disk Operating System (MSDOS) version 2.11. An Epson LQ-100 printer provided hard copy.

2.2.1.2 Sperry IT Microcomputer:

This machine is IBM-AT compatible and consists of a CPU with 512 Kb of RAM, a further 512 Kb of extended RAM, one 1,2 Mb flexible disk drive, one 360 Kb flexible disk drive and one 44 Mb Miniscribe fixed hard disk. A partition of about 40% of this hard disk was allocated to the OS under which this project was developed, namely MSDOS version 3.1. A Fujitsu DPMG9 printer provided hard copy.

2.2.1.3 A Corona, IBM-XT compatible microcomputer with 512 Kb RAM, one 360 Kb flexible disk drive and one 10Mb Miniscribe hard disk drive, served as a standby.

2.2.2 The Utility Software.

2.2.2.1 Turbo Pascal.

The bulk of the programming for this project was undertaken using Turbo Pascal version 3.0 from Borland International. This soft ware tool was elected for the following reasons:

- i) The package is affordable and was available in the department of human genetics.
- ii) It is extremely fast and effective and is a very comprehensive implementation of the Pascal language. Comparative tests, performed by the author and which will not be detailed here, proved it to be significantly superior to the Microsoft C language compiler for MSDOS, GW Basic, True Basic, UCSD Pascal, Waterloo Pascal and Microsoft Pascal.
- iii) The package is very stable and no serious "bugs" were encountered.
- iv) Turbo Pascal is compact and can easily reside on customary microcomputer peripherals such as flexible disks and electronic (RAM) disks.
- v) It has in the meantime become a world standard.

2.2.2.2 The Macro-86 Assembler.

Version 1 of this Microsoft product was used to develop program modules where speed and efficiency were crucial. As the author is acquainted with 8086 assembler language programming, this was a logical choice for certain unique features of this project (see chapter 4.4).

CHAPTER 2.3
THE CHOICE OF DOMAIN.

Being associated with the department of human genetics under the direction of professor Peter Beighton, the field of skeletal dysplasias was a natural choice for this study. Additional considerations make this choice pre-eminent:

2.3.1 Well Demarcated Domain.

The skeletal dysplasias comprise a well demarcated domain for the application of computer-assisted diagnosis. As was noted earlier, several hundred of these conditions have been described. Yet, in comparison with genetic disease in its entirety or compared with the realm of internal medicine as a whole, the field is limited and becomes manageable on a microcomputer. Should it prove to be successful in this limited context, the approach could well be followed in more adventurous projects encompassing more extended environments. In view of the rapid technological development which is taking place, it is not difficult to envisage the evolution of larger and larger pseudo-expert systems on machines that are continually getting smaller in physical size. The subject of a pioneer study at this point in time should, however, preferably be of a closely circumscribed nature.

2.3.2 Ease of Representation.

The fact that many of the features of the skeletal diseases relevant to this study are often, although by no means always, of the "all or none" kind, ie. either present or not present. This characteristic lends itself to a binary representation in a computer - a feature that will become evident later.

2.3.3 Lexical Considerations.

A minor consideration is the fact that the nomenclature and jargon of the subject of skeletal dysplasia tend to be complex and seve rely tax the spelling prowess of any individual. This may bedevil search and retrieval strategies and present a challenge on a small computer where space for validating procedures is often wanting. However, if by following strategies of modern database technology, a knowledge base of high data integrity and low redundancy could be developed, the microcomputer could prove to be a valuable weapon in an area beset with lexical pitfalls and semantic overlap.

Finally it can be stated that the subject of inherited skeletal disorder is fascinating to a very high degree and a whet of inte rest. This is, however, tempered by the fact that, as is so often the case in medicine, interesting conditions tend to be associated with much suffering and hardship. Any contribution towards the alleviation of these afflictions, cannot be in vain.

SECTION THREE.

THE KNOWLEDGE BASE

CHAPTER 3.1
INTRODUCTION.

An Expert System (ES) is regarded by knowledge engineers to consist of two major components (Hayes-Roth et al, 1985):

3.1.1 The Knowledge Base.

The knowledge base is a repository for the expert knowledge acquired by the system.

3.1.2 The Inference Engine.

The inference engine has been described as a "gate-keeper" between the knowledge base and the program which uses it. It is in itself a program responsible for adding and deleting "beliefs", ie. facts or combinations of facts believed to be true at a particular time, as well as performing certain classes of inference (Charniak and McDermott, 1985).

The present project fulfills the responsibility of adding, changing and deleting "beliefs" to a knowledge base, but does not perform inference directly. Hence it cannot be said that a true inference engine is employed and the project does not qualify for the title of expert system as usually defined. Rather, it should be seen as an aid to the human diagnostician and not as a diagnostic machine per se. It has, however, become a significant repository of expert medical knowledge with a fairly extensive repertoire of beliefs. Hence the same stages which are valid for the implementation of a knowledge base for a true expert system as outlined by Buchanan et al (1985), were considered to be applicable and were consequently followed during the establishment of this knowledge base. With the preparation of the rest of this section, the paper by Buchanan et al (1985) was extensively consulted. Repeated reference to that source will therefore not be made.

CHAPTER 3.2
THE STAGES OF KNOWLEDGE ACQUISITION.

The following major stages of knowledge acquisition for an expert system are recognised:

- **Identification**
- **Conceptualization**
- **Formalization**
- **Implementation**
- **Testing**
- **Prototype Revision**

This approach was used with the implementation of the knowledge base of this project. The first two stages will be expounded in this chapter. The subsequent stages will each have a section devoted to them.

CHAPTER 3.3
THE IDENTIFICATION STAGE.

3.3.1 The Participant Identification and Roles:

This is regarded as the first responsibility of the Identification Stage. Usually knowledge acquisition, at any one time, involves interaction between a solitary expert and a single knowledge engineer. The expert acts as a source of information and tells the knowledge engineer about his knowledge and expertise. The knowledge engineer receives the knowledge and restates what has been understood of the problem and introduces the information to the computer system.

•

3.3.1.1 The Domain Experts.

The author, and indeed this project, were fortunate to have access to the expertise of two scientists of world reknown in the field of skeletal dysplasia, who agreed to act as domain experts for this knowledge base, namely prof. Peter Beighton of the Department of Human Genetics of the University of Cape Town and prof Jurgen Spranger of the Department of Paediatrics of the Johannes Gutenberg University in Mainz, Federal Republic of Germany.

3.3.1.2 The Knowledge Engineer.

The author, who has developed a keen interest in microcomputer applications over the years since 1977, acted as knowledge engineer for this project. In this capacity he was priviledged to obtain advice and guidance from prof. K J MacGregor of the Department of Computer Science of the University of Cape Town.

3.3.2 Problem Identification.

This involves an exchange of views on various aspects of the problem between the experts and the knowledge engineer. Such discussions revealed that a satisfactory computer system to facilitate the accurate diagnosis of the skeletal dysplasias had not yet been developed and that mainframe implementations, current at the time, were not entirely satisfactory.

Some of the problems that begged answering may be identified as follows:

3.3.2.1 Consistent Differential Diagnosis.

Given a set of clinical and radiological features from a domain replete with complexities, what are all the possible diagnoses? Even the most astute of experts cannot access all their knowledge at any one time. Indeed, it is rare for an individual to produce a complete differential diagnosis when confronted with a complex mix of manifestations. The computer, on the other hand, should be capable of always presenting a full list from its current "be liefs" that will fit the bill with a given unknown profile.

To resolve a problem of this kind with, say, the aid of textbooks, will prove to be a complex task. Most textbooks are arranged according to a disease-related index and not to manifestations. To obtain a list of conditions in which "wormian bones", for instance, are present, would

require that the text be scanned from beginning to end and every condition scrutinized for the presence of the said manifestation. In this way a comprehensive differential diagnosis will probably never be obtained. While it is true that the computer cannot be guaranteed to provide a fully comprehensive list, its beliefs may be added to, should these be incomplete, at any time, a capability of which books are clearly devoid. In addition it should also be possible to modify the set of beliefs held by a computer system at will, should there be indication to do so. In the case of a book this can only be done at the time of a new edition.

3.3.2.2 A further class of problems may be stated:

Having reached a final diagnosis in a particular case, are there any other diagnostic features pertaining to the disease which the examiner may be unaware of, but which may be important to look for? Alternatively, does the patient in question show features which, unknown to the examiner, are not part of the diagnosed condition or, when indeed part of the condition, have not yet been documented?

An extensive literature scan, albeit cumbersome, can resolve the first of these subproblems. A knowledge base with high integrity should, theoretically at least, be much more rapid and efficient.

The second subproblem will be even more difficult to resolve with conventional methods. The electronic device, on the other hand, should again prove to be faster and more efficient.

A problem can therefore be identified which this project will attempt to resolve:

Can a microcomputer system be developed to host a respectable knowledge base of inherited skeletal disorders and their stigmata in a compact way, enabling rapid retrieval and searches as well as aiding in the establishment of accurate diagnoses and meaningful differential diagnoses?

3.3.3 Resource Identification.

The domain experts and the knowledge engineer must use various sources to obtain knowledge relevant to the establishment of a knowledge base (Buchanan et al, 1983). For the experts this will, in general, consist of their own problem-solving experiences in the past, textbooks dealing with subjects in the domain and examples of problems and solutions . To this may be added the professional press where profiles pertaining to the discipline as well as the experiences of others can be obtained.

3.3.3.1 Expert Knowledge Resources of this Project.

The following sources of knowledge provided the majority of the information for inclusion in the data base:

- **i) The Gamut Index of Skeletal Dysplasias (Kozlowski and Beighton, 1984).**

The complete technique of knowledge acquisition will be outlined under the Implementation stage. Suffice it to state here that this work provided an important core of basic information for the

knowledge base, which could be amplified from other sources. At the same time it supplied a rich source of references which could be consulted in order to augment the knowledge base and to resolve the few ambiguities that were encountered in the Gamut Index itself.

- **ii) Atlas of Skeletal Dysplasias (Spranger, Langer and Wiedemann, 1974).**

This work supplied valuable descriptions of skeletal dysplasias known up to 1974, and was extensively consulted.

- **iii) The Professional Literature.**

The importance of the literature as a source of information will become apparent when the implementation stage is discussed. Many of the profiles of conditions mentioned in the Gamut Index could be amplified by descriptions of the disease conditions as given in the various journals. It also proved to be a valuable source of radiological and clinical features for inclusion in the relevant part of the knowledge base.

3.3.3.2 Knowledge Engineer Resources.

While reading for the Diploma of Datametrics at the University of South Africa, the author acquainted himself with methods that could be fruitfully applied to the development of a system of this nature. The methods and structures will be discussed fully under the Implementation stage. In summary, it can be stated here that knowledge of the following methods and software structures proved to be valuable in the development of this project:

- **i) Modular Programming using the Pascal language.**

The modular programming approach made it possible to subdivide the relevant programs into managable modules and often-used routines could be included in software libraries. The Turbo Pascal language lends itself admirably to modular programming, a characteristic which facilitated the structuring and writing of the application programs.

- **ii) The record structures provided by the Pascal language.**

Pascal record types are very suitable for this kind of knowledge base. Every disease condition can be represented in a record which can be retrieved as an entity for processing. A random file-access facility is available which allows for rapid retrieval and searching of data items.

3.3.3.3 Assembly Language.

While assembler programming tends to be laborious and unproductive, programs produced by this means are extremely fast and efficient. It was possible to write some of the critical modules using this mode, thereby achieving remarkable speed in the search, comparison and retrieval procedures. Turbo Pascal has a very efficient interface with assembly language routines, a facility which proved of great value with the optimisation of the application programs.

3.3.3.4 The Technique of Bitmapping.

With this technique, which will be discussed in detail in section 4, the presence or absence of diagnostic features could be

represented in a binary fashion. Expanding on this facility, the author was able to develop a new unique method of compressing bit maps and thereby facilitating the storage of extensive quantities of information in a significantly confined memory space.

3.3.4 Goal Identification.

The problem of diagnosing the large number of skeletal dysplasias as a result of their complexity and individual rarity, was stated in section 2. Buchanan et al (1983) are, however, of the opinion that "it is helpful to separate the goals from the specific tasks of the problem." In addition to attempting to resolve the problem as formulated for inherited diseases of the skeleton, a goal could be envisaged: to provide a model for emulation by other spheres of medicine and similar domains. In section 8 this aspect will be enlarged upon.

CHAPTER 3.4
THE CONCEPTUALIZATION STAGE.

The key concepts and relations identified in the first stage are made explicit during the conceptualization stage. This stage in volves repeated interaction between the knowledge engineer and the domain experts.

In the present study a heuristic approach was called for at this stage. Initially the problem was approached along the same lines as the Gamut Index whereby applicable selections are made from a sequence of lists until a limited number of conditions remained as a differential diagnosis. Section one of the Gamut Index was thus computerised. Theoretically the microcomputer gamut index (MGI) did not offer any significant improvement compared with the book version; nevertheless, a number of practical benefits did come to light.

3.4.1 The Microcomputerized Gamut Index (MGI).

Some aspects of the MGI are presented (for convenience in this section) as a lesser feature of the project and for the following reasons:

- i) Although not as extensive as the main project, it does represent a considerable investment of time and programming effort.
- ii) It serves to illustrate the fact that the process of conceptualization is intricate and taxing, involving considerable heuristic effort, and perhaps never reaching finality.
- iii) The MGI is, by its own right, a microcomputer aid to the diagnosis of skeletal dysplasia that has been found to be useful in practice.

3.4.2 Benefits of the MGI.

As previously stated, some advantages over the book version of the Gamut Index became evident with use of the MGI:

- **i) Ease of Updating.**

The most significant advantage of the MGI is the ease with which information can be edited on a day to day basis. It is a trivial task to keep the MGI up to date, whereas the book version can never be kept current. Any significant changes to the latter can only be made at the time of new editions.

- **ii) Immediate Availability.**

The microcomputer will eventually become commonplace on the average desktop. Being a multipurpose device used for many diverse tasks such as spreadsheet manipulation and word processing, it should and most probably will constantly be ready for immediate use. Under these circumstances an application program such as the MGI could be used immediately and without effort, unlike a book which often is not at hand when needed.

- **iii) Speed Advantage.**

A minor but significant factor is the speed at which the MGI can be traversed as compared to the book version.

3.4.3 An Illustration of the MGI.

Figures 3-1 to 3-4 are examples produced by the microcomputer of the menus employed by the Gamut Index. In fig 3-1 the section of generalized skeletal abnormalities is selected by entering the number 1. This causes the menu depicted in fig 3-2 to follow. As an example, "Skeletal Dysplasia of the Newborn" is selected from this list, by entering the number 12. This results in the appearance of the menu shown in fig 3-3. Selecting item 1.12.2 by entering a 2, produces the final differential diagnosis illustrated in fig 3-4, from which a possible diagnosis may be selected. The facility ends at this point, but a future development could be to link this list with the database of disease profiles of the main project. In this way a means of obtaining disease descriptions can be established. This point will be discussed further in section 8.

Figures 3-5 to 3-7 illustrate another example with different selections. These are 1.4, "Periosteal Thickening and Periostitis", producing fig 3-6, and 1.4.1, "Bone Dysplasias", which, in turn, reveals the final list shown in fig 3-7.

As it is a less important part of the present project, the finer details of the MGI will not be expounded. A listing of the main Gamut program, "Gamshow.Pas" appears in Appendix F.

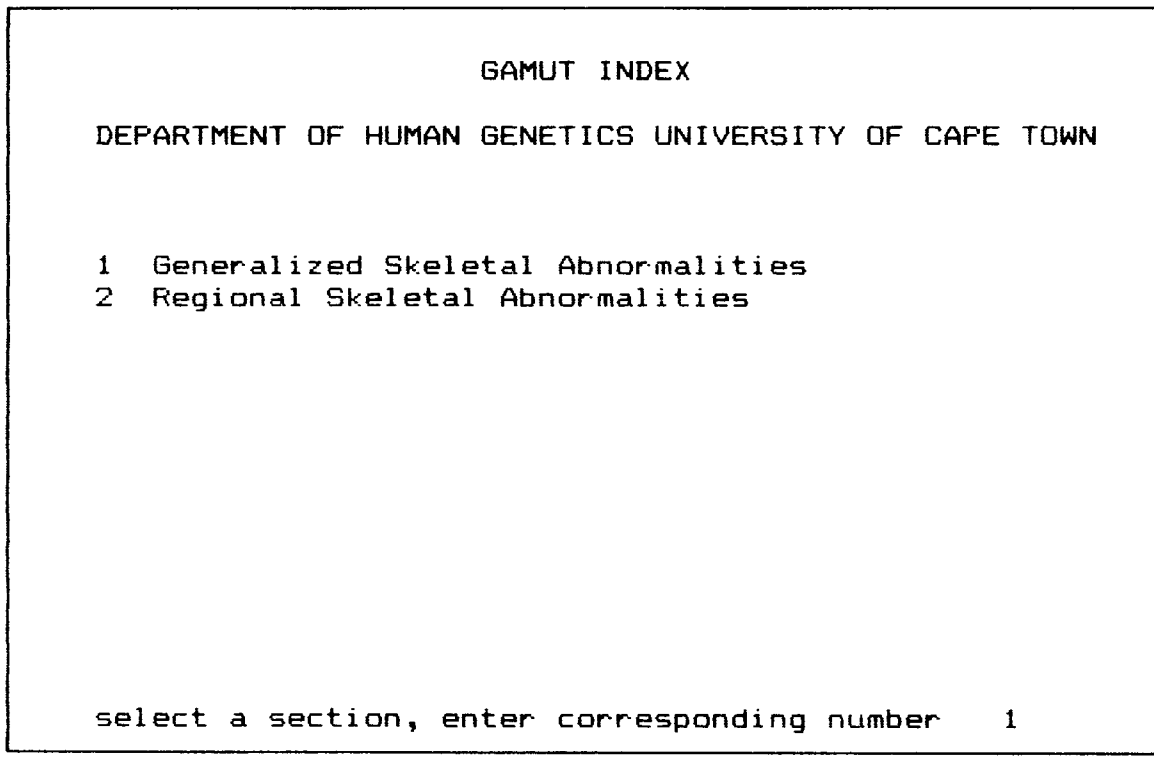

GAMUT INDEX
DEPARTMENT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN
1 Generalized Skeletal Abnormalities
2 Regional Skeletal Abnormalities
select a section, enter corresponding number 1

Fig. 3 - 1

Section 1 Generalized Skeletal Abnormalities

1.1 Osteoporosis
1.2 Multiple Fractures
1.3 Osteosclerosis and Hyperostosis
1.4 Periosteal Thickening and Periostitis
1.5 Exostoses
1.6 Multiple Radiolucent Defects
1.7 Advanced and Retarded Bone Age
1.8 Complex Generalized Abnormalities
1.9 Asymmetry
1.10 Multiple Dislocations
1.11 Soft Tissue Calcification
1.12 Skeletal Dysplasia in the Newborn

Select number, 0 to exit, B to go backward 12

Fig. 3 - 2

Subsection 1.12 Skeletal Dysplasia in the Newborn

1.12.1 Lethal
1.12.2 Potentially Lethal
1.12.3 Viable

Select number, 0 to exit, B to go backward 2

Fig. 3 - 3

Subsection 1.12.2 Potentially Lethal

1.12.2.1 Chondrodysplasia Punctata - Rhizomelic Form
1.12.2.2 Camptomelic Dysplasia
1.12.2.3 Asphyxiating Thoracic Dysplasia
1.12.2.4 Osteogenesis Imperfecta Type II
1.12.2.5 Osteopetrosis with Precocious Manifestations
1.12.2.6 Dyssegmental Dysplasia

Select number, 0 to exit, B to go backward

Fig. 3 - 4

Section 1 Generalized Skeletal Abnormalities

1.1 Osteoporosis
1.2 Multiple Fractures
1.3 Osteosclerosis and Hyperostosis
1.4 Periosteal Thickening and Periostitis
1.5 Exostoses
1.6 Multiple Radiolucent Defects
1.7 Advanced and Retarded Bone Age
1.8 Complex Generalized Abnormalities
1.9 Asymmetry
1.10 Multiple Dislocations
1.11 Soft Tissue Calcification
1.12 Skeletal Dysplasia in the Newborn

Select number, 0 to exit, B to go backward 4

Fig. 3 - 5

Subsection 1.4 Periosteal Thickening and Periostitis

1.4.1 Bone Dysplasias
1.4.2 Other Bone Diseases
1.4.3 Traumatic
1.4.4 Infection
1.4.5 Hypo/Hypervitaminosis
1.4.6 Skin Disorders
1.4.7 Tumors
1.4.8 Metabolic Disorders
1.4.9 Miscellaneous

Select number, 0 to exit, B to go backward 1

Fig. 3 - 6

Subsection 1.4.1 Bone Dysplasias

1.4.1.1 Idiopathic Hypertrophic Osteoarthropathy
1.4.1.2 Familial Idiopathic Hypertrophic Osteoarthropathy
1.4.1.3 Benign Osteosclerosis of Infancy
1.4.1.4 Engelmann Disease
1.4.1.5 Neurofibromatosis (subperiosteal hemorrhages)
1.4.1.6 Fibrous Dysplasia (McCune-Albright Type)

Select number, 0 to exit, B to go backward

Fig. 3 - 7

3.4.4 The Concept for the present Study.

After several interactions between the author, acting as knowledge engineer, and the domain experts, it was decided to follow an approach different from the gamut index. The following scheme was conceived:

3.4.4.1 Establish a Knowledge Base.

This was to consist of two major components:

- **i) A Database of Radiological and Clinical Features.**

Using the sources of information enumerated previously, a pool of manifestations was to be established. Since X-ray findings are the more important in the diagnosis of skeletal dysplasias, the **emphasis was to be on the radiological aspects** with a complement of applicable clinical and other non-radiological features.

- **ii) A Database of Disease Entities.**

This would contain, in record form, as many pathological conditions as could be gleaned from the resources outlined previously.

3.4.4.2 Create a set of Application Programs with the following aims in mind:

- i) Provide a mapping between the set of disease entities and the common pool of features, striving to maintain data integrity and minimizing redundancy.

- ii) Create a user-friendly interface.

- iii) Write programs that would:

  - Aid the diagnosis of skeletal dysplasias;

  - Answer queries which are not easy to satisfy with the gamut index approach;

  - Retrieve other useful information;

  - Provide database administrative (DBA) procedures such as the facility to add new features and diseases, changing features, editing disease profiles as well as other utilities that may be needed to maintain a small system.

SECTION FOUR.

THE FORMALIZATION STAGE

CHAPTER 4.1
INTRODUCTION.

The formalization stage is important and extensive and will be

presented as an autonomous section.

According to Buchanan et al (1983), "the formalization stage should involve the mapping of the concepts, subproblems and information flow characteristics identified during conceptualization into more formal representations based on various knowledge-engineering tools or frameworks. It is at this stage that the knowledge engineer takes on a more active role by informing the expert about existing tools, representations and possible types of problems. The output of this stage is a set of partial specifications describing how the problem can be represented within the chosen tool or framework."

The author, having assumed the role of knowledge engineer, decided to use the above quidelines during the formalization stage of this undertaking. During this stage descriptions of the various components and structures that were to make up this project, were formulated. In addition the tasks that were to be performed, were defined.

CHAPTER 4.2
THE KNOWLEDGE BASE DESCRIPTION.

As explained in the previous section, the knowledge base was to consist of two components: namely, a pool of diagnostic features and a file of disease descriptions.

4.2.1 Diagnostic Features (radiological and clinical).

A random access computer file, consisting of 1024 records and thus catering for a maximum of 1024 disease manifestations, was chosen as the structure for this component of the knowledge base. The significance of this particular number will become clear when the subject of bitmapping is broached in section 4.3.1. It was considered large enough for this project but it should be emphasized that a simple adjustment in the design will allow for a larger structure, the size of which is limited only by computer memory considerations.

Each record of this file describes one manifestation by means of a sentence consisting of up to 70 characters. While it should be possible to make minor alterations to a particular record such as correcting spelling errors, the basic meaning conveyed by the applicable sentence is to be rigorously maintained. The positions of the records in the file are permanent and of crucial significance. To allow a change of meaning would seriously degrade the integrity of the database. An example will be presented:

Consider the following radiological feature:

- Ribs Broad

While this could be changed to "Ribs Broad (oar-shaped)" without a serious change in meaning, changing it to "Ribs Short", say, will result in the unsatisfactory situation where ALL the conditions that feature broad ribs, will now have the characteristic of short ribs instead. This may be regarded as a disadvantage of the "one fact in one place" concept associated with high degrees of database normalization. In order to deal with this anomaly, it was decided to make the editing facility password-controlled so that only individuals conversant with allowable alterations would be able to change the text of the manifestations file.

The file of diagnostic features was implemented in Turbo Pascal as the following structure:

- Feature_File : File of String70;

where string70 had been declared a data type as follows:

- String70 = String[70];

The features were to be stored permanently on disk in a file named Features.Ran. To use this random access features file, the following lines of code would have to appear near the beginning of the procedure which addresses the file:

```
Assign (Feature_File,'Features.Ran');
Reset  (Feature_File);
```

The use of the two names, Feature_File and Features.Ran, is the consequence of the quirk of the Turbo Pascal dialect. It requires that a disk file must be addressed through a declared file structure in the manner outlined.

The file Features.Ran was partitioned as illustrated in fig 4.1 on the next page. It is shown in its entirety in appendix D.

4.2.2 The Diseases File.

The file of disease descriptions was also implemented as a random access file with a somewhat more complicated structure, as follows:

Disease_File : File of Disease_Record;

The Disease_Record was declared as the following structure:

```
Disease_Record =  Record

  Available       : Boolean;  {1}
  Profile         : String50; {50}
  Name:           : String50; {50}
  Synonym         : String50; {50}
  Abbreviation    : String6;  {6}
  Genetics        : String50; {50}
  Note1           : String70; {70}
  Note2           : String70; {70}
  Note3           : String70; {70}


                  {Total      417}
```

The numbers shown in braces above indicate that every disease description would occupy 417 bytes. Stated differently one can say that 200 conditions would need 83 400 bytes of storage space, a figure which is modest and indicative of significant economy of storage resources.

SECTION	FROM RECORD	TO RECORD
NON-SKELETAL	0	127
SKULL	128	191
THORAX	192	255
SPINE	256	383
PELVIS	384	447
TUBULAR BONES	448	575
JOINTS	576	767
HANDS	768	895
FEET	896	959
GLOBAL SKELETAL	960	1023

Fig 4.1 Partitioning of File Features.Ran.

4.2.2.1 Explanation of the Disease Record.

The functions of the various record fields are as follows:

Available: This field is a simple Boolean variable which denotes whether a particular record is available or not. If so, a new disease entity may be inserted into it.

Profile: This 50 character string variable represents the whole description of the disease profile. The mechanics of this profile construct is crucial to this project and will be dealt with extensively in chapter 4.3. Here it will merely be stated that it contains the code for the mapping from the disease file to the features file. Decoding this string in a way that will be described in 4.3.4, produces a list of numbers, each representing the position of a manifestation in the features file. The diagnostic features can thus be selected by means of these numbers and listed on an output device.

Name: A character string depicting the most widely accepted name of the disease entity.

Synonym: Since diseases often have alternative names, it was considered prudent to include this facility in the disease record.

Abbreviation: This field caters for generally accepted abbreviations for the diseases, eg. "OI" for Osteitis Imperfecta.

Genetics: The original intention was to code the genetics of the various diseases in a fashion similar to the diagnostic features. The genetic descriptions vary considerably in spite of the fact that the domain of possibilities is rather limited. It was therefore decided to describe the genetics by means of a 50 character sentence. Keywords such as AD, AR and XL could still be employed for retrieval purposes.

Notes: Note1, Note2 and Note3 provide a total of 210 characters for terse notes such as references and other apt remarks.

CHAPTER 4.3
MAPPING THE DISEASE PROFILE.

Extensive heuristic efforts eventually resulted in a storage structure for disease profiles which will be described in this chapter.

4.3.1 Bit Maps.

Each disease profile is represented by a matrix consisting of 1024 bits where each bit would indicate the presence or absence of a particular manifestation. Fig 4.2 shows such a map for Carpenter Disease while Achondroplasia is represented by the matrix demonstrated in fig 4.3. It is evident that each condition will have a distinctive bit map. At any time, bits may be set (changed to the one-state), to indicate the presence of the corresponding diagnostic feature in the profile, or reset (changed to the zero-state), denoting the absence of a feature from the picture. A particular bit in the matrix will always represent the same manifestation, regardless of the disease being represented. The various disorders therefore, use the same matrix, but each condition will have different set bits.

The concept of bit maps has been well described (Tremblay and Sorenson, 1976; Sohr, 1983). Bit maps consist of bit strings made up of characters from the binary alphabet which consists of 0 and 1 only. The bit map under discussion consists of a bit string which is 1024 bits in length. This will be handled by the computer as a structure consisting of 128 (1024/8) bytes represented diagrammatically in figs 4.2 and 4.3.

Bit maps result in very compact storage representation of information (Tremblay and Sorenson, 1976a). Furthermore, it is important to note that a bit map is a fixed-length item representing a variable number of values from the particular domain. In the knowledge base under discussion, every disorder has a profile consisting of a variable number of features, but which is represented by a field which has the same length in every case. In practice this means that, no matter how many stigmata a disease manifests (within reason), the size of the disease profile in computer storage will remain the same. Tremblay and Sorenson emphasize the importance of this fact since variable length items normally generate variable length records,

00000000 00000000 00000000 00000000 **10000000** 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 **00000100 00000100** 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 **00110000**
00000000 00000000 00000000 00000000 00000000 00000000 **10000000** 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

The set bits represent diagnostic features

31 Cardiac Defect
146 Prominent frontal bones
154 Craniostenosis
772 Polydactyly/hexadactyly (hands)
773 Syndactyly (hands)
847 Reduplicated proximal phalanx of thumb

Fig 4.2 Bit map of Carpenter Syndrome

00000000 00000000 00000000 00000000 **00000100** 00000000 00000000 **00000110**
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 **11000000** 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 **00000001 10000000** 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 **00001000** 00000000 00000000 00000000
00000000 **00000001** 00000000 00000000 00000000 00000000 **00000001** 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00001100 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

The set bits represent diagnostic features:

1 Small stature or Dwarfism
2 Unusual/characteristic facies
26 Depressed nasal bridge
174 Shortened base of skull
175 Small foramen magnum
279 Lumbar spinal canal narrowed (stenosed)
280 Interpedicular narrowing of lumbar spine
411 Small/shortened sacro-iliac notches
456 Short shafts of tubular bones
496 Tubular bones with metaphyseal flaring

Fig 4.3 Bit map for Achondroplasia

the processing of which is far less efficient than is the case with fixed-length records, the approach favoured for this project.

cessful in representing "ordered lists" as compressed bit maps which, with appropriate programs, could be expanded again. The advantages included significantly reduced storage requirements, vastly increased speed of list manipulation and the ability to carry out useful logical operations on the bit strings.

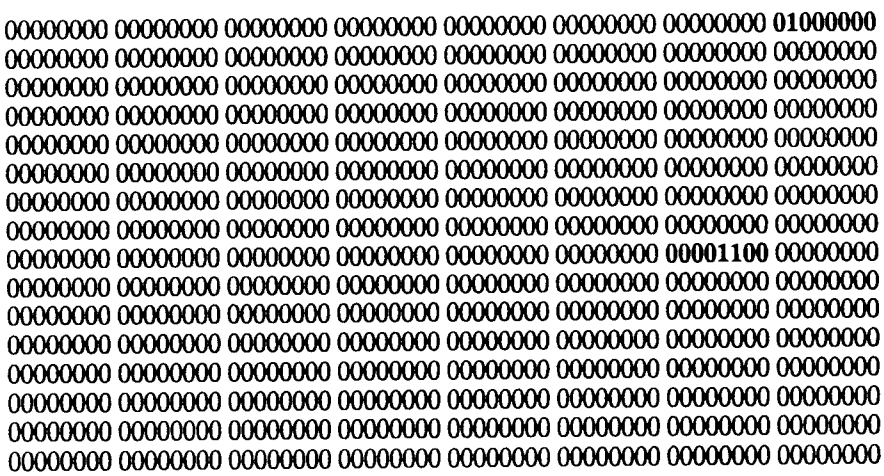


Fig 4.4 Representative bit map.

The set bits represent manifestations 6, 522 and 523 from feature file.

4.3.2 Compressed Bit Mapping.

It is clear that most diseases will have relatively few manifestations. By far the largest part of each profile will be represented by zeros, indicating features which are, in this case, conspicuous by their absence. In mathematical parlance it can be said that this method of disease profile representation results in a sparse matrix, with most of the bits not conveying any information. It is, however, possible to condense such matrices by removing redundant all-zero bytes.

One such method is bit map compression as implemented by Sohr (1983) during the development of a literature manipulation system (LITMAS) in Montana, USA. Using the technique on a 6502-based Apple II microcomputer, Sohr was suc-

4.3.3 Compression Technique employed in this Project.

To illustrate the approach followed with this study in an 80286/8088 environment, use will be made of an example.

Figure 4.4 illustrates an arbitrary disease which manifests only three diagnostic features. This example happens to illustrate the original representation of Carpenter syndrome which at the time comprised only these three features namely craniostenosis, syndactyly and thumb reduplication. These stigmata initially occupied

records 6, 522 and 523 in the file features.ran. The three numbers form a small ordered list. The profile for the fictitious disease may be regarded, for processing purposes, as comprising all the manifestations in the knowledge base, three of which in a "true" way (ie. present and represented with ones) and the other 1021 in a "false" way (not present and represented with zeros).

Bit strings are most conveniently demonstrated from right to left with the lower order bits to the right of the higher order bits, in the same fashion as decimal integers. The bits are numbered from 0 to 1023 and a back to front representation results. The top rightmost bit is bit no. 0 and the leftmost bit in the bottom row is bit no. 1023. The bits are grouped into 128 eight-bit bytes, the numbering of which also starts with 0 in the top right hand corner and ending with no 127 at the bottom left.

Consider byte no. 0. Bit no. 6 is set indicating the presence of manifestation no. 6 of the knowledge base. This position in the features file can be calculated by multiplying the byte number by eight and adding the bit position: 0 x 8 + 6 = 6. Similarly the set bits in byte no. 65 indicate the presence of features no. 522 (65 x 8 + 2) and 523 (65 x 8 + 3). All the other bits in the matrix are zero, indicating the absence of all the other features in the knowledge base, from this particular condition.

In order to deal with the sparse matrix problem with its associated redundance of bits which convey no information, the following technique was developed:

Only the bytes which convey information (bytes 0 and 65 in the example) were to be stored, together with a few newly computed bytes which would be needed to re-establish the original matrix. Towards this purpose three levels (tiers) of bytes were defined and named Tier0 bytes, Tier1 bytes and Tier2 bytes respectively.

The Tier2 bytes represent the data conveying bytes from the matrix (bytes 0 and 65 of the example). The example Tier2 structure, consisting of the two bytes may be represented as follows:

```
          00001100 01000000
bit no.   76543210 76543210

Tier2 byte no.      65          0
```

The uncompressed matrix consists of 128 (1024/8) bytes, each of which may be represented as a bit in 16 (128/8) Tier1 bytes. Once again the majority of bits in this smaller tier1 matrix will be zero and only the bytes containing set bits were to be stored, resulting in the following tier1 structure:

```
          00000010 00000001
bit no.   76543210 76543210

Tier1 byte no.  8          0
```

The retained tier1 bytes may be represented by set bits in 2 (16/8) tier0 bytes

which, for convenience, will be regarded as a single 16-bit structure:

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1

The two tier0 bytes are always retained as a single structure, regardless of the sparseness of the small tier0 matrix, ie. no further compression is applied.

The full representation of the disease profile of the example case will therefore be the following six bytes:

Tier2	00001100 01000000
Tier1	00000010 00000001
Tier0	00000001 00000001

As a single alphabetic character can normally be represented with a byte, the significant economy of this approach is readily apparent. In the example the whole disease profile is represented in the space otherwise taken up by a six-letter word. Diseases with more extensive profiles naturally will require more space but in practice 30 bytes were never exceeded in the make-up of a disease profile.

4.3.4 Restoring the ordered List.

The total number of bytes in a profile can readily be ascertained by an application program. By design the tier0 bytes always number two, while it is a trivial matter to determine the number of tier1 bytes by counting the number of set bits in the tier0 bytes by means of a bit rotating routine in assembly language. The remaining bytes must belong to tier2. Referring to the exam ple case, the various bytes may be illustrated as follows:

00001100 01000000	00000010 00000001	00000001 00000001
Tier2	Tier1	Tier0

To decode, the following steps are taken:

The set bits in the tier0 bytes are inspected. In the example bits no. 0 and 8 are set, indicating that the stored tier1 bytes are byte no's. 0 and 8 respectively.

The tier1 bytes are investigated next:

Tier1 byte no. 0 represents tier2 bytes 0 to 7. Bit 0 of this byte is set indicating that the first stored tier2 byte is no. 0. Tier1 byte 8 represents tier2 bytes 64 (8 x 8) to 71. Bit 1 is set indicating the presence of tier2 byte no. 65 (64 + 1).

Processing the tier2 bytes then follows:

Tier2 byte no. 0 represents features 0 to 7. Since bit 6 is set, feature no. 6 is an element of the ordered list.

Tier2 byte no. 65 represents features 520 (65 x 8) to 527. Bits 2 and 3 are set indicating the presence of features 522 (520 + 2) and 523.

In summary it may be stated that encoding the ordered list reduces to repeated division by 8:

i) 523/8 = 65 remainder 3 ie. set bit 3 of tier2 byte 65.

522/8 = 65 remainder 2 ie. set bit 2 of tier2 byte 65.

6/8 = 0 remainder 6 ie. set bit 6 of tier2 byte 0.

ii) 65/8 = 8 remainder 1 ie. set bit 1 of tier1 byte 8.

0/8 = 0 remainder 0 ie. set bit 0 of tier1 byte 0.

iii) 8/8 = 1 remainder 0 ie. set bit 0 of tier0 byte 1.

0/8 = 0 remainder 0 ie. set bit 0 of tier0 byte 0.

Decoding the compressed bit map follows essentially an opposite routine. Initially the tier1 byte numbers are calculated:

Tier1 byte number = Tier0 byte number x 8 + set bit in tier0 byte

Thus the first tier1 byte = 0 x 8 + 0 = 0.

Similarly the second tier1 byte = 1 x 8 + 0 = 8.

The tier2 byte numbers are then determined :

Tier2 byte no. = Tier1 byte no. x 8 + set bit in tier1 byte

First tier2 byte = 0 x 8 + 0 = 0;

Second tier2 byte = 8 x 8 + 1 = 65.

Finally the elements of the ordered list, ie. the feature numbers are then calculated as follows:

Feature number = tier2 byte no. x 8 + set bit in tier2 byte.

The feature numbers making up the profile of the example are therefore as follows:

0 x 8 + 6 = 6;

65 x 8 + 2 = 522;

65 x 8 + 3 = 523.

As far as the author could determine, this variation of compressed bit mapping had not been previously used. Sohr (1984) employs a similar strategy in a 6502 microprocessor environment and the basic idea must be credited to him. The strategy of a fixed number of first level bytes (designated tier0 by the author), however, is unique to this project and, as far as can be ascertained, it is the first application of compressed bit mapping in the 80286/8088 environment; in the computer representation of genetic disease in general, and of the skeletal dysplasias in particular.

It is interesting to note that the database of diseases as contained by the file profiles.ran, may be regarded as a normal-

ized 9-tuple relational table, one of the attributes being the field "profile" which holds the bit map. Since the bit map can be decoded to reveal several manifestations, the attribute ceases to be atomic, in data base terms, when thus processed. The table may thus be viewed as being normalized and unnormalized at the same time!

CHAPTER 4.4
THE APPLICATION PROGRAMS.

As was indicated in section 3.4.4, the knowledge base constituted one main component of this project. The other ingredient is the set of application programs by means of which meaningful information was to be gleaned from the knowledge base.

In this chapter brief descriptions of the functions of the programs are presented. Full listings are provided in Appendix A and Appendix B.

The following is a list of the application programs. These were all designed and written by the author for use in this project.

Pascal Programs:

- Menu.pas
- Mkdescrp.pas
- Features.pas
- Disenter.pas
- Prospect.pas
- Dsearch.pas
- Testlook.pas
- Profedit.pas
- Library.pas
- Proflib.pas
- Asslib.pas

Assembly language programs:

- Search.com
- Compare.com
- Decode.com
- Convert.com
- Scan.com
- Passwd.com

The fact that both MSDOS and Turbo Pascal limit file names to eight characters or less (excluding the extension), explains the cryptic nature of some of the program names.

4.4.1 Pascal Programs.

4.4.1.1 Menu.pas

This program presents the utilities offered by the package, in the form of a

menu from which selections may be made. It will be described in section 6.

4.4.1.2 Mkdescrp.pas

This Pascal program was used at the beginning of the project to create a small descriptor file named Descriptor.ran in which the commencing addresses of the various sections of features are being kept. As was shown in fig. 4.1, the features file is partitioned and the sections consist of differing numbers of records. When dealing with the implementation stage, it will be shown that, after consultation with one of the domain experts (Spranger,1985), it was decided to present the diagnostic features under the headings as enumerated in fig 4.1. Each section can contain a specific maximum number of features. These numbers vary from section to section, hence the place in the features file where each section commences, should be known. Program Mkdescrp.pas provides these addresses to the file Descript.ran.

4.4.1.3 Features.pas

As will be explained more fully under Implementation in section 5, the various manifestations which were used to form a kernel for the program file, were initially entered into ten ordinary ASCII text files by means of the word processor Wordstar, version 3.30, used in the N-mode (N for non-document).

Program Features.pas was used to copy the features into the Pascal random access file, Features.ran, ensuring that the various sections commence at the appropriate positions in the file. Unused records, available for the later addition of new stigmata are marked as available by storing "XX" in them.

4.4.1.4 Disenter.pas

Information regarding individual diseases is entered into the knowledge base by means of this program. As it will be discussed in detail in section 5, only a brief summary of its function will be presented here. It commences with presenting the logo shown in fig 4.5. When the required items have been typed in, it offers an elementary editing facility to rectify errors that might have been made. The program then checks for the possible existence of the disease entity. If a duplicate is found, the operator is alerted, otherwise it produces the logo depicted in fig 4.6 from which a particular section may be selected. A screen-full of manifestations is then presented, from which features relevant to the disease may be selected. A section may consist of several screens of features, hence means are provided of "paging" forwards and backwards. Stigmata erroneously selected, may be cancelled, and the descriptions of the diagnostic features may be edited at this stage. The latter facility is password controlled for the reason stated in chapter 4.2.

It may be added that sections from which an exit has already been made, may be revisited at any time in order to select additional or to cancel wrongly chosen features.

Once all the applicable features, radiological and otherwise, have been selected, the integers which represent their positions in the 1024 bit matrix are sorted into an ordered list. This in turn is

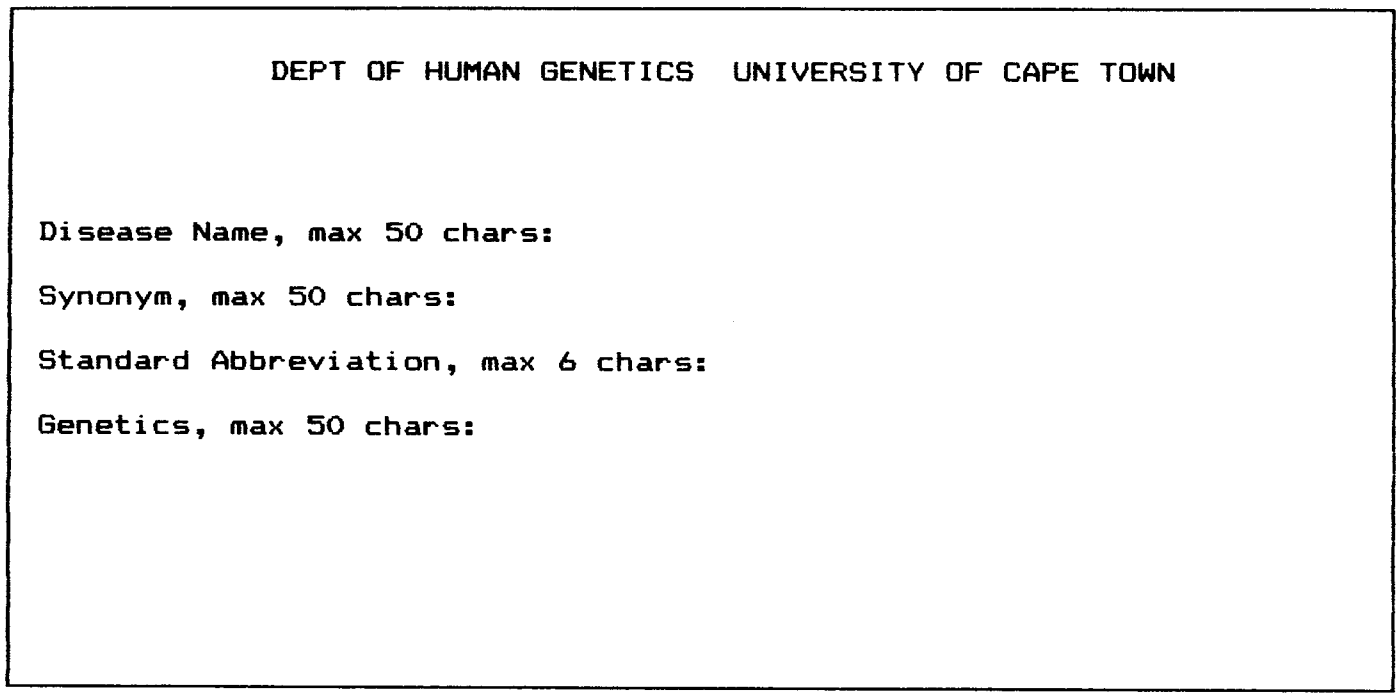


Fig. 4 - 5 Logo for Profile Entry.

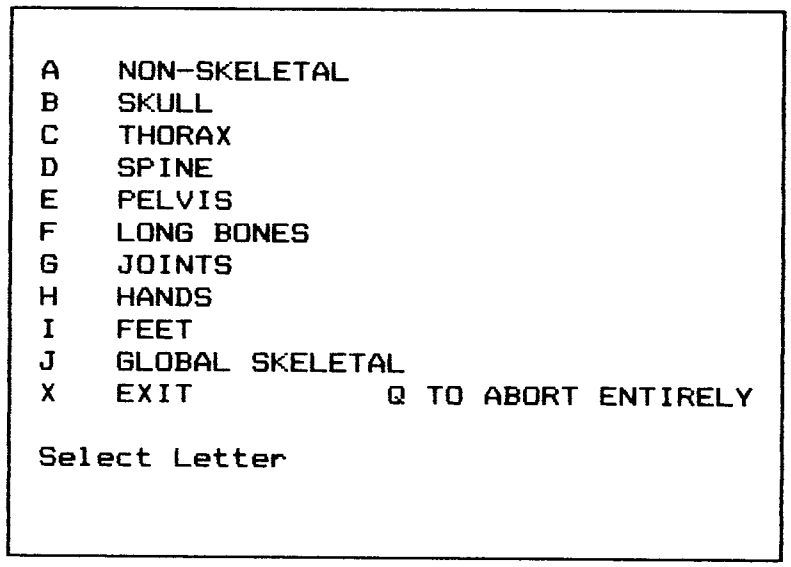


Fig 4 - 6 Menu for Selection of Diagnostic Features.

converted into a compressed bit map as explained in 4.3.3. The compressed bit map is created by a procedure written in assembler language and contained in program Convert.com which is described later in 4.4.3.

Finally the program enters the various items of information into the respective fields. The resulting record is written to the disk file Profiles.ran.

4.4.1.5 Testlook.pas

This program was needed to inspect disease profiles, and should be regarded as part of the development armamentarium, and not as a utility. It did not therefore find application in the final package. In addition, it assumes that the position of the required disease record in the file Profile.ran, is known to the operator, a situation which, clearly, will only rarely prevail.

The program prompts for a number. If the record indicated by this number is within the range of the knowledge base, it will be retrieved and the relevant information displayed.

4.4.1.6 Dsearch.pas

Disease profiles are retrieved by means of this program in the following way:

Any part of the name or synonym of a disease is entered as a character string. This string is split into two halves and the commencing half is used as a search string which is scanned against all the names and existing synonyms in the data base. Every time a match is found, the relevant information, including the list of decoded features, is presented on the screen. The program will be described in greater detail in sections 5 and 6.

4.4.1.7 Prospect.pas

This is the major retrieval program in the applications set. It compares a test profile, such as belonging to a problem case under investigation, with the profiles currently known to the system and provides information on similarity and non-agreement.

During its initialisation process, the program "reads in" the tier0 bytes of every condition in the disease file. These are kept in an array in the primary memory of the computer where they are rapidly accessible. A list of diagnostic features which describes the unknown condition as accurately as possible, is selected from the features file by the operator in a way similar to that used for entering a new disease profile. The resulting ordered list is encoded in the manner described in 4.3.3. The tier0 bytes of the unknown condition are then compared with each pair of tier0 bytes in the array by means of an assembler program, Search.com. If there are no set bits common to the two pairs of bytes, there can be no features in common, and the particular condition from the data bank will be ignored.

As soon as common set bits are found at tier 0, the profile from the data bank is decoded. The resulting ordered list is compared with that from the test profile by means of an assembler procedure "Compare.com", which produces a further ordered list consisting of features common to the two lists under comparison. Expressed in Boolean terms:

$C = A.B$

where C is the "common" list, A the "unknown" list and B the "known" list.

A Pascal procedure "CANCEL" processes the three lists and produces two further lists; the one with features in the "unknown" but not in the "known" profiles ($C1 = A.\overline{B}$); the other with features in the "known" profile but not in the "unknown" ($C2 = \overline{A}.B$).

A level of congruence between the known and unknown profiles is employed as a measure of completeness of the matches. This level is initially set equal to the number of features in the unknown profile, but searches may be repeated at lower levels if necessary. The higher the level at which a match is found, the greater the similarity between known and unknown. The level may be reduced right down to 1 in which case a match will indicate the presence of only one feature in common. At every presentation the operator is supplied with the list of common features followed by the other two lists ($\overline{A}.B$ and $A.\overline{B}$). In this fashion the operator may be alerted to possibly overlooked features which may be present in the unknown profile. Similarly, it may be possible to discover features applicable to a particular condition which were not known to the knowledge base before and which may then be added.

4.4.1.8 Profedit.pas

Should any changes to a disease description be desired, this program may be used. Any of the "text" attributes may be changed by elementary editing, while features may also be added to or deleted from the profile and a new compressed bit map computed. Profedit.pas will be described in detail in section 5.

4.4.2 Pascal Library Programs.

4.4.2.1 Library.Pas.

This library contains utility procedures and functions used in, but not necessarily specific to this project. The utilities of which listings are provided in Appendix A, are as follows:

- **Procedure Msg** prints a required message at a desired position on the screen.
- **Procedure Frame** draws a double-rimmed "box" of any required size, using the IBM ASCII character set.
- **Function Stupcase** converts all the characters in a string from lower to upper case. This function was obtained from the Turbo Pascal "Toolbox" (Turbo Graphix Toolbox Reference Manual, 1985).
- **Procedure Flush** initializes all the elements of a two-dimen sional integer array to zero.
- **Procedure Bubble_sort** sorts an integer vector in ascending order according to the bubble sort algorithm (Tremblay and Sorenson, 1975b).
- **Procedure Cancel** compares one-dimensional arrays A and B and

forms a one-dimensional array C which contains those integers present in A but not present in B, i.e C = A.$\overline{B}$. The integers in C are counted.

- **Procedure Fill** stores duplicates of any required integer in the elements of a one-dimensional array.

- **Procedure Pause** lets a program wait until a key is pressed on the keyboard.

- **Procedure Respond** forces a program to accept only the characters "y", "Y", "n" and "N" to a particular question.

- **Procedure String_Edit** allows the text of any string up to 70 characters long, to be changed.

4.4.2.2 Proflib.pas

Proflib contains the following procedures dedicated to this system. Listings are provided in Appendix A:

- **Procedure Present_Features** displays 20 disease stigmata on the screen, obtaining them from the file "Features.ran".

- **Procedure Process_Features** allows the operator to select diagnostic features from the lists provided by the previous utility, for inclusion in a disease profile that is being compiled. Any reasonable number of stigmata may be selected. Provision is made to cancel items that may have been selected erroneously.

- **Procedure Dsearch** scans the Name and Synonym fields in the disease profiles table for occurrences of a particular character string.

- **Procedures Logo1 and Logo2** provide logos for some of the interactive programs.

4.4.2.3 Asslib.pas.

This library contains the assembler programs described in 4.4.3.

4.4.3 Assembly Language Programs.

Six assembler programs were written to accommodate procedures which would have executed too slowly if implemented in Pascal code. Listings of the source code (with extension .asm) are provided in Appendix B. These programs were assembled to modules with a .com extension according to the specifications for proper interfacing as outlined in the reference manual (Turbo Pascal Version 3.0 Reference Manual, 1985). The functions of the modules are as follows:

- **Convert.com** changes an ordered list of integers into a compressed bitmap as discussed in 4.3.3. Two tier0 bytes and as many bytes on the other two tiers as required by a particular disease profile, are generated.

- **Decode.com** performs a process opposite to "Convert.com". It processes the profile bytes to form an ordered list of integers, as described in 4.3.4.

- **Compare.com** compares two ordered lists of integers and forms an additional one which contains those elements which the other two vectors have in common.

- **Search.com** compares the tier0 bytes from the disease profiles with those of a profile under test. It scans from a particular position in the profiles.ran file (initial position is zero) until it finds a Tier0-pair from the disease profiles which has set bits in common with the test pair. A logical AND-operation is used in the comparison. If no common set bits are found in a particular comparison, there cannot be any stigmata which are shared by the known and unknown profiles. As soon as corresponding set bits are found, the position of the profile in question is returned to the main program for processing.

To speed up the search process, the following strategy is followed:

As part of the initialization process, the program "Prospect.pas" copies the tier0 bytes from all the known disease profiles in the diskfile "profiles.ran" into a two-dimensional array in primary memory. The search is conducted on the bytes in this array rather than on the tier0 bytes in the records of the file, thus effecting a significant speed gain.

- **Scan.com** scans any character string for the presence of a substring. By employing this procedure at assembly level rather than in the higher language, a significant speed advantage is obtained. An example of this procedure would be to enter any part of a disease name or synonym. All the relevant fields of the file "Profile.ran" may be scanned using this procedure. Should a match be found, the applicable record may be copied from disk into main memory for processing.

- **Passwd.com** provides password control at strategic points in the application programs. The password cannot easily be discovered by means of the conventional "unassemble" techniques, nor can it be gleaned without difficulty from the source code (Passwd.asm). Since great motivation to "crack" the password was not anticipated, elaborate security measures were not considered necessary.

SECTION FIVE.

THE IMPLEMENTATION STAGE

CHAPTER 5.1
INTRODUCTION.

The implementation of the formal structures on the microcomputer will be described in this section.

Although there might appear to be a rigid division between formalization and implementation, this was not the case in practice. Too much heuristics were involved to allow for the complete orderly design of data structures and programs and their implementation in two discrete stages. The data structures were indeed designed almost to finality during the initial stages of the project, but frequent adaptations to both programs and structures became necessary during implementation. This often resulted in new formalization and alterations to the design.

The descriptions of the programs given in Section 4 are presented there for convenience. Finality was only reached after implementation and testing. Indeed, true finality has not yet been accomplished as development is an on-going process.

CHAPTER 5.2
A NUCLEUS OF DIAGNOSTIC FEATURES.

A prototype knowledge base is usually implemented by using whatever knowledge engineering aids are available for the chosen project (Buchanan et al, 1983). These aids include editors, intelligent editors and acquisition programs. In this chapter it will be shown how information was collected, prepared into files by means of text editing, and prepared for computer access by means of appropriate programs.

A nucleus of diagnostic features for the knowledge base was drawn up by listing the discrete manifestations that could be gleaned from the Gamut Index (Kozlowski & Beighton, 1984). This collection of manifestations was then classified under the following sections:

- Skull
- Spine
- Thorax
- Pelvis
- Tubular bones
- Extremities
- Joints
- General

Using the word processing package Wordstar Version 3.30, the various groups of diagnostic features were entered into text files on the computer. The files were named as follows:

- Skull.txt
- Spine.txt

- Thorax.txt
- Pelvis.txt
- Tubular.txt
- Extreme.txt
- Joints.txt
- General.txt

Printouts of these preliminary texts were produced and presented to the domain experts for critical comment. Many redundancies were eliminated and ambiguities resolved with this approach.

After perusal of suggestions received from domain expert Spranger the grouping of the manifestations was altered and new text files produced, conforming to the following scheme:

- Non-skeletal
- Skull
- Thorax
- Spine
- Pelvis
- Tubular bones
- Joints
- Hands
- Feet
- Global skeletal.

The corresponding computer files were named as follows:

- Soft.txt (Manifestations mainly soft tissue)
- Skull.txt
- Thorax.txt
- Spine.txt
- Pelvis.txt
- Tubular.txt
- Joints.txt
- Hands.txt
- Feet.txt
- Global.txt

Further diagnostic features for inclusion in the various text files were obtained from the literature when the latter was consulted for the amplification of disease descriptions. This will be referred to under the treatment of the disease profiles in chapter 5.3. The complete list of diagnostic features in its ultimate form is presented in appendix D.

5.2.1 Random access feature file.

Once the text files were considered to be provisionally complete, the program Mkdescrp.pas (see appendix A) was executed in order to create the small descriptor file from which the commencing addresses of the various sections could be obtained. By means of program Features.pas (appendix A) the random access

file Features.ran was produced. The listing in appendix A on page A-6 illustrates that provision is made for the following numbers of items in each category:

- Non-skeletal 128
- Skull 64
- Thorax 64
- Spine 128
- Pelvis 64
- Tubular bones 128
- Joints 192
- Hands 128
- Feet 64
- Global skeletal 64
- Total 1024

Stated briefly, these two programs converted the various files that had been produced with the text editor, into a computer storage structure, consisting of 1024 records, and suitable for random access by application programs. This implementation would allow for rapid processing by both Pascal and Assembler procedures. This random file constitutes one of the two important components of the knowledge base, namely the supply of diagnostic features from which known and unknown disease profiles could be established. The other component is introduced in the next chapter.

CHAPTER 5.3
A NUCLEUS OF DISEASE PROFILES.

5.3.1 Main Sources of Information.

Descriptions of 201 skeletal disorders were obtained from the resources enumerated in chapter 3.3. The Gamut Index provided adequate information on several of the diseases that were included in the knowledge base. Where it was felt that the disease profiles should be amplified, the domain experts were consulted, or more information sought from papers in the literature. The latter approach was used in the following conditions:

Aarskog Syndrome (Berry et al, 1980; Furukawa et al, 1972)

Acro-osteolysis (Gorham type) (Gorham, 1955)

Acro-osteolysis (Hajdu-Cheney type) (Cheney, 1965; Weleber & Beals, 1976)

Acro-osteolysis (Phalangeal type) (Brown et al, 1976)

Acro-osteolysis (Torg type) (Torg et al, 1969)

Acrodysostosis (Reiter, 1978)

Acromesomelic Dysplasia (Beighton, 1974)

Aicardi Syndrome (Phillips et al, 1978)

Apert syndrome (Beligere et al, 1981; Schauerte & St-Aubin, 1966)

Arteriohepatic Dysplasia (Levin et al, 1980)

Basal Cell Nevus Carcinoma Syndrome (Gorlin & Goltz, 1960)

Beckwith-Wiedeman Syndrome (Lee, 1972)

Blount Disease (Bathfield & Beighton, 1978)

Cerebral Gigantism (Sotos Syndrome) (Poznanski & Stephenson, 1967)

Cerebro-costo-mandibular Syndrome (Leroy et al, 1981; Williams & Sane, 1976)

Cerebro-hepato-renal (Zellweger) Syndrome (Williams, 1972)

Cerebro-oculo-facio-skeletal Syndrome (Pena & Shokeir, 1974)

Cheirolumbar Dysostosis (Wackenheim, 1981)

Chondrodysplasia Punctata, Dominant Form (Shaul et al, 1975)

Chondrodysplasia Punctata, Rhizomelic Form (Gilbert et al, 1976)

Cockayne Syndrome (Riggs & Seiberg, 1972)

Coffin-Lowry Syndrome (Hunter et al, 1982)

Craniometaphyseal Dysplasia (Beighton et al, 1979)

Dentinogenesis Imperfecta (Beighton, 1981)

Diaphyseal Aclasis (Epstein & Levin, 1978)

Diaphyseal Dysplasia (Sparkes & Graham, 1972)

Diastrophic Dysplasia (Lachman et al, 1981)

Distichiasis-Lymphedema Syndrome (Robino et al, 1970)

Dyschondrosteosis (Langer, 1965)

Dysplasia Epiphysealis Hemimelica (Connor et al, 1983)

Dyssegmental Dysplasia (Gruhn et al,1978)

Enchondromatosis (Ollier Disease) (Spranger et al, 1978)

Fanconi Pancytopenia Syndrome (Glanz & Fraser, 1982)

Femoral Facial Syndrome (Eastman & Escobar, 1978; Lord & Beighton, 1981)

Fibrodysplasia Ossificans Progressiva (Cremin et al, 1982)

Freeman-Sheldon Syndrome (O'Connell & Hall, 1977)

Frontometaphyseal Dysplasia (Beighton & Hamersma, 1980)

Fucosidosis (Brill et al, 1975)

Goldenhar Syndrome (Setzer et al, 1981)

Holt-Oram Syndrome (Kaufman et al, 1974)

Humerospinal Dysostosis (Kozlowski et al, 1974)

Hypoplastic Radius and Tibia type of Mesomelic Dysplasia (Leroy, 1975)

Incontinentia Pigmenti (Morgan, 1971)

Klinefelter Syndrome (Oshawa, 1971)

Klippel-Feil Syndrome (Da Silva, 1982)

Klippel-Trenauny-Weber Syndrome (Phillips et al, (1978)

Langer type of Mesomelic Dysplasia (Espiritu et al, 1975)

MLS I (Spranger et al, 1977)

MLS II (Lemaitre et al, 1978)

MLS III (Melhem et al, 1973)

MPS I-H (Hurler Syndrome) (Stevenson, 1976)

MPS I-S (Scheie Syndrome) (Stevenson, 1976)

MPS II (Hunter Syndrome) (Young et al, 1982)

MPS III (Van de Kamp et al, 1976)

MPS IV (Langer & Carey, 1966)

MPS VI (Strumpf et al, 1973)

MPS VII (Sly et al, 1973)

MURCS Association (Duncan et al, 1979)

Maffucci Syndrome (Anderson, 1965)

Mannosidosis (Spranger et al, 1976)

Marshall Syndrome (Marshall et al, 1971)

Melorheostosis (Campbell et al, 1968)

Menkes Kinky Hair Syndrome (Danks et al, 1972)

Mesomelic Dysplasia (Beighton, 1974)

Metaphyseal Chondrodysplasia, Mc-Kusick Type (Lux et al, 1970)

Metaphyseal Dysplasia (Pyle Disease) (Heselson et al, 1979)

Metatropic Dysplasia (Kozlowski et al, 1976; Marotoux et al, 1966)

Multiple Pterygium Syndrome (Escobar et al, 1978)

Myotonic Chondrodysplasia (Horan & Beighton, 1975)

Nail-patella Syndrome (Bennett et al, 1973)

Oculodento-osseous Dysplasia (Beighton et al, 1979)

Oculomandibulo-facial Syndrome (Steel & Bass, 1970)

Oro-facial-digital Syndromes (Haumont & Pelc, 1983)

Osteopathia Striata Syndromes (Gehweiler et al, 1973; Horan & Beighton, 1978)

Osteopoikilosis (Melnick, 1959)

Osteoporosis-Pseudoglioma Syndrome (Neuhauser et al, 1976)

Pachydermoperiostosis (Hedayati et al, 1980)

Phocomelia (Roberts Syndrome) (Hall & Greenburg, 1972)

Pseudoachondrodysplasia (Heselson et al, 1977)

Pseudohypoparathyroidism (Steinbach & Young, 1966)

Radial aplasia-Thrombocytopenia Syndrome (Hall et al, 1969; Luthy et al, 1979)

Reinhardt-Pfeiffer type of Mesomelic Dysplasia (Reinhardt & Pfeiffer, 1967)

Robinow type of Mesomelic Dysplasia (Robinow et al, 1969)

Rubinstein-Taybi Syndrome (Filippi, 1972)

Russel-Silver Syndrome (Angehrn et al, 1979; Escobar et al, 1978)

Short rib syndrome type 3 (Naumoff et al, 1977)

Shwachman Syndrome (McLennan & Steinbach, 1974)

Spondylo-epiphyseal Dysplasia with ocular changes (MacDessi et al, 1978)

Spondylocostal Dysostosis (Beighton & Horan, 1981)

Spondylometaphyseal Dysplasia (Heterogenous) (Kozlowski et al, 1980)

Spondyloperipheral Dysplasia (Kelly et al, 1977)

Stickler Syndrome (Opitz et al, 1972)

Taybi-Linder Syndrome (Taybi & Linder, 1967)

Trichorhinophalangeal Dysplasia (Felman & Frias, 1973)

Weaver Syndrome (Majewski et al, 1981; Weisswickert et al, 1981)

Weissenbacher-Zweymuller Syndrome (Kelly et al, 1982)

CHAPTER 5.4
ENTERING THE DISEASE PROFILES.

The method of entering disease profiles into the knowledge base was briefly referred to in subsection 4.4.1 but will now be presented in greater detail. At this stage it should be emphasised that no weighting is employed as yet. Manifestations are therefore not necessarily listed in order of importance, but rather according to their positions in the database. The criterion for the inclusion of a particular manifestation is whether it is considered important enough to serve as a differential diagnostic feature. The question of weighting as a future system enhancement will be discussed in section 8.

For descriptive purposes the Cranio-carpotarsal Dysplasia (CCTD) or Freeman-Sheldon syndrome will be used.

According to the Gamut Index, the following manifestations are present in this condition:

Clinical:

- Small pursed mouth
- Ulnar deviation of digits
- Inconsistent skeletal deformities.

Radiographic:

- Retarded bone age

Genetics:

- AD with variable expression.

This is an example of a condition whose profile, it was felt, could fruitfully be amplified from disease descriptions in the literature. The following clinical and radiological features that had been described for the Freeman-Sheldon syndrome (O'Connell and Hall, 1977), were therefore added.

- Microstomia (in lieu of small pursed mouth)
- Hypertelorism (subsequently removed)
- Low set ears
- Long philtrum
- Short mandible
- Brachycephaly
- Mental retardation (subsequently removed)
- Herniae (subsequently removed)
- Kyphoscoliosis
- Hip dislocation
- Vertical talus deformity
- Genu valgum
- Tall vertebral bodies
- Ulnar deviation of the fingers

The manifestations listed as subsequently removed in the list, were considered, after consultation with a domain expert (Beighton, 1987) to be unimportant. They were deleted by the method to be described in chapter 5-5.

Pages 5-i to 5-xv illustrate successive screens as they would appear on the computer terminal for the capture of information on CCTD. Explanatory notes follow, commencing on page 5-9, immediately after the subsidiary pages.

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN

Disease Name, max 50 chars: Cranio-Carpo-Tarsal Dysplasia

Synonym, max 50 chars: Freeman-Sheldon (Whistling Face) Syndrome

Standard Abbreviation, max 6 chars: CCTD

Genetics, max 50 chars: AD with very variable expression

Do you want to change?

Enter XXX for Name to abort process

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN

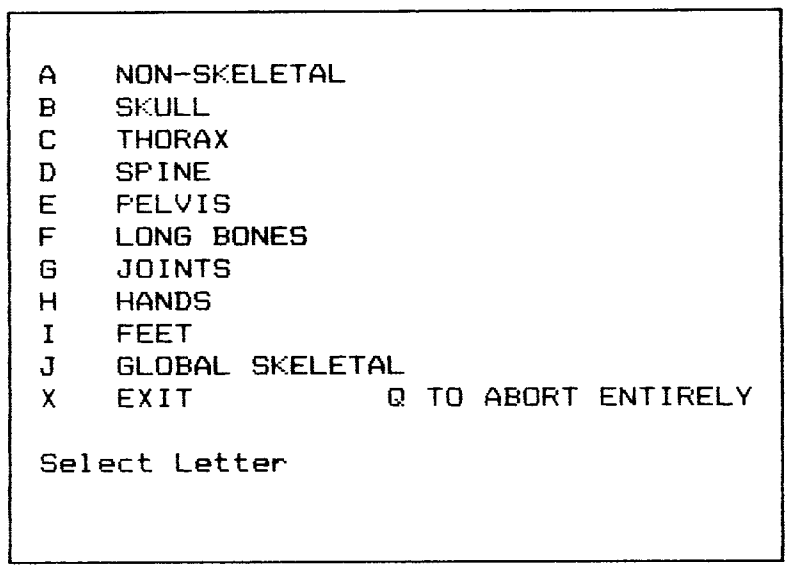
Freeman-Sheldon Syndrome

Synonym: Cranio-carpo-tarsal (whistling face) syndrome

Genetics: AD variable expression

This profile exists at record 91

Does it relate?


A NON-SKELETAL
B SKULL
C THORAX
D SPINE
E PELVIS
F LONG BONES
G JOINTS
H HANDS
I FEET
J GLOBAL SKELETAL
X EXIT Q TO ABORT ENTIRELY
Select Letter

Small stature/Dwarfism
» Face - Unusual/Characteristic
Face - Coarse features
» Mental retardation
Lethal chondrodysplasia
Eyes - Coloboma
Eyes - Microphthalmia
Eyes - Myopia/Hypermyopia
Eyes - Corneal clouding/opacity/cataract
Eyes - Non-specific defects
Eyes - Pseudoglioma
Eyes - Proptosis/Bulging
Eyes - Cherry-red macular spots
» Ears - Abnormal external ears
Ears - Deafness/impaired hearing
Vacant
Vacant
Mouth - Cleft lip
Mouth - Cleft palate
Mouth - Macrostomia

A<ccept> C<hange> D<own> E<xit> N<ext page> P<rev page> R<emove> U<p>

» Mouth - Microstomia
Mouth - Macroglossia
Mouth - Gingival hypertrophy
Vacant
Nose - Abnormal shape
Nose - Depressed nasal bridge
Nose - Broad nasal bridge
Philtrum - abnormal
Vacant
Lymphedema
Cardiac defect
Arterial abnormalities
Venous abnormalities
Hemangiomata - widespread
Arterio-venous fistula
Pulmonary defect
Abnormal genitalia/Urogenital defect/renal disease
Gastrointestinal defect
» Hernia
Abnormal skin/pigmentation

A<ccept> C<hange> D<own> E<xit> N<ext page> P<rev page> R<emove> U<p>

Macrocephaly
Microcephaly
Dyscephaly/Trigonocephaly
» Brachycephaly
Dolicocephaly
» Hypertelorism
Turricephaly
Calvaria - Osteomata
Calvaria - Poor ossification
Calvaria - Wide sutures
Calvaria - Sclerosis
Calvaria - Hyperostosis
Calvaria - Thickened
Calvaria - Coarse trabecular appearance
Calvaria - Late open fontanelles
Calvaria - Large/wide fontanelles
Calvaria - Small fontanelles
Frontal sinuses - Large
Frontal bones - Prominent/bossed
Supraorbital ridges - Prominent

Vacant
Vacant Large or wide fontanelles
Vacant Small fontanelles
Cranial sutures - Wormian bones
Cranial sutures - Premature fusion
Sella turcica - J-shaped
Craniostenosis
Maxilla/Midface - hypoplasia
Maxilla - prognathism
Mandible - cysts
» Mandible - hypoplasia/Micrognathia
Mandible - prognathism
Vacant
Vacant
Vacant
Mandible - Osteomata
Mandible - Hyperostosis
Mandible - Radiolucent defects
Mandible - Fibrous dysplasia
Mandible - Wide angle

A<ccept> C<hange> D<own> E<xit> N<ext page> P<rev page> R<emove> U<p>

```
  Spine - Degenerative changes (non-specific)
  Spine - Generalized structural abnormalities
» Spine - Generalized Scoliosis/ Kyphoscoliosis
  Vacant
  Vacant
  Vacant
  Cervical spine - Kyphosis
  Cervical spine - Lordosis
  Cervical spine - Hypoplastic vertebrae
  Cervical spine - Fused vertebrae
  Odontoid Process - Hypoplastic
  Vacant
  Vacant
  Vacant
  Thoracic spine - Hypoplastic vertebrae
  Thoracic spine - Accentuated kyphosis/gibbus
  Thoracic spine - Fused vertebrae
  Vacant
  Vacant
  Vacant
```

A<ccept> C<hange> D<own> E<xit> N<ext page> P<rev page> R<emove> U<p>

```
   Lumbar spine - Accentuated lordosis
   Lumbar spine - Incomplete ossification vertebrae
   Lumbar spine - Fused vertebrae
   Lumbar spine - Canal narrowed (stenosed)
   Lumbar spine - Interpedicular narrowing
   Lumbar spine - Spina bifida
   Vacant
   Vacant
   Vertebral bodies - Poor or absent ossification
   Vertebral bodies - Calcific stippling
   Vertebral bodies - Short
» Vertebral bodies - Tall
   Vertebral bodies - Ovoid
   Vertebral bodies - Hookshaped
   Vertebral bodies - Anterior wedging
   Vertebral bodies - Posterior wedging
   Vertebral bodies - Concave posterior surfaces
   Vertebral bodies - Concave anterior surfaces
   Vertebral bodies - Irregular endplates
   Vertebral bodies - Central indentations
```

A<ccept> C<hange> D<own> E<xit> N<ext page> P<rev page> R<emove> U<p>

Hip - Spurs of metaphyses
Hip - Premature fusion of metaphyses
Hip - Thin femoral neck
Vacant
Hip - Short femoral neck
Hip - Broad/thick femoral neck
Hip - Medial elongation of femoral neck
Hip - Absent femoral neck
Hip - Aseptic necrosis of femoral neck
Hip - Collapsed femoral head
» Hip - Dislocation
Hip - Contracture
Hip - Coxa valga
Hip - Coxa vara
Vacant
Vacant
Vacant
Knee - Delayed ossification of epiphyses
Knee - Large epiphyses
Knee - Small epiphyses

A<ccept> C<hange> D<own> E<xit> N<ext page> P<rev page> R<emove> U<p>

Knee - Flat epiphyses
Knee - Irregular epiphyses
Knee - Fragmented epiphyses
Knee - Sclerotic epiphyses
Knee - Cone-shaped epiphyses
Knee - Hat-shaped tibial epiphyses
Knee - Splayed metaphyses
Knee - Cupped metaphyses
Knee - Irregular margins of metaphyses
Knee - Sclerotic margins of metaphyses
Knee - Circumscribed ossification defects of metaphyses
Knee - Spurs of metaphyses
Knee - Premature fusion knee metaphyses
Knee - Absent/ hypoplastic patella
Knee - Dislocated patella
Knee - Granular calcifications of patella
Knee - Prominent medial plateau of tibia
Knee - Dislocation/subluxation
Knee - Contracture
» Knee - Genu valgum

A<ccept> C<hange> D<own> E<xit> N<ext page> P<rev page> R<emove> U<p>

Thumbs - Short proximally implanted
Thumbs - Short distal phalanx
Thumbs - Triphalangeal
First Metacarpal - Hypoplastic
First Metacarpal - Ovoid
Hypoplasia of first fingers (switch)
Fourth Metacarpals - Short
Fifth finger - Hypoplasia of middle phalanx
Fifth finger - Clinomicrodactyly
Simian crease
» Ulnar deviation of fingers
Vacant
Vacant

A<ccept> C<hange> D<own> E<xit> N<ext page> P<rev page> R<emove> U<p>

Feet - Absent
Feet - Equinovarus
Feet - Polydactyly
Feet - Syndactyly
Feet - Oligodactyly
Feet - Brachydactyly
Feet - Arachnodactyly
Feet - Macrodactyly
Vacant
Vacant
Vacant
Tarsal bones - Calcific stippling
Tarsal bones - Fusions
Tarsal bones - Osteolysis
Tarsal bones - Supernumerary
Tarsal bones - Delayed ossification
Tarsal bones - Malformed
Tarsal bones - Irregular overgrowth and calcification
» Tarsal bones - Vertical talus
Tarsal bones - Double ossification centre in calcaneum

A<ccept> C<hange> D<own> E<xit> N<ext page> P<rev page> R<emove> U<p>

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN

Cranio-Carpo-Tarsal Dysplasia

Freeman-Sheldon (Whistling Face) Syndrome

CCTD

AD with very variable expression

Disease entered into record 202

Press any key to continue...

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN

Freeman-Sheldon Syndrome

Cranio-carpo-tarsal (whistling face) syndrome

CCTD

Genetics: AD variable expression

Diagnosis essentially clinical.

See O'Connell and Hall (1977) Cranio-carpotarsal dysplasia.

A report of 7 cases. Radiology 123: 719-723.
Press any key to continue...

The following manifestations have been described:
(Not necessarily in order of importance!)

Press any key to continue...

Face - Unusual/Characteristic
Mental retardation
Ears - Abnormal external ears
Mouth - Microstomia
Philtrum - abnormal
Hernia
Brachycephaly
Hypertelorism
Mandible - hypoplasia/Micrognathia
Spine - Generalized Scoliosis/ Kyphoscoliosis
Vertebral bodies - Tall
Hip - Dislocation
Knee - Genu valgum
Ulnar deviation of fingers
Tarsal bones - Vertical talus
Retarded bone age - Generalized

N<ext> P<rev page> <e>X<it>

On the "pro forma" depicted on p5-i, the disease name, an optional synonym, a possible abbreviation and a terse genetic description are entered. An opportunity to alter any of these items is offered on the last line with "Do you want to change?" Responding with "n" or "N" will terminate the entry of this part of the data. A search for the possible presence of the condition in the database is then conducted. Should the particular disorder already exist in the knowledge base, the screen on p5-ii will appear. An affirmative response to the question "Does it relate?", will terminate the data entry process forthwith.

Should the disorder in question not be present in the database, the screen shown on p5-iii will appear. It presents a menu of sections from which manifestations may be selected for inclusion in the disease profile.

On p5-iv the first list of features, from the Non-skeletal section, appears after selecting "A" from the menu of options. Stigmata, appropriate to the disease concerned, are selected by moving the screen cursor to the particular items by means of the "U" (up) and "D" (down) keys on the computer keyboard. "A" (accept) will select a particular feature, "R" (remove) will "unselect" a manifestation (erroneously selected for instance), "N" (next) will bring up the next page of items on to the screen and "P" (previous) will list the previous page. "E" will effect an exit from the particular section and return the operator to the menu of p5-iii. The small arrows indicate which features have been selected from a particular page. On p5-iv, for example, the following have been chosen:

Unusual/characteristic facies

Mental retardation

Ears - Abnormal external ears.

Page 5-v shows a further list from the non-skeletal section with the features:

Mouth - Microstomia

Hernia

having been selected.

Should a "C" (change) be picked from the options at the bottom of the page, opportunity will be given to change the description of a selected item. As was pointed out in chapter 4.2 this facility should be exercised with caution as indifferent manipulation here could seriously jeopardise the integrity of the knowledge base. It is therefore password controlled. It is also possible, by means of this facility, to enter a new manifestation by editing a "vacant" entry. This too is password controlled.

Selecting "B" from the menu on p5-iii will produce the first page from the skull section. This is shown on p5-vi. The features

Brachycephaly

Hypertelorism

have been selected.

A further page from the skull section is found on p5-vii, enabling the operator to select "Mandibular hypoplasia/Micrognathia".

Choosing "D" from the menu on p5-iii will access the section of spinal features. On p5-viii the entry "Generalized scoliosis/Kyphoscoliosis" is selected and on p5-ix, the item "Tall vertebral bodies" is chosen from another page in the same section. Page 5-x illustrates the selection

of "Dislocation of the hip" from the "Joints" section. On p5-xi "Genu valgum" is obtained from a different part of the same section. Page 5-xii shows a list from the section "Hands". "Ulnar deviation of fingers" is selected here and "Vertical talus" from a page in section "Feet" (p5-xiii). "Retarded bone age" is obtained in a similar fashion from the "Global" section.

This completes the selection of clinical and radiographic features. The process is terminated by selecting "X" (exit) from the menu (p5-iii). The computer response shown on p5-xiv indicates that the disease profile has been entered into record no. 202 of the file "Profiles.ran".

Page 5-xv shows the response of program "Testlook.pas" to a request to inspect the disease profile at record number 202. It illustrates that the disease Cranio-Carpotarsal Dysplasia has been successfully entered into the knowledge base. It also shows how the profile will be presented with future retrieval requests after the addition of some notes.

CHAPTER 5-5
CHANGING A DISEASE PROFILE.

Pages 5-xvi to 5-xxi illustrate how alterations to disease descriptions that became necessary during the implementation stage, were effected. Explanatory notes follow on page 5-11.

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN

Enter Password

Enter Disease Name: Freeman-Sheldon

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN

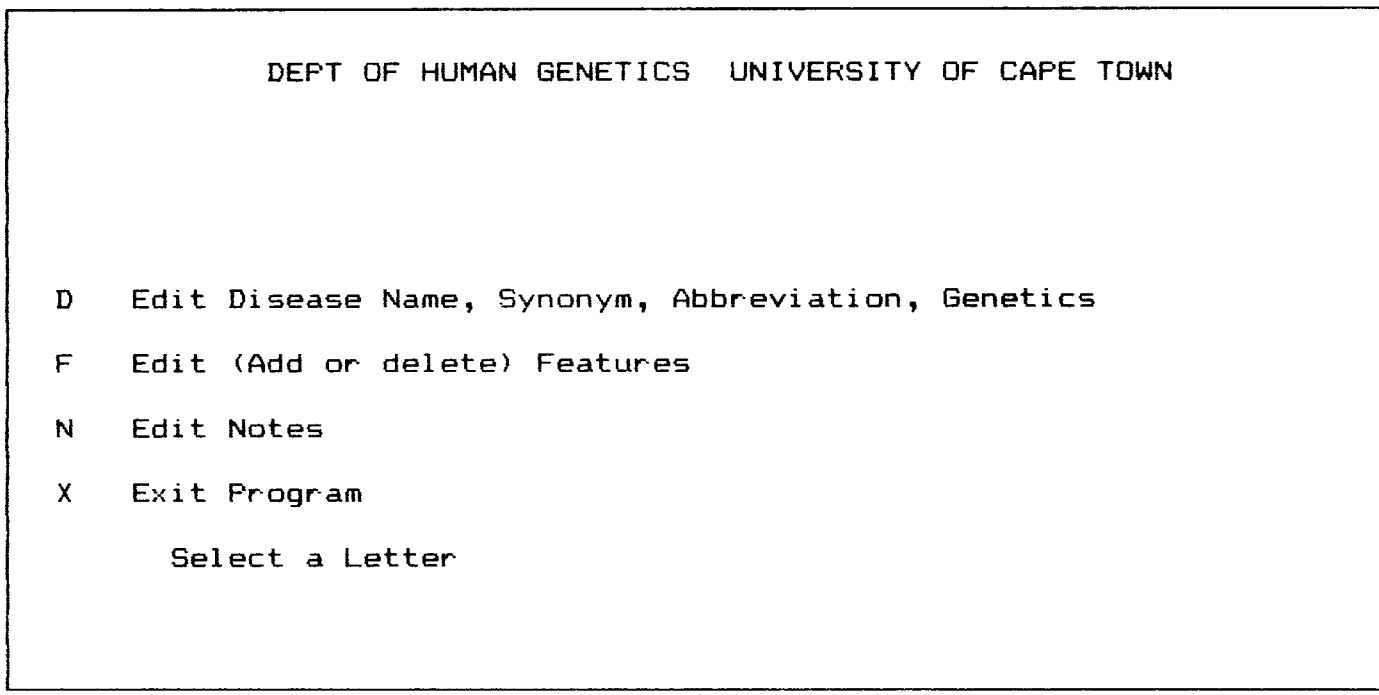
Freeman-Sheldon Syndrome

Synonym: Cranio-carpo-tarsal (whistling face) syndrome

Genetics: AD variable expression

This profile exists at record 91

Does it relate?


DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN
D Edit Disease Name, Synonym, Abbreviation, Genetics
F Edit (Add or delete) Features
N Edit Notes
X Exit Program
Select a Letter

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN

Disease Name, max 50 chars: Freeman-Sheldon Syndrome

Synonym, max 50 chars: Cranio-carpo-tarsal (whistling face) syndrome

Standard Abbreviation, max 6 chars: CCTD

Genetics, max 50 chars: AD variable expression

Do you want to change?

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN

Disease Name, max 50 chars: Cranio-carpo-tarsal Syndrome

Synonym, max 50 chars: Freeman-Sheldon (whistling face) syndrome

Standard Abbreviation, max 6 chars: CCTD

Genetics, max 50 chars: AD variable expression

Do you want to change?

IN EDIT MODE <Return> ends editing of particular string

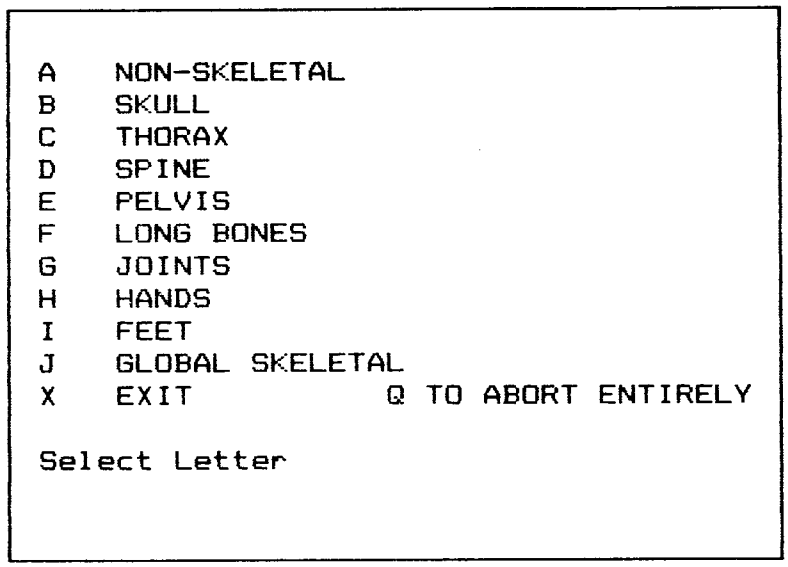
> moves cursor right, < moves left, # deletes char

 1 Face - Unusual/Characteristic
 2 Mental retardation
 3 Ears - Abnormal external ears
 4 Mouth - Microstomia
 5 Philtrum - abnormal
 6 Hernia
 7 Brachycephaly
 8 Hypertelorism
 9 Mandible - hypoplasia/Micrognathia
10 Spine - Generalized Scoliosis/ Kyphoscoliosis
11 Vertebral bodies - Tall
12 Hip - Dislocation
13 Knee - Genu valgum
14 Ulnar deviation of fingers
15 Tarsal bones - Vertical talus
16 Retarded bone age - Generalized

<A>dd <D>elete <N>ext page <P>rev page e<X>it <Q>uit

 1 Face - Unusual/Characteristic
 2 Ears - Abnormal external ears
 3 Mouth - Microstomia
 4 Philtrum - abnormal
 5 Brachycephaly
 6 Mandible - hypoplasia/Micrognathia
 7 Spine - Generalized Scoliosis/ Kyphoscoliosis
 8 Vertebral bodies - Tall
 9 Hip - Dislocation
10 Ulnar deviation of fingers
11 Tarsal bones - Vertical talus
12 Retarded bone age - Generalized

<A>dd <D>elete <N>ext page <P>rev page e<X>it <Q>uit



Knee - Flat epiphyses
Knee - Irregular epiphyses
Knee - Fragmented epiphyses
Knee - Sclerotic epiphyses
Knee - Cone-shaped epiphyses
Knee - Hat-shaped tibial epiphyses
Knee - Splayed metaphyses
Knee - Cupped metaphyses
Knee - Irregular margins of metaphyses
Knee - Sclerotic margins of metaphyses
Knee - Circumscribed ossification defects of metaphyses
Knee - Spurs of metaphyses
Knee - Premature fusion knee metaphyses
Knee - Absent/ hypoplastic patella
Knee - Dislocated patella
Knee - Granular calcifications of patella
Knee - Prominent medial plateau of tibia
Knee - Dislocation/subluxation
Knee - Contracture
» Knee - Genu valgum

A<ccept> C<hange> D<own> E<xit> N<ext page> P<rev page> R<emove> U<p>

```
 1 Face - Unusual/Characteristic
 2 Ears - Abnormal external ears
 3 Mouth - Microstomia
 4 Philtrum - abnormal
 5 Brachycephaly
 6 Mandible - hypoplasia/Micrognathia
 7 Spine - Generalized Scoliosis/ Kyphoscoliosis
 8 Vertebral bodies - Tall
 9 Hip - Dislocation
10 Knee - Genu valgum
11 Ulnar deviation of fingers
12 Tarsal bones - Vertical talus
13 Retarded bone age - Generalized
```

<A>dd <D>elete <N>ext page <P>rev page e<X>it <Q>uit

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN

Edit complete, returning to menu
Press any key to continue...

Changes to a disease profile may impair the integrity of the database. A password is therefore requested by the program as shown on p5-xvi. If the password is accepted, a prompt for the disease name appears. In the example it is shown that the complete name need not be entered. Indeed, any part of the disease name or synonym may be used. The search algorithm uses the commencing half of the supplied character string, therefore disease names or synonyms containing the string "freeman" will be looked for. The shorter the provided string, the less selective the search will be. This is often desirable however, as it is better to be supplied with redundant names which may be skipped if necessary, than to miss a disease with a long name as a result of, for instance, misspelling. The search program converts all lower case characters in the search string to upper case, so that care need not be taken to enter the capital letters normally required with proper names.

The text at the bottom of p5-xvi indicates that a matching name happens to be present in record 91. An affirmative answer to "Does it relate?" will produce the screen on p5-xvii which illustrates the various options available under the editing facility. Selecting "D" (disease name) will result in the screen shown at the top of p5-xviii. The bottom diagram on the same page shows changes made to Disease name and Synonym by means of the editing facilities provided. In the example, the disease name and synonym were interchanged by character deletion and retyping.

Selecting "F" from the list on p5-xvii will allow for the addition or deletion of disease manifestations. Initially a table of existing stigmata is presented as shown by the first table on p5-xix. By selecting the "D" (delete) option, the operator is requested (not shown) to enter the number next to the feature for deletion. As an example, number 13, Genu valgum, is selected, together with those manifestations marked on page 5-11 as subsequently removed. The lower table on sub-page 5-xix, in which the mentioned stigmata are absent, is consequently produced. To illustrate how features may be added to the disease profile, Genu valgum will be reinstated in the profile. In a manner similar to that used with the entry of a disease profile, the appropriate section is selected from the table on p5-xx (top). The desired feature is located and marked as shown at the bottom of p5-xx. Genu valgum reappears in the list reproduced at the top of p5-xxi. New features may be added in similar fashion. When selection is complete, "X" from the menu on p5-xx will return the user to the editing menu.

Changing the notes involves trivial text manipulation using the editing facilities in a way similar to those employed with the name, synonym and genetics. It will therefore not be expounded.

It would appear that a fair knowledge of the workings of a microcomputer is required during implementation of the knowledge base and programs. To facilitate use of the programs by "non-computerminded" users, an interface was devised. This will be outlined in the next section.

SECTION SIX.

THE USER INTERFACE

CHAPTER 6.1
INTRODUCTION.

In the previous section it was shown how the knowledge base was implemented by means of the programs Mkdescrpt.pas, Features.pas, Disenter.pas and Profedit.pas. Two other programs, Testlook.pas and Analyse.pas, the listings of which appear in appendix A, were employed in subsidiary roles during the implementation process. The former was needed to inspect disease profiles from time to time. Analyse.pas yielded valuable information on the length of profile bytes, the number of times the various manifestations appear in disease profiles, and also provided complete lists of diseases present in the knowledge base when such were needed.

These programs were not intended for the ultimate users of the application. They were conceived and written for use by the author during the design, implementation and testing stages of the project. In this section it will be shown how a "user interface" was established by means of modifications to existing programs and the addition of others designed for the easy retrieval of information.

Turbo-Pascal allows for the execution of programs in two ways. With the first method, the source program (with extension .Pas) is executed with the "run" command in a way similar to that employed by the interpretive languages (eg. most versions of BASIC on microcomputers). The programs are not interpreted, however, but rather compiled very rapidly and immediately executed. This method was employed by the author during the implementation phase, since a single program was involved at any one time. Alterations to the program could be effected with ease when the need arose. This approach is not suitable in a situation where one program, after completion, initiates another one to execute, as would be the case where a "menu" is involved.

In order to establish a user interface where the operator could select program facilities at will, and always return to the "menu", the second approach was followed. All the application programs are compiled to a final form with this method. A main controlling program can call upon any subordinate program to execute and, on completion, the latter invokes the former again. In Turbo Pascal parlance, the main program (with the extension .Com) "chains" the subprogram (extension .Chn). When completed, the subprogram "executes" the main program again. "Chain" and "Execute" are both syntactical terms used by Turbo Pascal (ref Turbo Pascal Version 3.0 Reference Manual).

CHAPTER 6.2
THE USER APPLICATION PROGRAMS.

6.2.1 The Menu Program.

Fig 6-1 on subsidiary page 6-i shows the screen produced by the main (although small) controlling program. A listing of

this program is shown as Menu.pas in appendix A. It was, as explained in the previous chapter, compiled to become the executable program Menu.Com.

The Menu program can be started by selecting the directory on the hard disk where the application programs reside with the "chdir" (for change directory) utility of MSDOS, and entering the command "Menu" on the keyboard. Alternatively, users may elect to start by switching the computer on after inserting a dedicated floppy disk into the A-drive of the machine. The flexible disk contains a file AUTOEXEC.BAT with a batch of instructions which will automatically start executing, select the appropriate directory on the fixed disk and initiate the menu program. The listing of the autoexec file is given in appendix B.

Fig 6-1 illustrates that four options are available to the user for interaction with the knowledge base. The fifth option effects an exit from the application programs and will return the user to the disk operating system. The various options accomplish chaining of the following compiled programs:

- Option 1 Disenter.Chn
- Option 2 Profedit.Chn
- Option 3 Dsearch.Chn
- Option 4 Prospect.Chn

Each of these will, on completion, "execute" Menu.Com again.

6.2.2 Entering a disease profile with Disenter.Chn.

This program operates exactly as outlined in chapter 5.4 where Disenter.Pas is described. It will not be detailed again. A listing of Disenter.Pas, which is the source program for Disenter.Chn, appears in appendix A.

6.2.3 Editing a disease profile with Profedit.Chn.

The mode of operation of this program is identical to that outlined in chapter 5.4. The listing of Profedit.Pas is included in appendix A.

6.2.4 Retrieving a disease profile using Dsearch.Chn.

The listing of Dsearch.Pas, the source file for Dsearch.Chn, is given in appendix A. Pages 6-i to 6-v show successive screens produced by this program. For the sake of variety another condition, Mucolipidosis I, is used as an illustration. Explanatory notes follow on page 6-3.

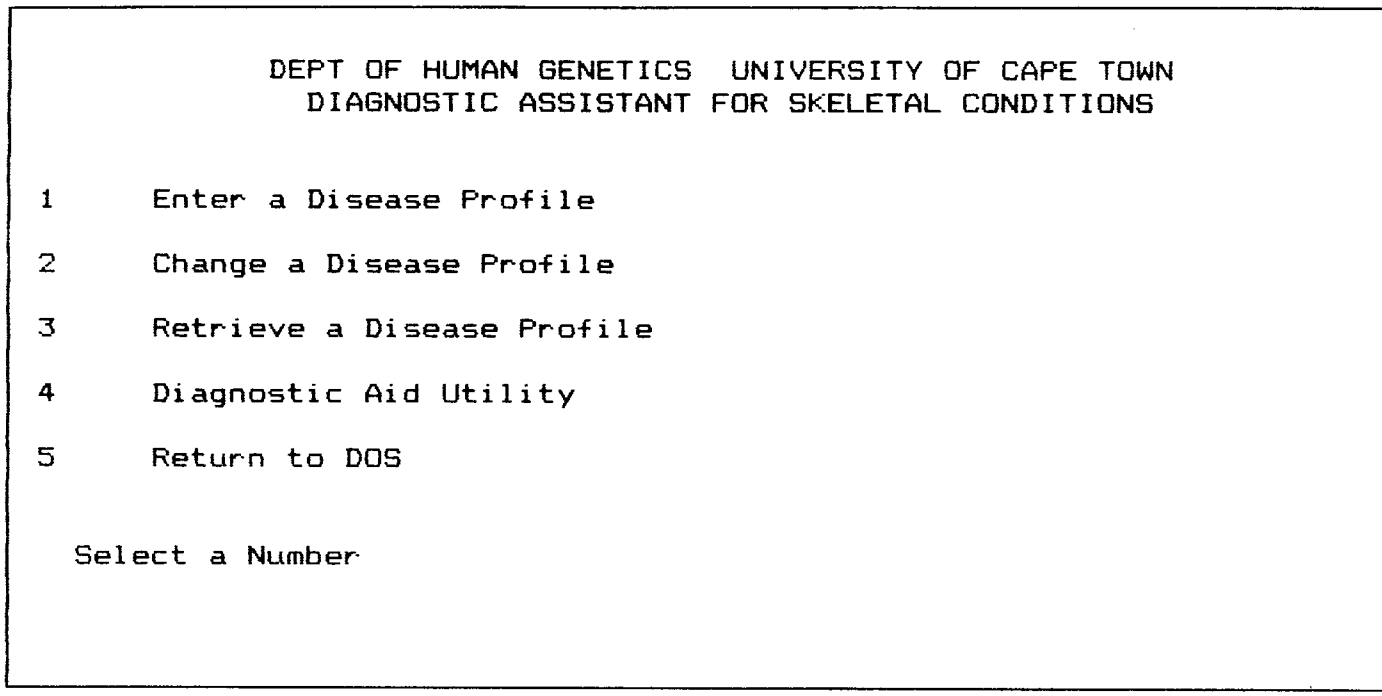


Fig. 6-1 Main Application Menu

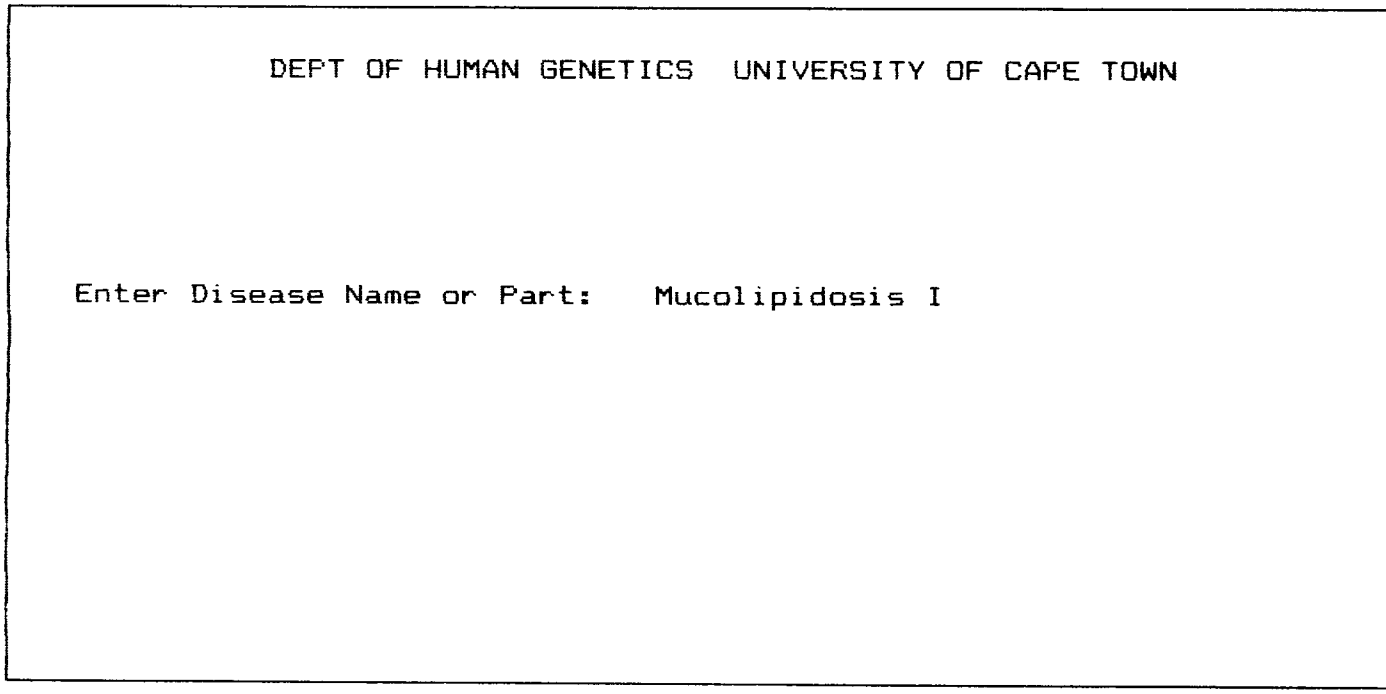


Fig 6-2

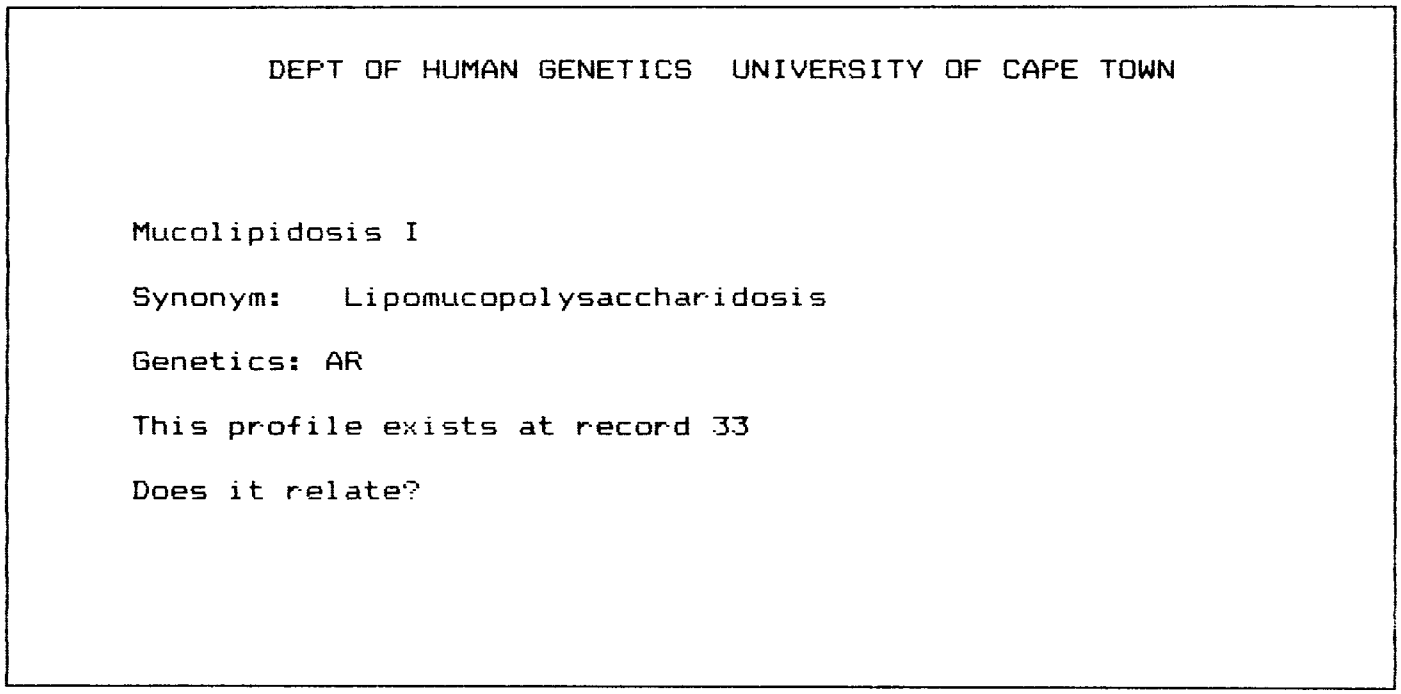

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN
Mucolipidosis I
Synonym: Lipomucopolysaccharidosis
Genetics: AR
This profile exists at record 33
Does it relate?

Fig 6-3

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN

Mucolipidosis I

Lipomucopolysaccharidosis

MLSI

Genetics: AR

Spranger and Wiedemann (1970): The genetic mucolipidoses

Humangenetik 9,113

Slowly progressive with increasing Hurler-like physical changes
Press any key to continue...

Fig 6-4

The following manifestations have been described:
(Not necessarily in order of importance!)

Press any key to continue...

Face - Coarse features
Mental retardation
Eyes - Non-specific defects
Eyes - Cherry-red macular spots
Hernia
Metabolic abnormality - heteroglycan
Metabolic abnormality - lipid
Hepatic disease/enlargement
Splenomegaly
CNS defect (excluding cranial nerves)
Muscle - Wasting/reduced mass
Calvaria - Thickened
Mandible - prognathism
Mandible - Wide angle
Odontoid Process - Hypoplastic
Vertebral bodies - Ovoid
Acetabula - Deficient ossification of roof
Ilia - Absent/hypoplastic
Long bones - Fractures
Hip - Irregular epiphyses

MORE ON NEXT PAGE...
N<ext> P<rev page> <e>X<it>

Hip - Coxa valga
Metacarpals - Wide
Hand phalanges - Wide
Osteoporosis - Generalized
Dysostosis multiplex - Moderate

Fig 6-2 on p6-i illustrates how the disease name or synonym is supplied to the program. As before, only part of the name need to be entered, but with a corresponding reduction in selectivity. For instance, entering "mucolipid" will result in all conditions commencing with "muco" (first half of the search string) to be announced. Fig 6-3 on p6-ii shows how a find is reported on the screen.

When the desired condition is located, fig 6-4 on p6-iii will appear. This is followed by the lists of clinical and radiological features shown on p6-iv and p6-v. The options at the bottom of the lists allow the operator to go from page to page at will when dealing with lengthy lists. The "X" option will effect a return to the main menu.

6.2.5 Comparing disease profiles with Prospect.Chn.

Program Prospect.Pas, the source for Prospect.Chn, is listed in appendix A.

The facility may be used in two ways:

- 1) To provide a list of those conditions which manifest a particular diagnostic feature or combination of features;

- 2) To conduct a differential diagnosis against a number of selected manifestations as may be the case when an undiagnosed case is being investigated.

The first option whereby a list of disorders, manifesting a certain feature or combination of features, is provided, is illustrated on subsidiary pages 6-vi and 6-vii. Explanatory notes follow on page 6-4.

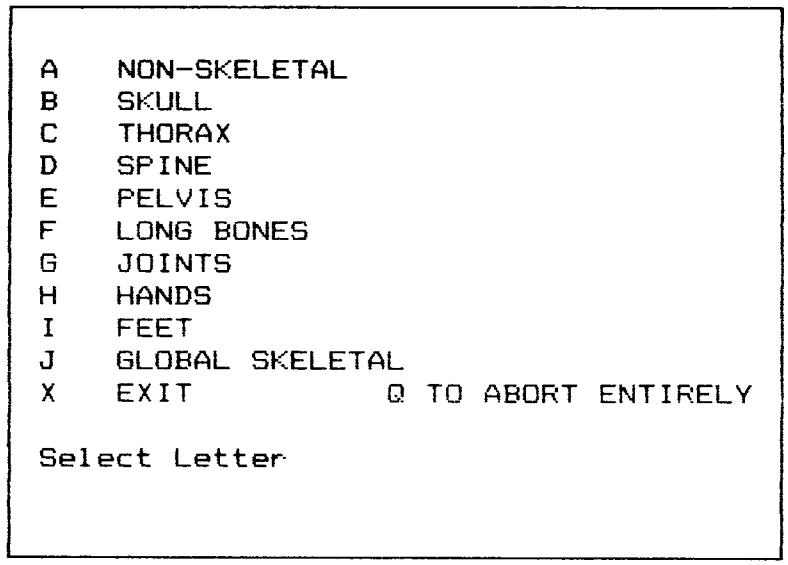


Fig 6-5 Sections Menu

```
  Vacant
  Vacant Large or wide fontanelles
  Vacant Small fontanelles
» Cranial sutures - Wormian bones
  Cranial sutures - Premature fusion
  Sella turcica - J-shaped
  Craniostenosis
  Maxilla/Midface - hypoplasia
  Maxilla - prognathism
  Mandible - cysts
  Mandible - hypoplasia/Micrognathia
  Mandible - prognathism
  Vacant
  Vacant
  Vacant
  Mandible - Osteomata
  Mandible - Hyperostosis
  Mandible - Radiolucent defects
  Mandible - Fibrous dysplasia
  Mandible - Wide angle
```

A<ccept> C<hange> D<own> E<xit> N<ext page> P<rev page> R<emove> U<p>

Fig 6-6

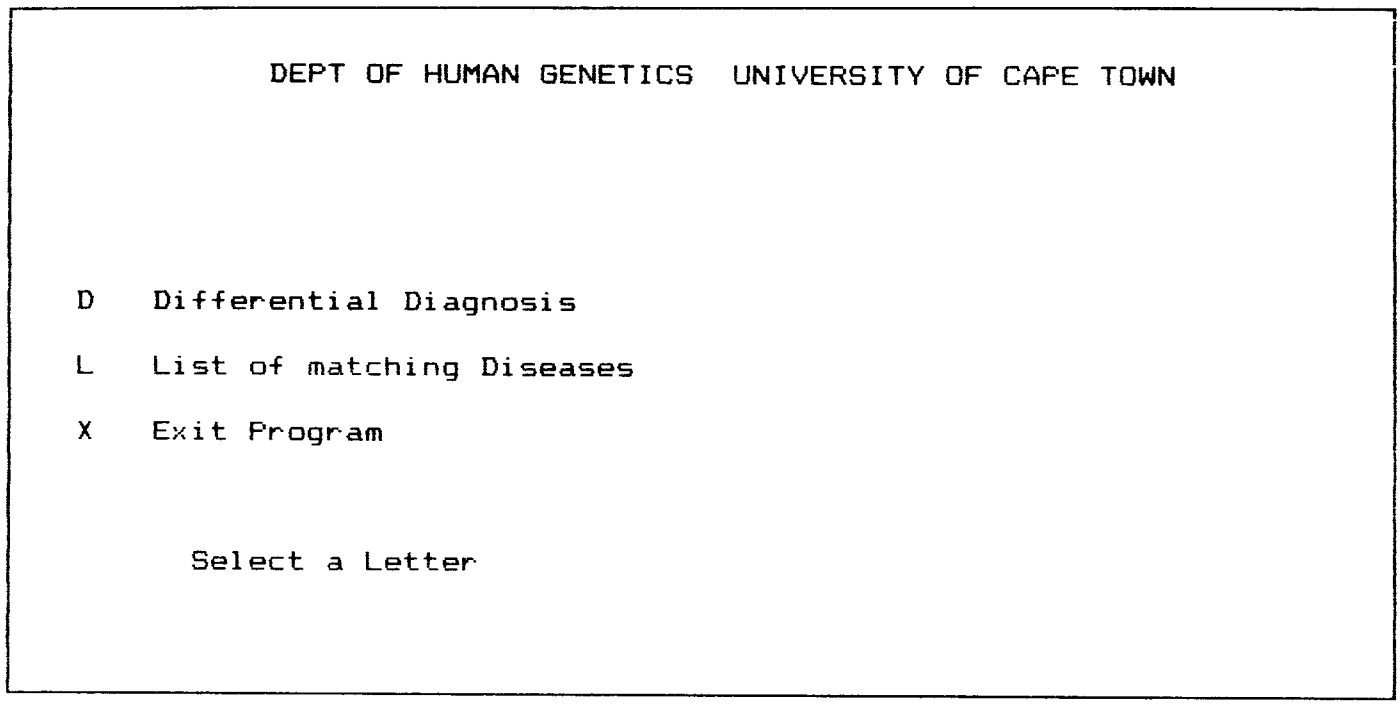


Fig 6-7 Diagnostic Program - Options Menu

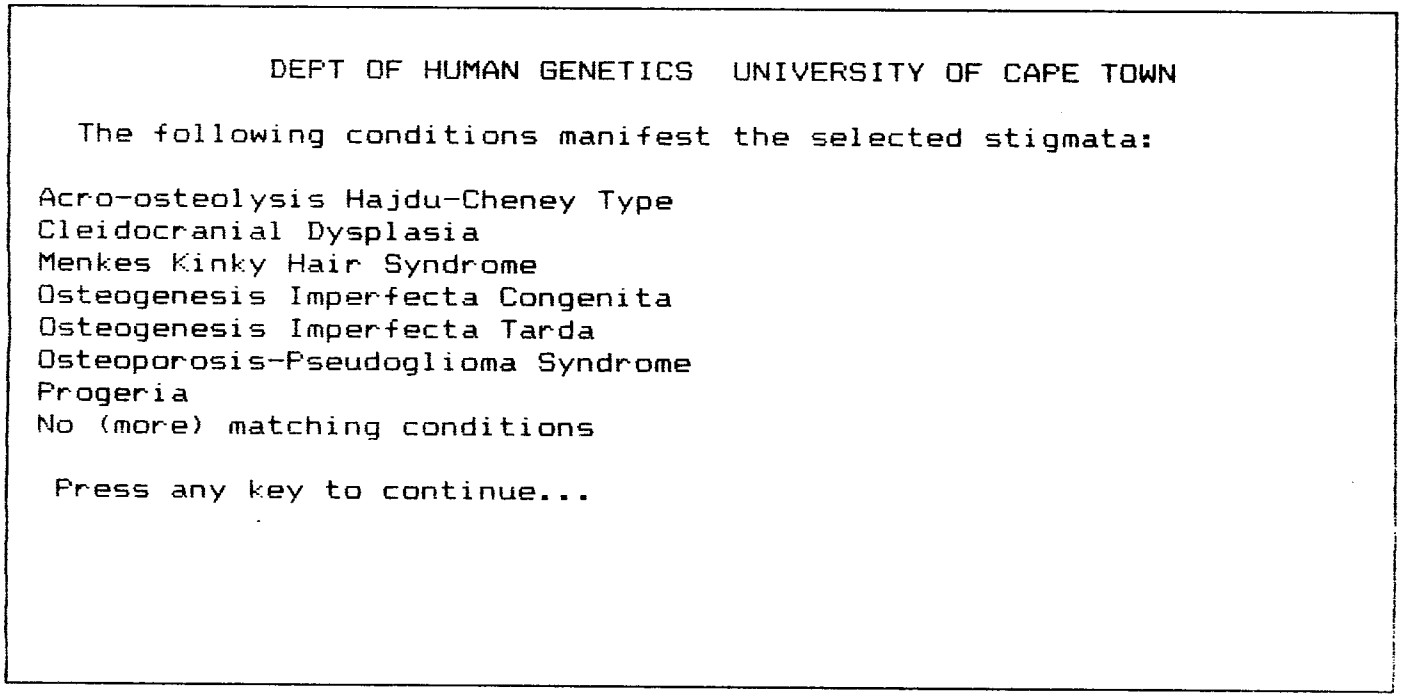


Fig 6-8 Response to L-option

On page 6-vi the manifestation "wormian bones" is selected from the features list and the "L" option chosen from the menu shown at the top of p6-vii. The list seen at the bottom of p6-vii is produced with great rapidity on the computer screen. It contains those conditions which are known to the knowledge base at the particular time to manifest the chosen feature or combination of features.

The second option, depicted by "D" on the menu illustrated on subsidiary page 6-vii, will lead the operator through a differential diagnosis. This is outlined on subsidiary pages 6- viii to 6-xx. An explanation of the protocol is given after the subsidiary pages and commencing on page 6-5.

```
» Small stature/Dwarfism
  Face - Unusual/Characteristic
  Face - Coarse features
  Mental retardation
  Lethal chondrodysplasia
  Eyes - Coloboma
  Eyes - Microphthalmia
  Eyes - Myopia/Hypermyopia
  Eyes - Corneal clouding/opacity/cataract
  Eyes - Non-specific defects
  Eyes - Pseudoglioma
  Eyes - Proptosis/Bulging
  Eyes - Cherry-red macular spots
  Ears - Abnormal external ears
  Ears - Deafness/impaired hearing
  Vacant
  Vacant
  Mouth - Cleft lip
  Mouth - Cleft palate
  Mouth - Macrostomia

A<ccept>  C<hange>  D<own>  E<xit>  N<ext page>  P<rev page>  R<emove>  U<p>
```

Fig 6-9 Selecting Dwarfism

```
  Macrocephaly
  Microcephaly
  Dyscephaly/Trigonocephaly
  Brachycephaly
  Dolicocephaly
  Hypertelorism
  Turricephaly
  Calvaria - Osteomata
  Calvaria - Poor ossification
  Calvaria - Wide sutures
  Calvaria - Sclerosis
  Calvaria - Hyperostosis
  Calvaria - Thickened
  Calvaria - Coarse trabecular appearance
» Calvaria - Late open fontanelles
  Calvaria - Large/wide fontanelles
  Calvaria - Small fontanelles
  Frontal sinuses - Large
  Frontal bones - Prominent/bossed
  Supraorbital ridges - Prominent

A<ccept>  C<hange>  D<own>  E<xit>  N<ext page>  P<rev page>  R<emove>  U<p>
```

Fig 6-10 Selecting late open Fontanelles

```
    Long bones - Non-specific dysplasia
    Long bones - Exostoses
    Long bones - Enchondromata
    Long bones - Generalised bowing
    Long bones - Calcific stippling
    Long bones - Lytic lesions
    Long bones - Osteolysis
    Long bones - Excessive new bone formation
    Long bones - Short shafts
    Long bones - Thin shafts (Hypertubulation)
    Long bones - Wide shafts (Hypotubulation)
    Long bones - Submetaphyseal undermodelling
    Long bones - Overconstriction
» Long bones - Fractures
    Long bones - Osteopenia of whole shaft
    Long bones - Submetaphyseal osteopenia
    Long bones - Diffuse sclerosis of all
    Long bones - Diffuse sclerosis of some
    Long bones - Diffuse sclerosis of parts
    Long bones - Striation


A<ccept>  C<hange>  D<own>  E<xit>  N<ext page>  P<rev page>  R<emove>  U<p>
```

Fig 6-11 Selecting Long Bone Fracturing

```
    Hyperostosis/New bone formation - Generalized
» Fractures - Widespread (tubular and flat bones)
    Enchondromata - Widespread involvement including flat bones

A<ccept>  C<hange>  D<own>  E<xit>  N<ext page>  P<rev page>  R<emove>  U<p>
```

Fig 6-12 Selecting widespread Fracturing

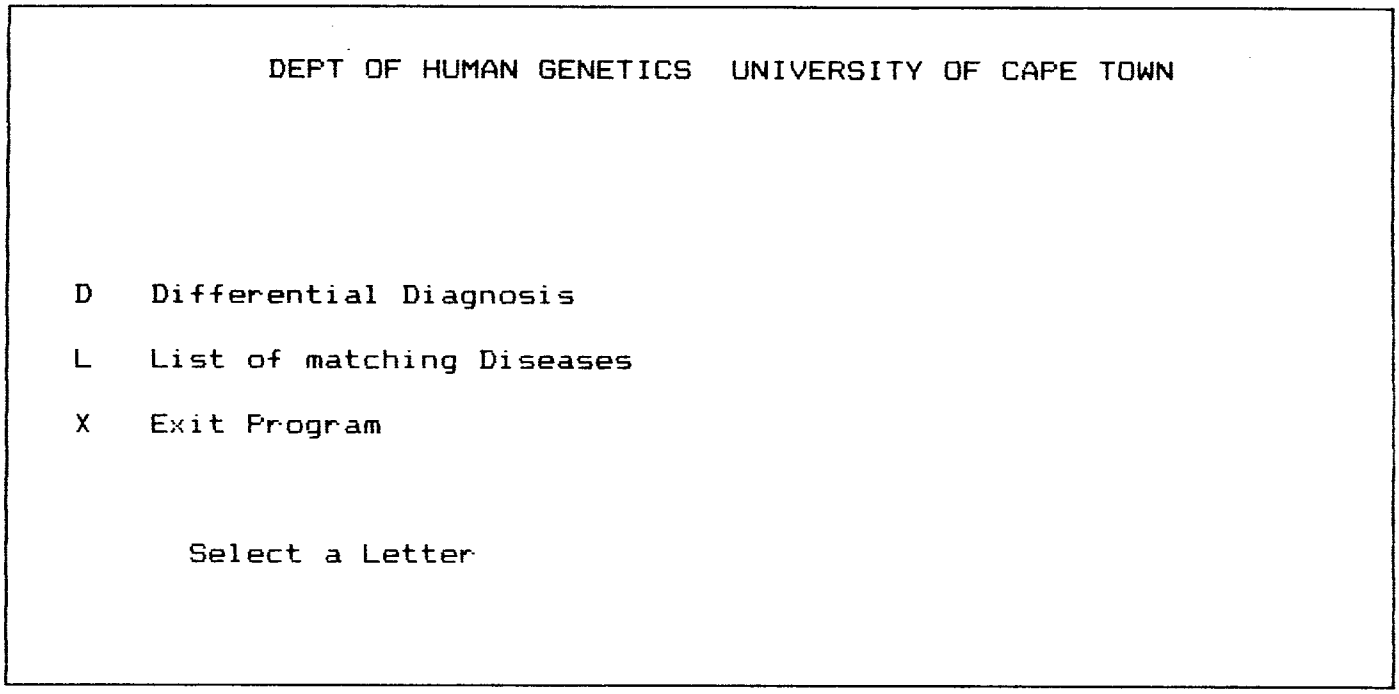


Fig 6-13 Options Menu Repeated for Convenience

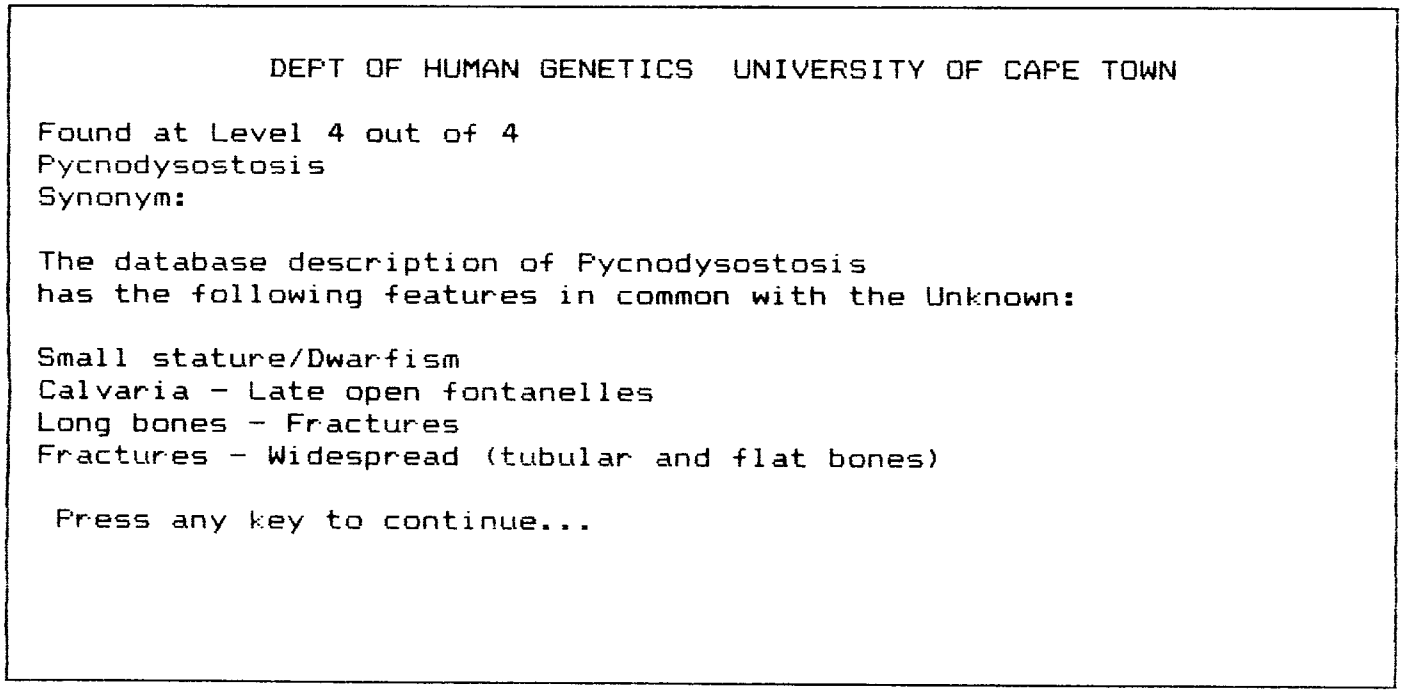


Fig 6-14 Announcing first Find

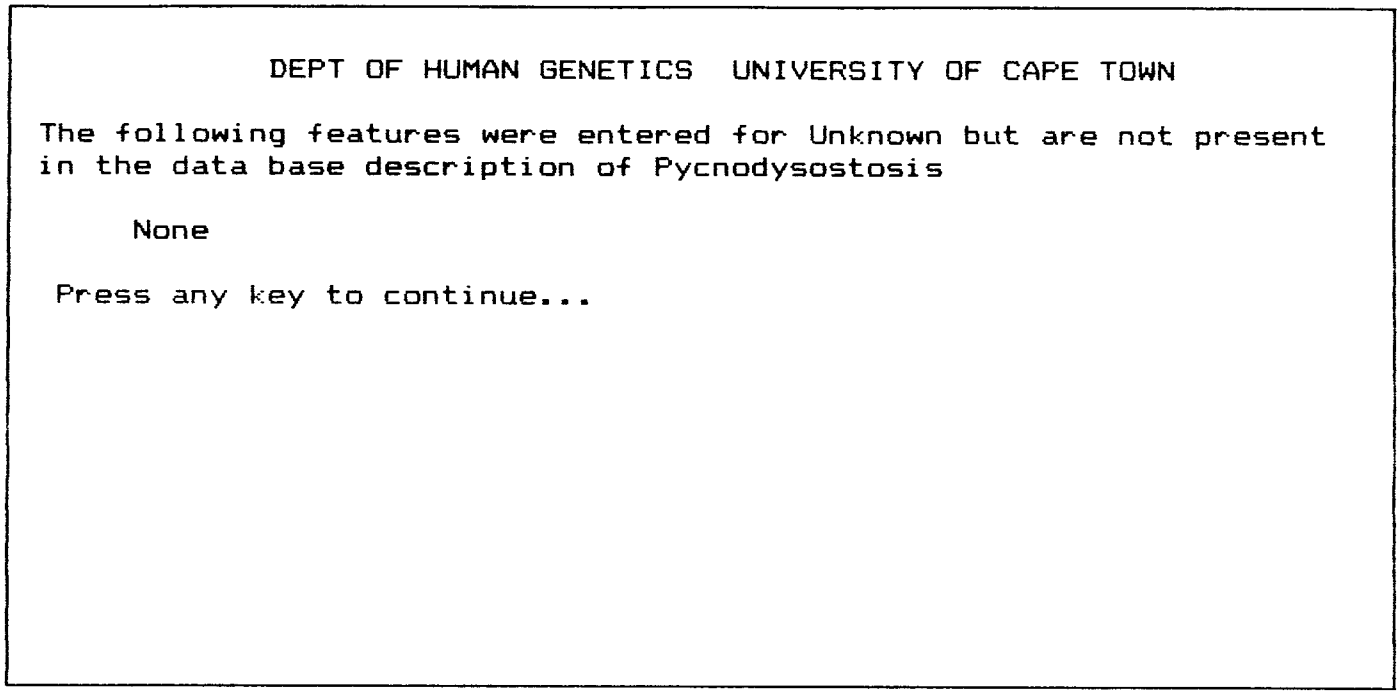


Fig 6-15

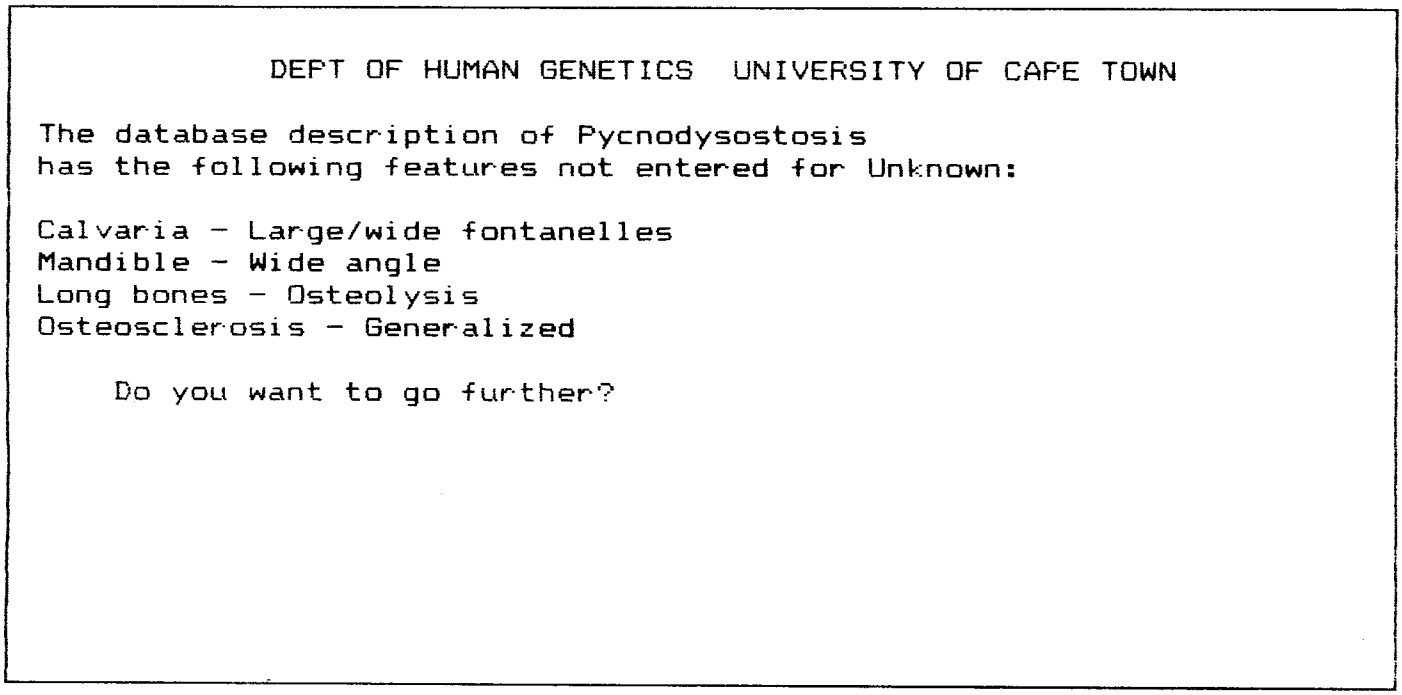


Fig 6-16

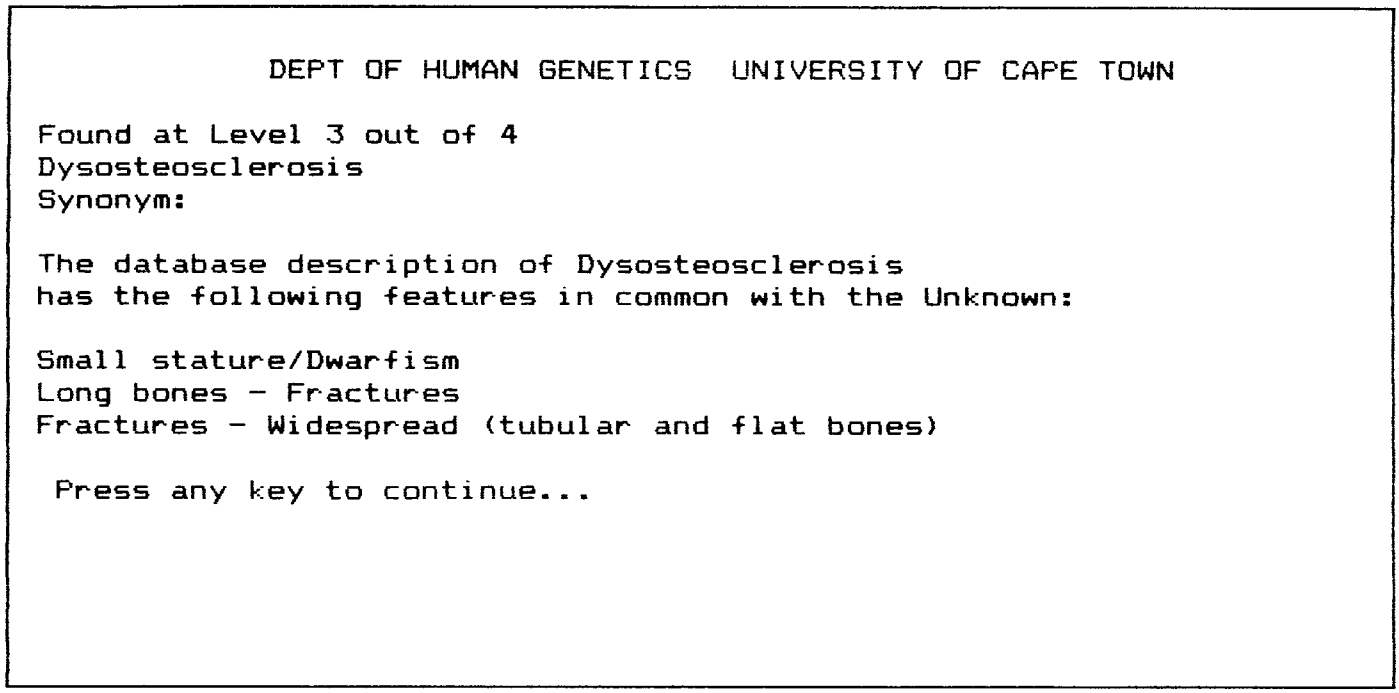


Fig 6-17 Second Find

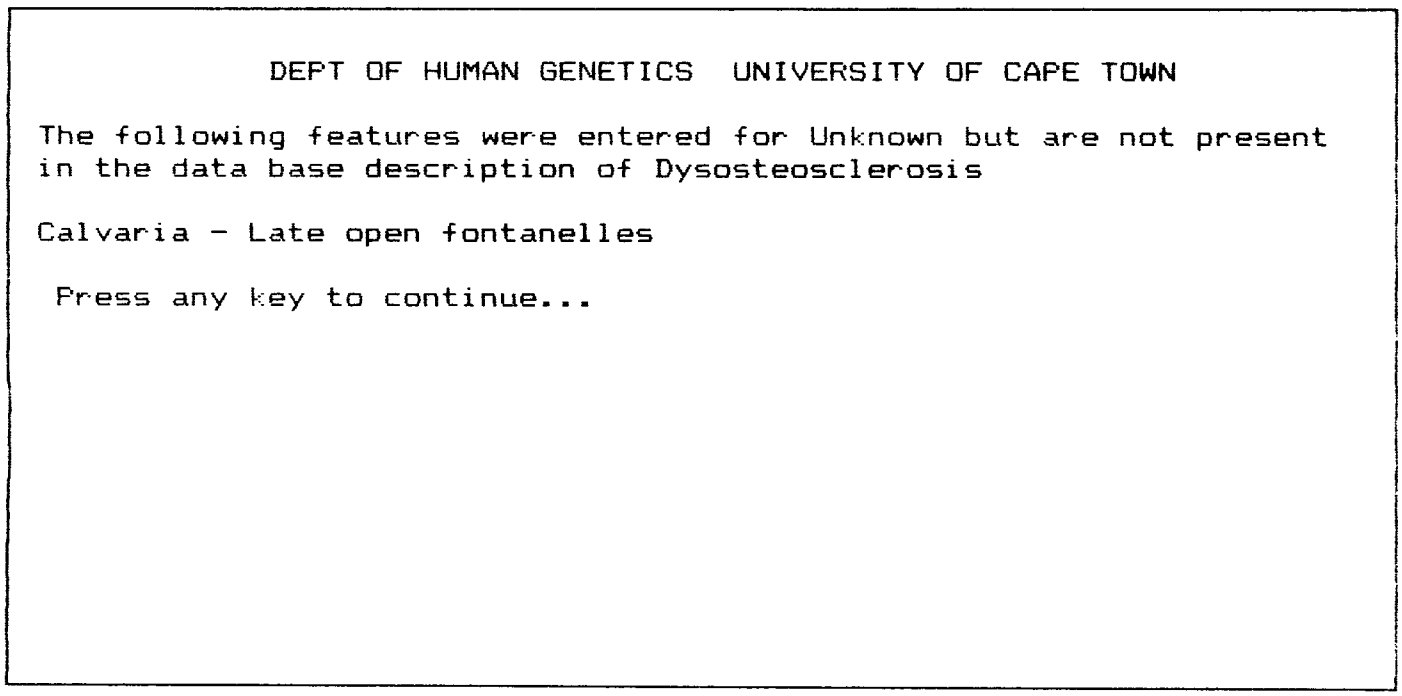


Fig 6-18 See Text

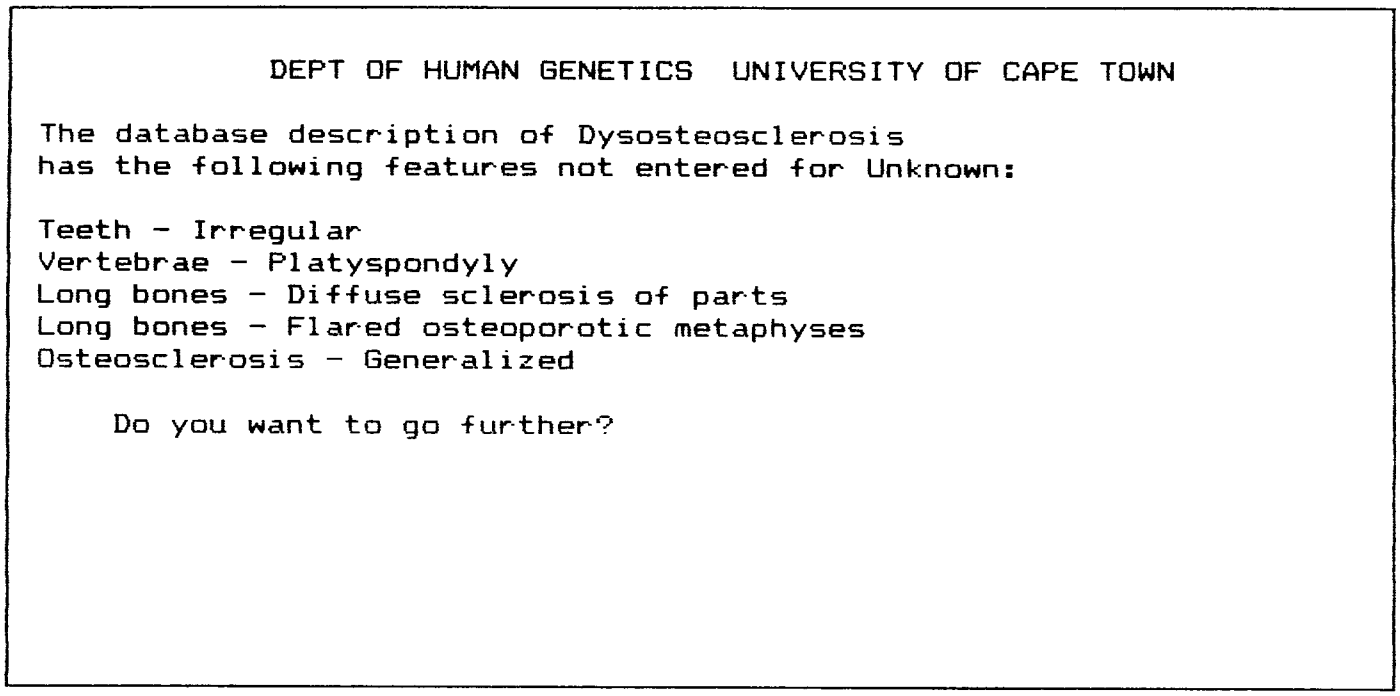


Fig 6-19 See Text

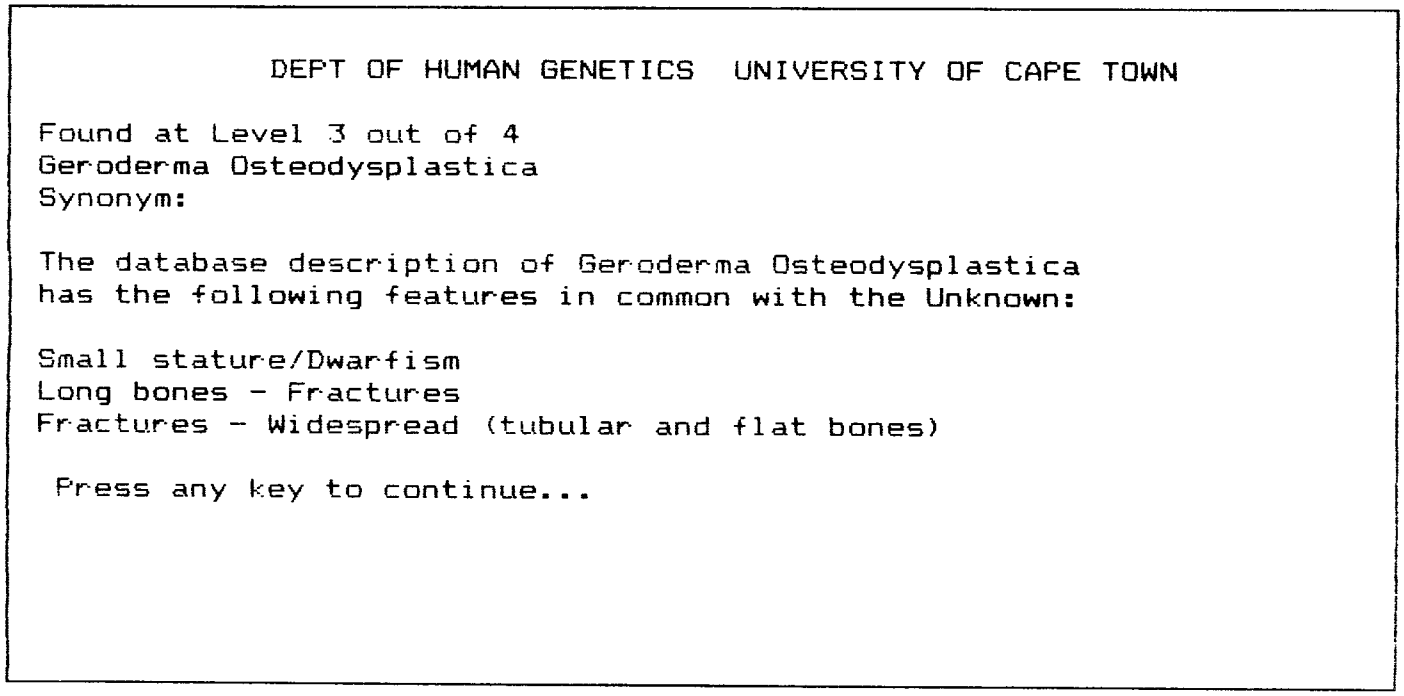


Fig 6-20 Third Find

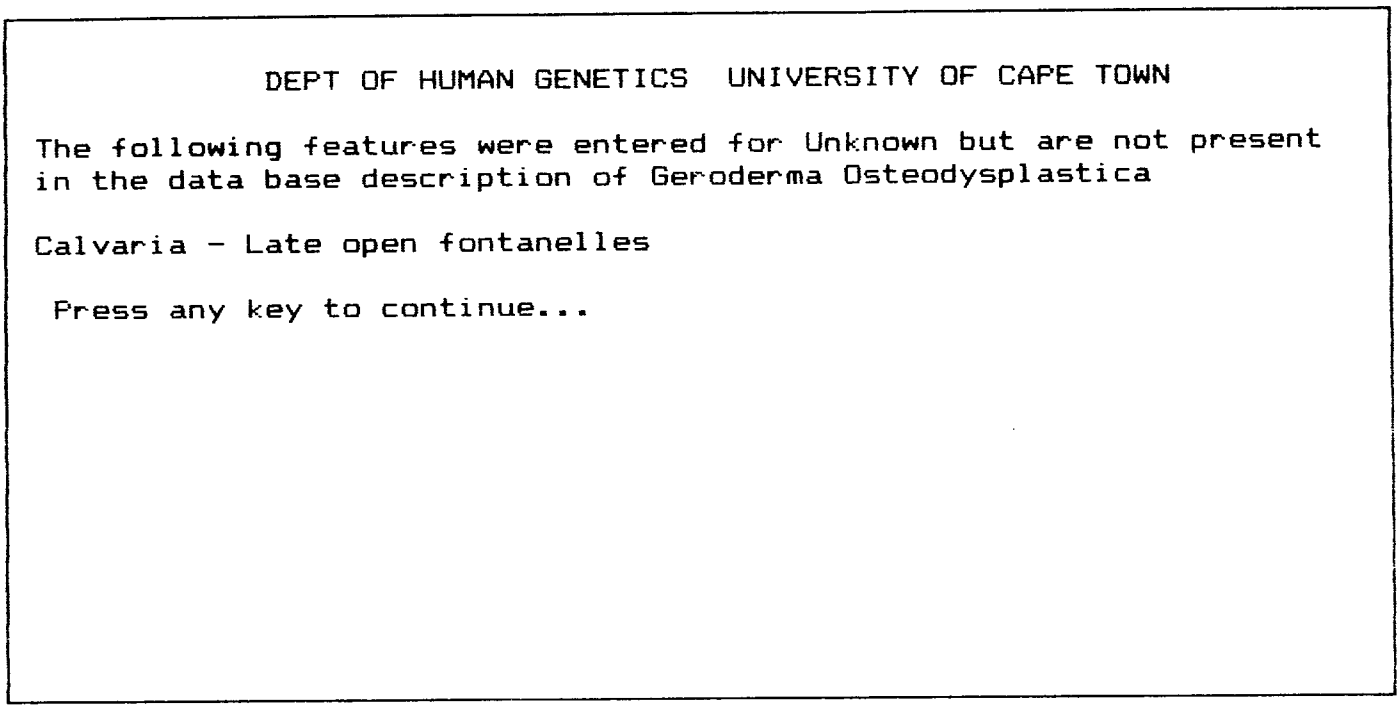


Fig 6-21 See Text

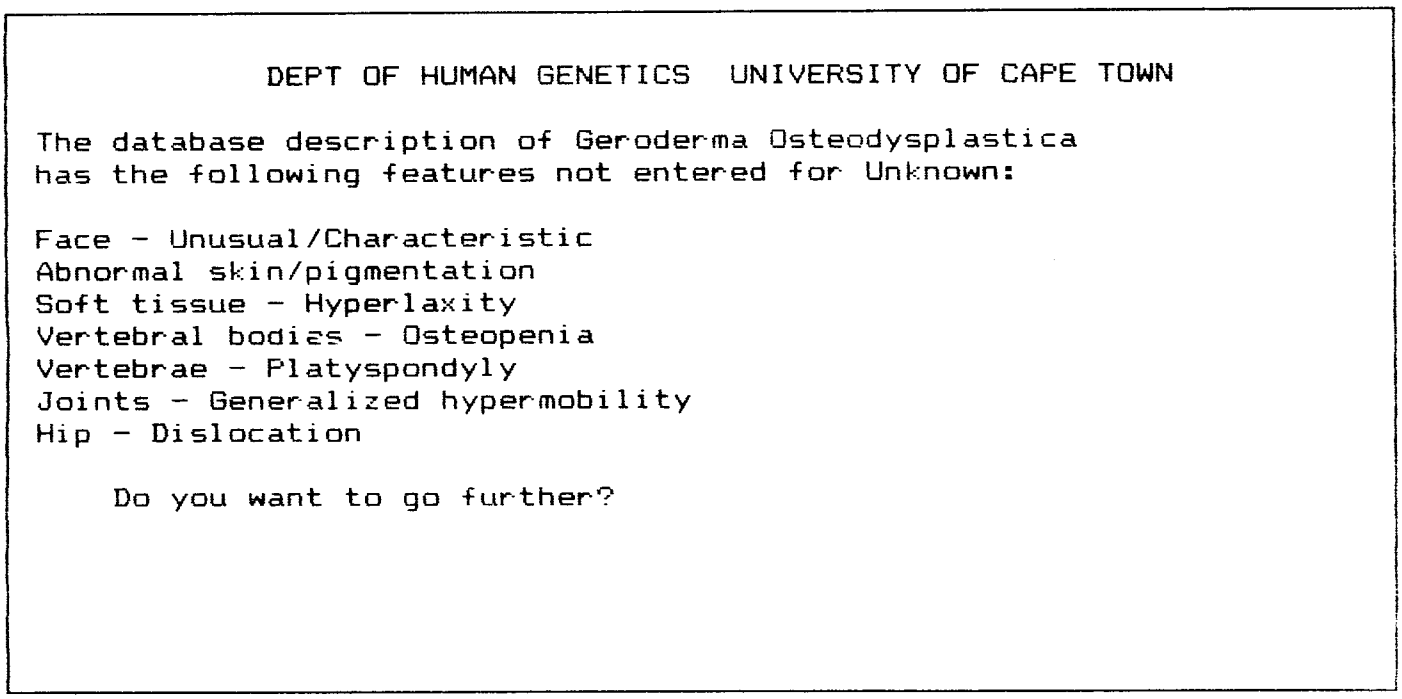


Fig 6-22 See Text

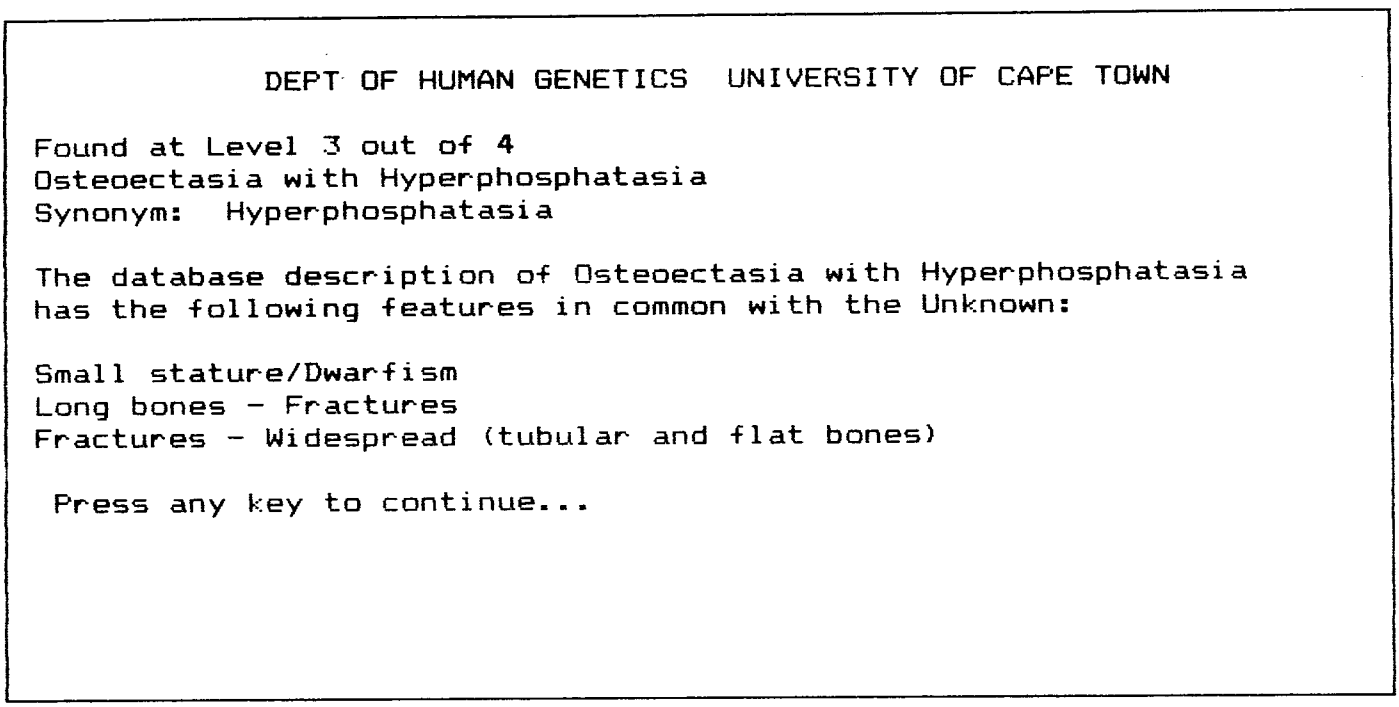


Fig 6-23 Fourth Find

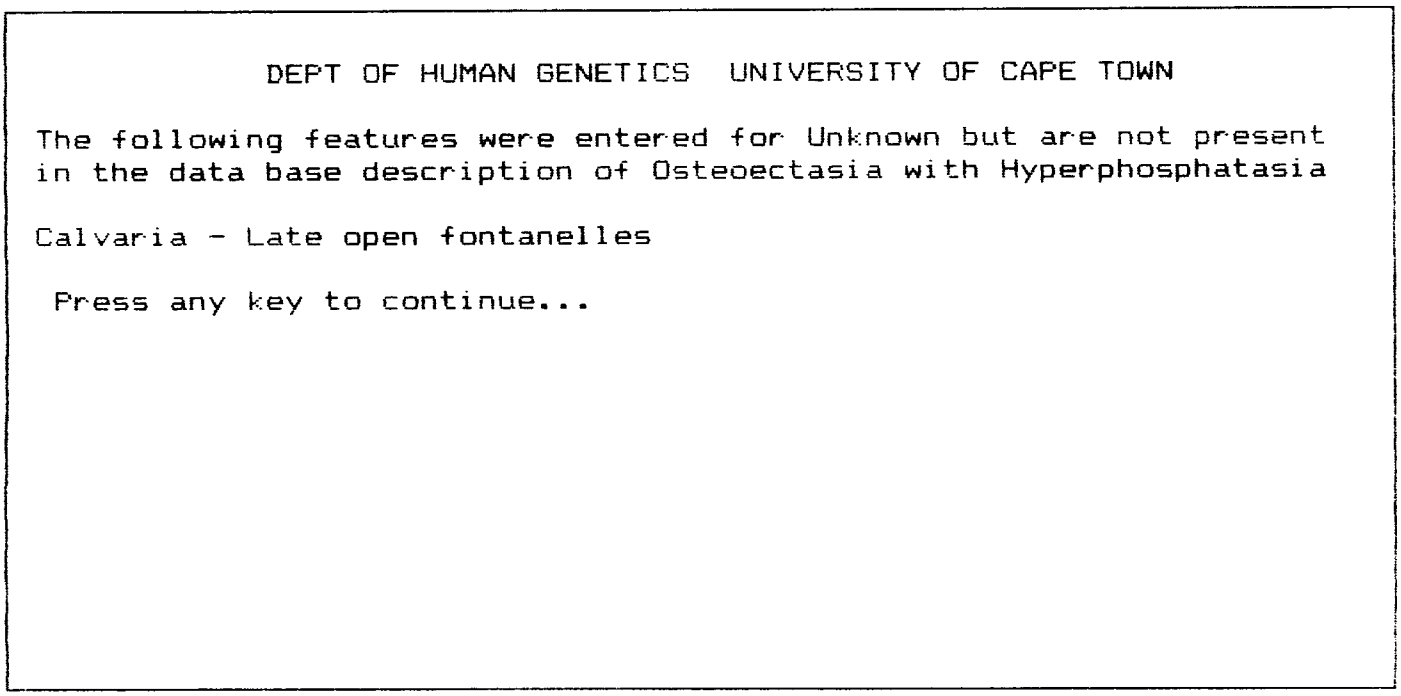


Fig 6-24 See Text

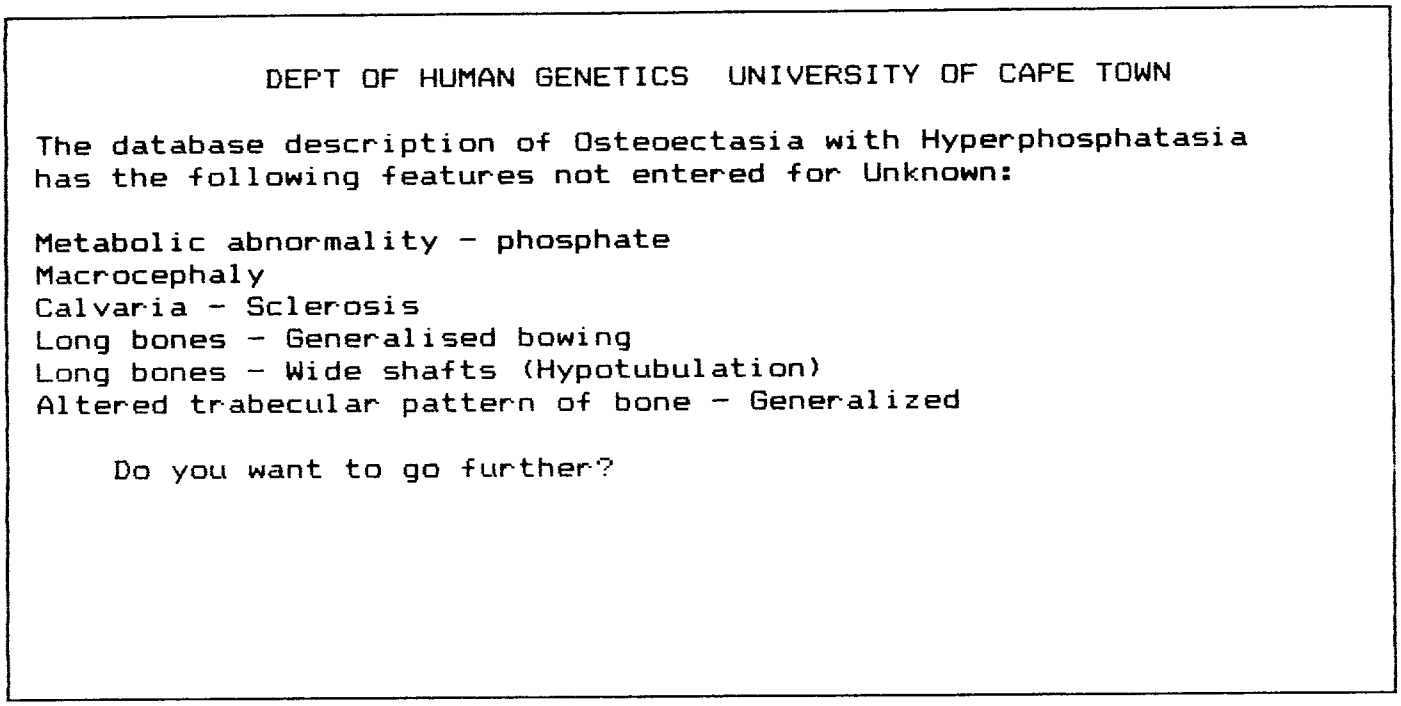


Fig 6-25 See Text

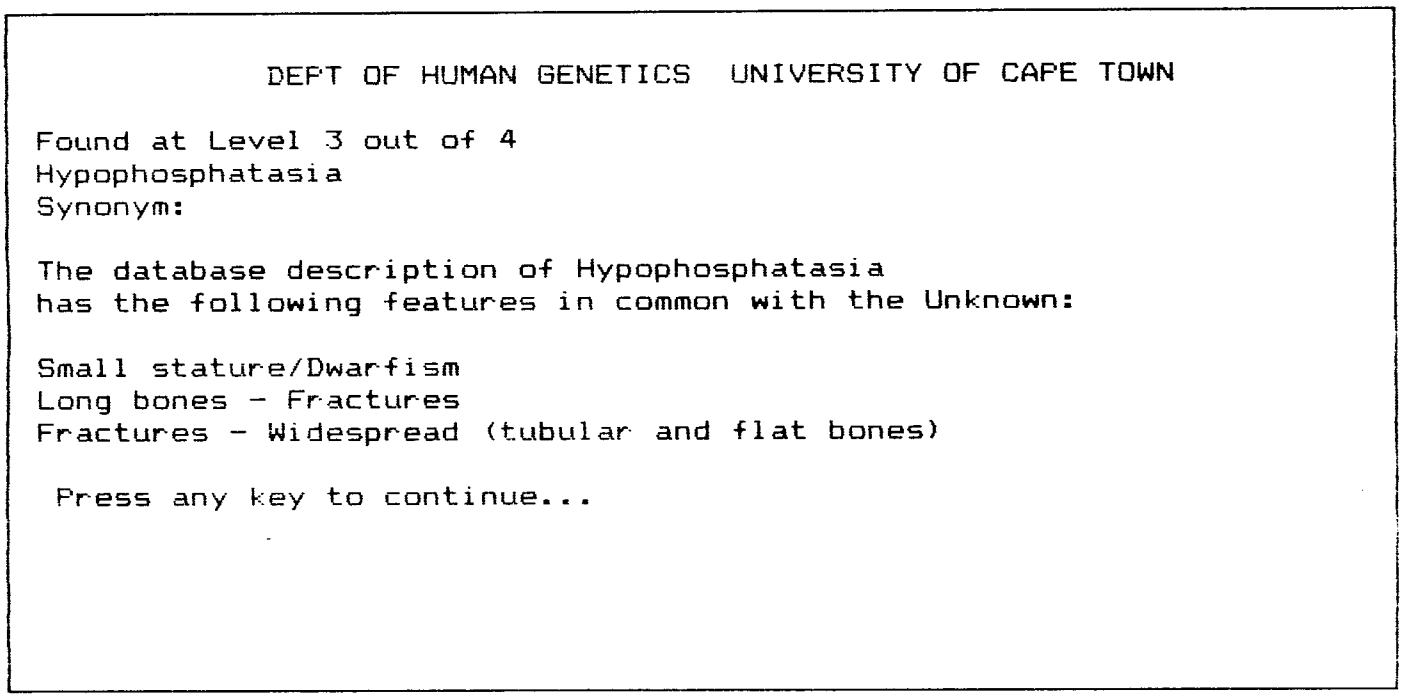


Fig 6-26 Fifth Find

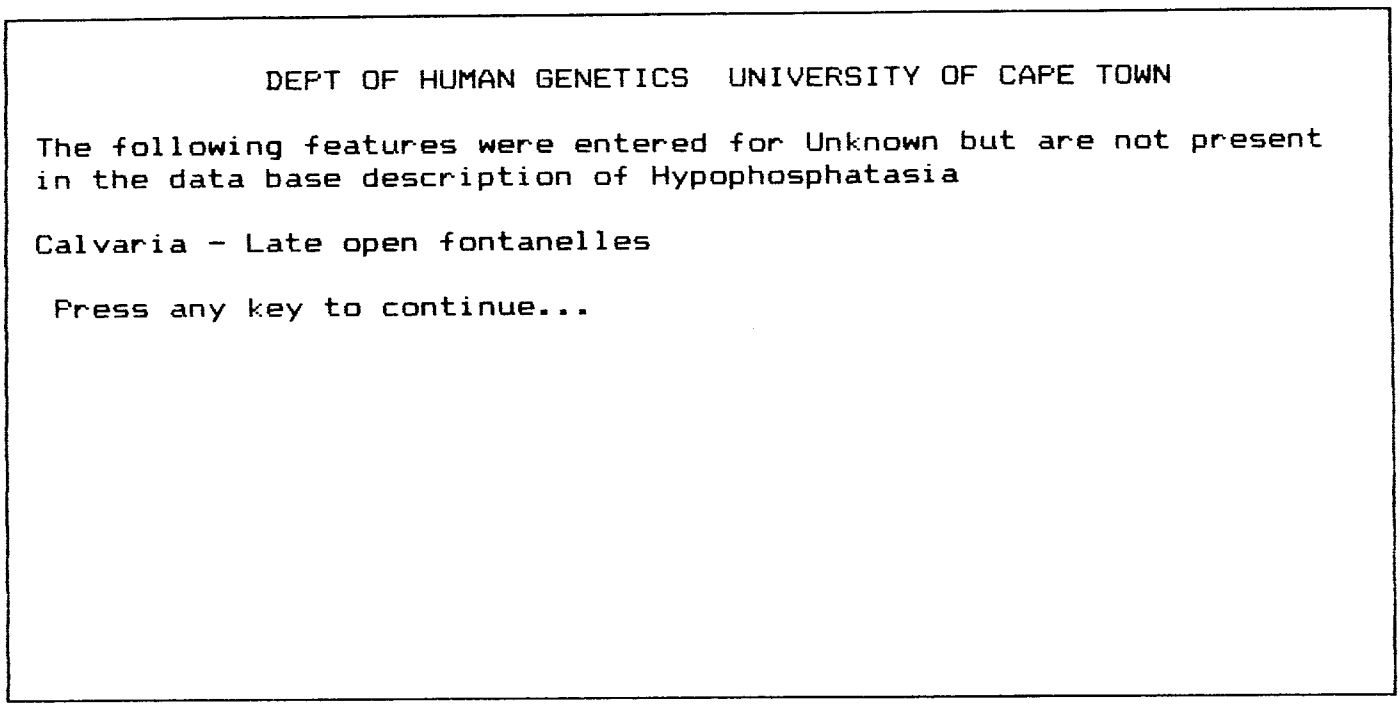


Fig 6-27 See Text

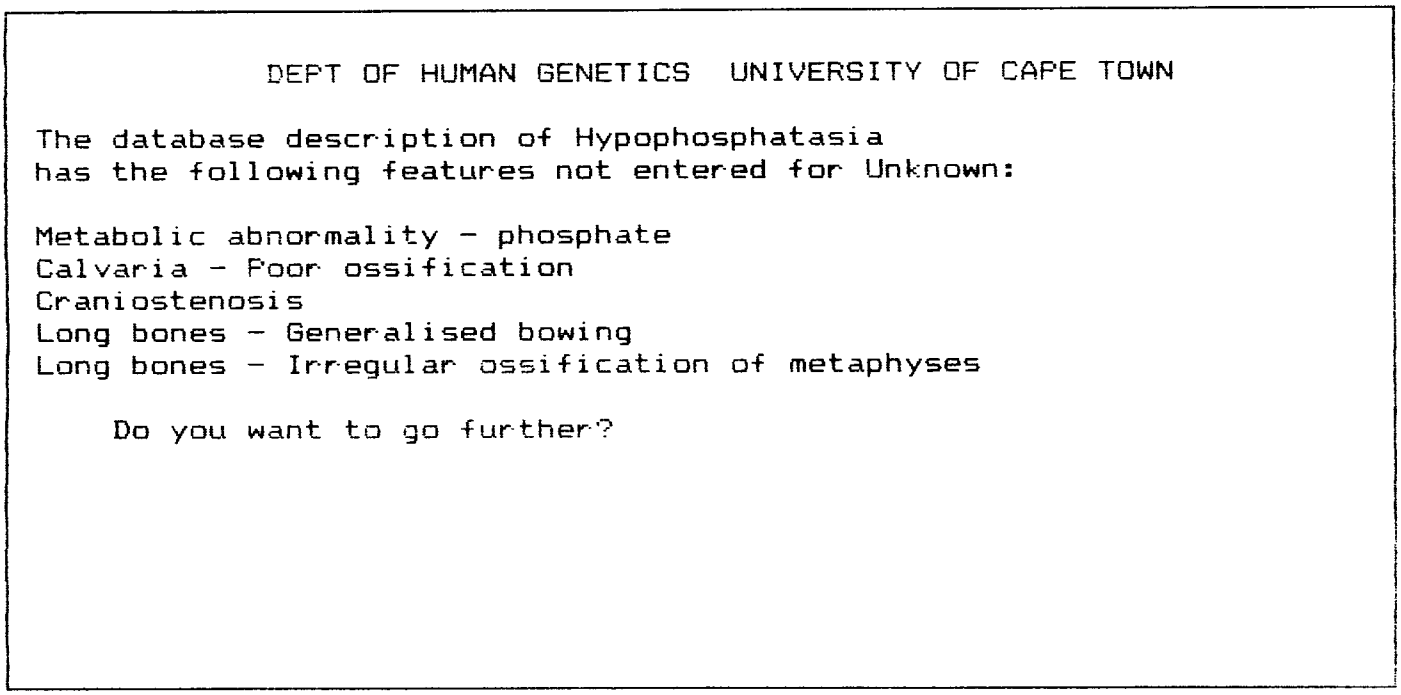


Fig 6-28 See Text

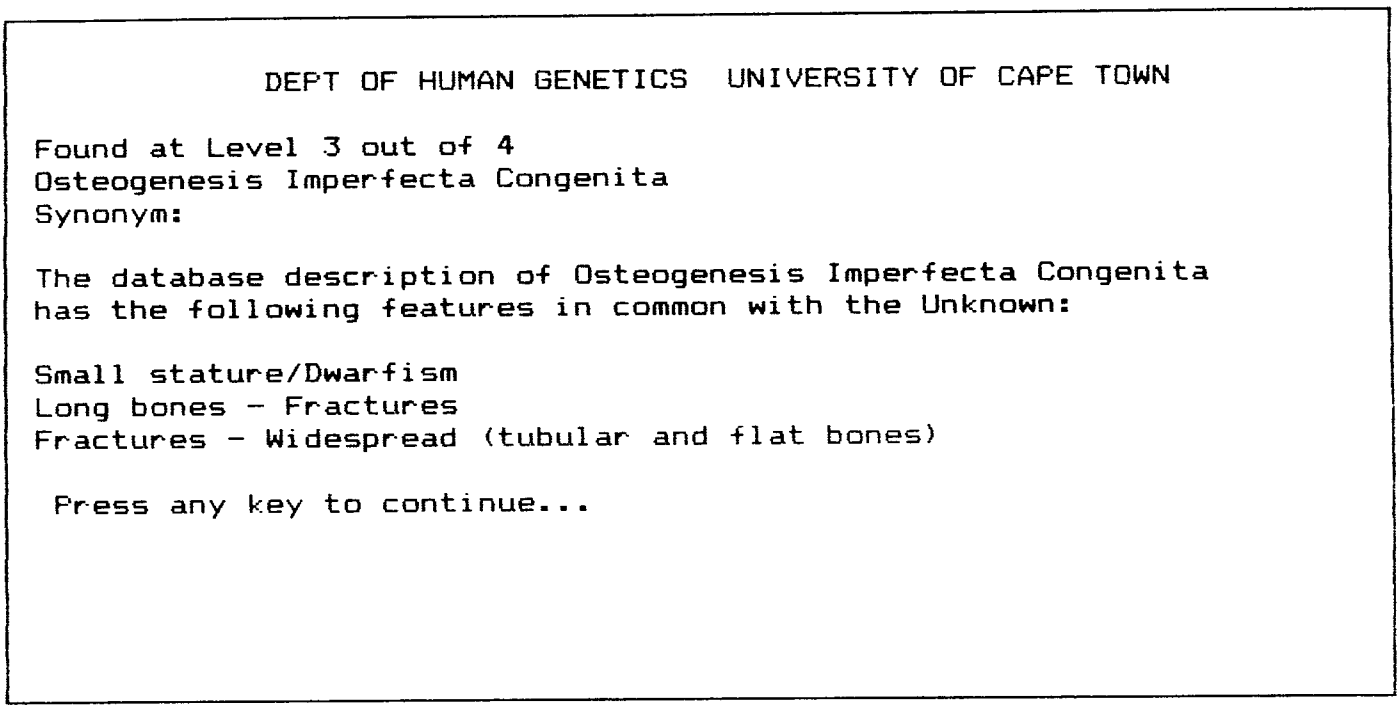


Fig 6-29 Sixth Find

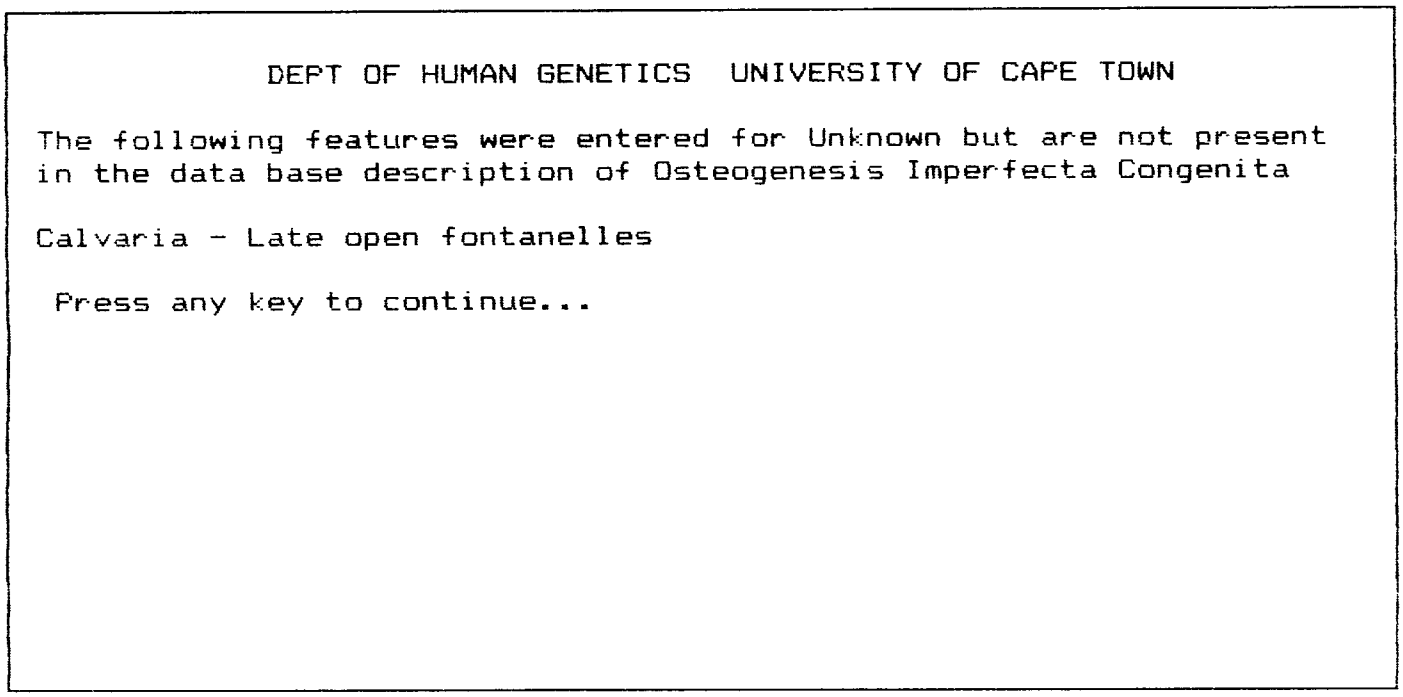


Fig 6-30 See Text

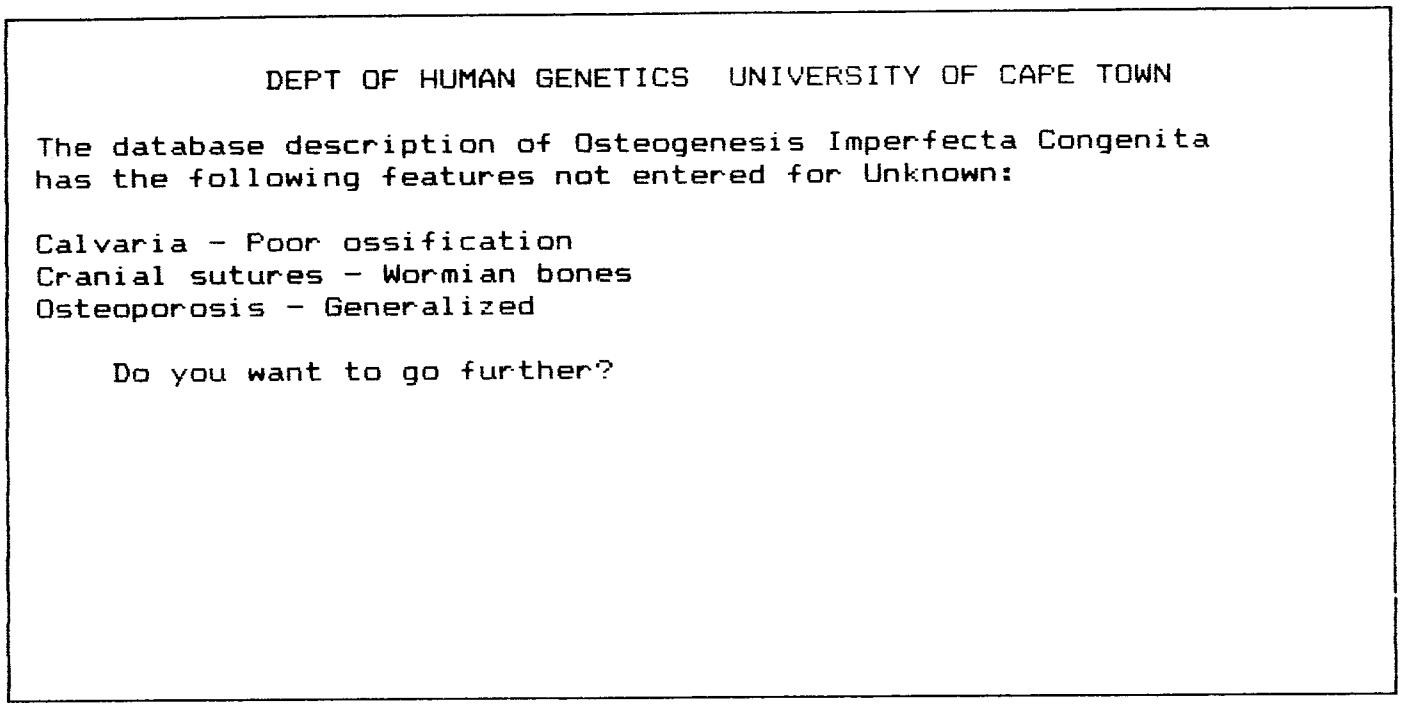


Fig 6-31 See Text

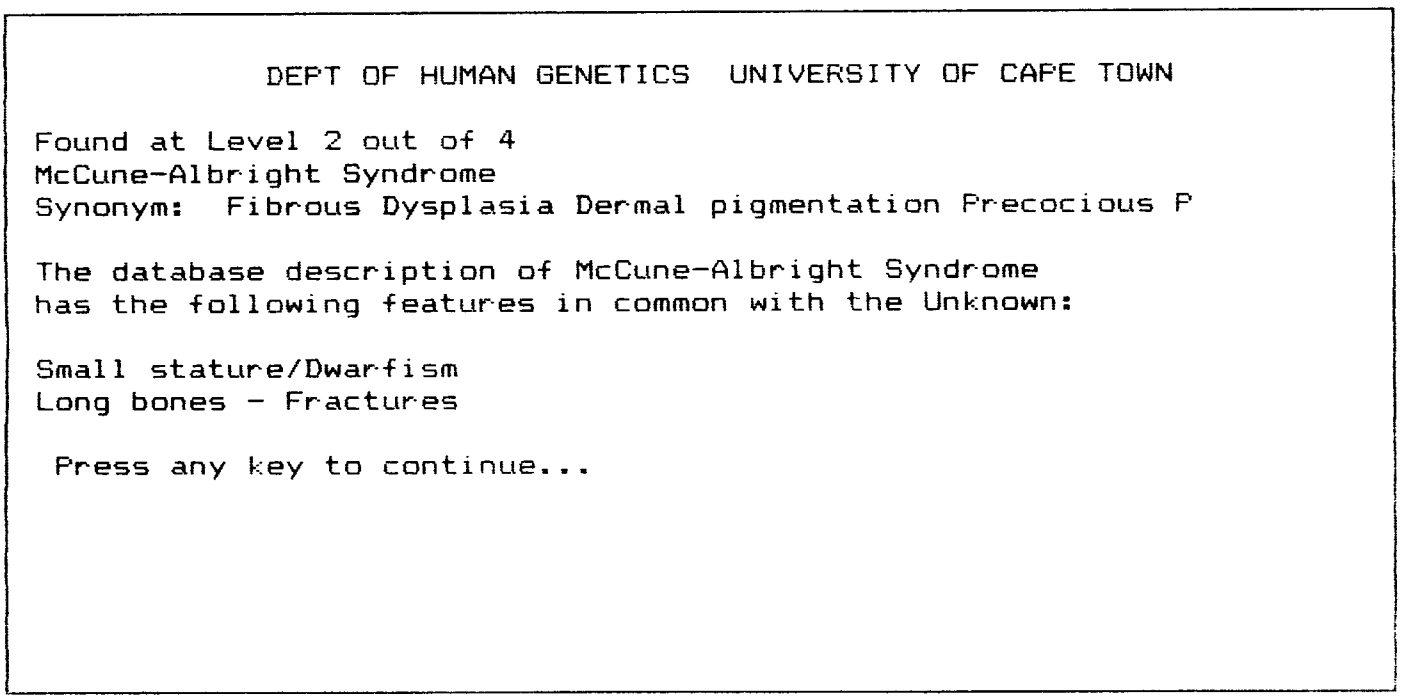


Fig 6-32 Seventh Find

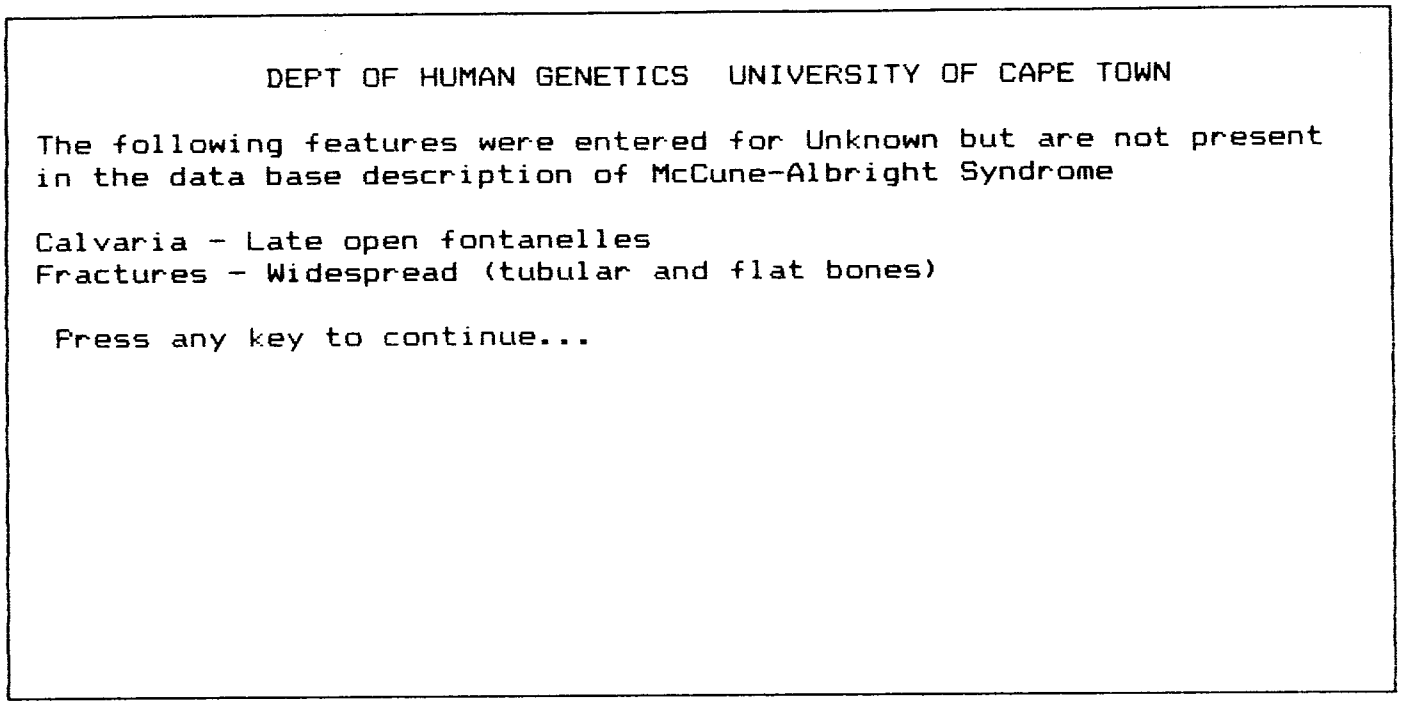


Fig 6-33 See Text

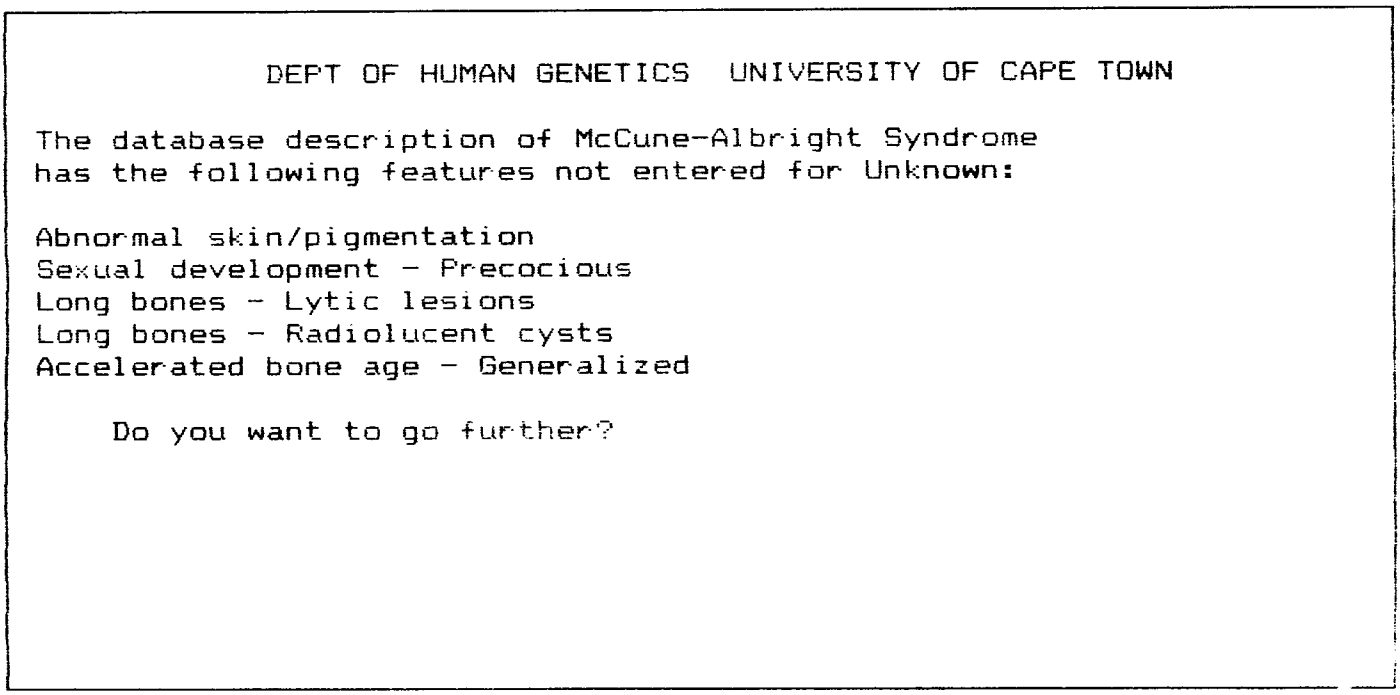


Fig 6-34 See Text

Pages 6-viii and 6-ix show how the following manifestations are selected:

- Dwarfism
- Late open fontanelles
- Fractures of long bones
- Widespread fracturing of bones.

The "D" option is selected from the menu on page 6-x. The first find is announced on the screen in a way shown at the bottom of page 6-x. Pycnodysostosis was found to agree with the entered set with four out of four of the manifestations. Two further screens are shown on page 6-xi, the first with a list of features present in the problem set but not in the current description of the recently matched disorder, and the second listing those stigmata present in the suggested condition but absent from the comparison list. Similar finds involving Dysosteosclerosis, Geroderma Osteodysplastica, Osteoectasia with Hyperphosphatasia, Hypophosphatasia, Osteogenesis Imperfecta Congenita and the McCune-Albright syndrome, are reflected on subsidiary pages 6-xii to 6-xx.

The following points should be noted:

- 1) The procedure may be terminated at any time by responding negatively to the prompt, "do you want to go further?".
- 2) Disorders are offered for consideration at successively lower levels of congruence of their manifestations with those in the entered list.
- 3) Conditions suggested at the same level are given in the order in which they appear in the data base and not necessarily in an order of importance or likelihood.

When the diagnostic aid program terminates, the menu depicted in Fig 6-1 is produced again. Option 5 from this list, will end the entire application and return the user to the computer's disk operating system.

SECTION SEVEN.

THE EVALUATION STAGE

CHAPTER 7.1
INTRODUCTION.

In this section the technique by which the application which resulted from this project was evaluated, together with the results, will be outlined. Two aspects were assessed:

- i) Program efficiency from a data processing point of view;
- ii) The value of the package as a diagnostic aid as seen from a medical perspective.

CHAPTER 7.2
PROCESSING EFFICIENCY.

It is widely appreciated that Turbo Pascal programs execute extremely rapidly. When some of the slower procedures are written in assembly language, as was done in this study, an additional significant gain in efficiency is obtained.

To assess the processing efficiency, tests were conducted with the programs and random access files stored on the fixed disk of the Sperry AT computer. The same tests were then repeated after these items had been transferred to a "virtual" disk in the random access memory of the same machine. The latter approach resulted in significantly reduced access times since the mechanical factors which are responsible for delays were eliminated.

The following tests, the results of which are summarised in table 7-1 on page 7-4, were performed:

7.2.1 Program Loading.

To ascertain loading efficiency, the option involving the largest program (Prospect.chn) was selected from the opening menu. The time was measured from the moment that the option number was entered on the keyboard, to the appearance of the first table of manifestations on the screen.

7.2.2 Disease Profile Retrieval.

Tuberous Sclerosis was selected for this test since the profile of this disorder is situated in record no. 188 near the end of the file Profiles.ran. As such it is representative of a "worst case" situation. Two times were measured; the first after "Tuberous Sclerosis" had been entered as a search string, and the second when the synonym "Epiloia" was employed.

7.2.3 Time to conduct a Differential Diagnosis.

Two tests were performed in this respect. In the first place, four of the manifestations of the Weaver syndrome were entered and the time taken for this disorder to appear on the screen as a suggested diagnosis, measured. Weaver syndrome occupies record no. 191 near the very end of the file.

In the second instance a fictitious and bad combination of six manifestations was

selected. The program scans the entire database for a condition manifesing all six the stigmata. Having failed to find such a condition, a search is made for a disorder with five of the features, and so on. The unlikely combination ensures that every scan which involves more than one common manifestation will result in failure. The first find on the sixth scan will be that of the first disorder with one manifestation in common with the combination. When this condition was presented on the screen, the elapsed time was noted.

time of 15,9 seconds taken by the fictitious combination suggests that the entire disease file is scanned in about 15,9 / 5 = 3,2 seconds, since just over 5 scans were employed in this test. When a virtual disk is used, the scanning time is reduced to 7,5 / 5 = 1,5 seconds. This finding is in accordance with the time of 2 seconds that was needed to reach Weaver Syndrome.

A database consisting of 2 000 similar profiles instead of 200, and residing on a 20Mb fixed disk of an IBM AT type com

Test	**Time (FD)** *(Seconds)*	**Time(VD)** *(Seconds)*
Program Load (Largest Module)	3,9	1,4
Disorder Retrieval (Worst Case)		
Tuberous Sclerosis	3,4	1,3
Epiloia	3,5	1,4
Differential Search		
Weaver Syndrome (see text)	3,4	2,0
Bad combination (see text)	15,9	7,5

FD = Fixed Disk VD = Virtual Disk

Table 7-1 Processing and Retrieval Efficiency

It is evident from table 7-1 that the programs execute extremely rapidly as judged by microcomputer standards. The

puter, should therefore require about 32 seconds to be scanned. A virtual disk of 750 Kb (which is feasible) on a similar

machine will also be able to host such a structure and should reduce the scanning time to about 15 seconds.

CHAPTER 7.3
THE VALUE OF THE APPLICATION AS A DIAGNOSTIC AID.

When evaluating an application such as this as a diagnostic aid, two factors should be borne in mind:

i) That an initial "inexperience" exists;

ii) That the system is able to "learn" and thus gain experience.

7.3.1 Initial Inexperience.

Although a significant amount of interaction took place between the knowledge engineer and the domain experts, the majority of the initial information was obtained and selected by the former. A kernel, consisting of disease profiles which the knowledge engineer believed to be correct, was estabished from sources such as the Gamut Index and the literature. It is possible, however, that some of the emphasis may have been misplaced and the relative importance of some of the manifestations in a particular profile assessed incorrectly. This situation may be likened to a newly qualified intern who often possesses an extensive knowledge base, but who still lacks the experience to assess the relative importance of the various facets making up the knowledge base.

7.3.2 The Ability to Learn.

The editing facilities incorporated in the application, provide for improvement of the knowledge base with the passage of time. It is likely that the "expertese" of the system will improve continually with repeated use by experts. This has already been experienced in recent sessions with the domain experts (Beighton, 1987; Spranger, 1987). In section 8 it will be shown that such improvement of the data base can be effected without significant demands on computer storage.

When the diagnostic facility of the package offers a suggestion that seems inappropriate for a particular set of manifestations, it should not summarily be regarded as failure, but rather as a sincere, albeit incorrect, contribution from a junior expert in training. If the senior domain expert were to review the profiles in question and, with the facilities available, make the necessary adjustments, a somewhat more "enlightened" junior results. Moreover, the lesson learned by the computer will never be misplaced or "forgotten".

7.3.3 Assessing the Diagnostic Aid Utility.

To assess the value of the application as a diagnostic aid, two approaches were adopted. First, various combinations of manifestations were presented to the computer and the results perused. In the second instance data from some real cases were entered and the differential diagnosis option selected from the menu.

7.3.3.1 Computer Response to Combinations of Manifestations

Subsidiary pages 7-i to 7-vi show the outcome of six tests in this category. These printouts were referred to domain expert Beighton for critical comment. A discussion follows on page 7-5, immediately after the subsidiary pages.

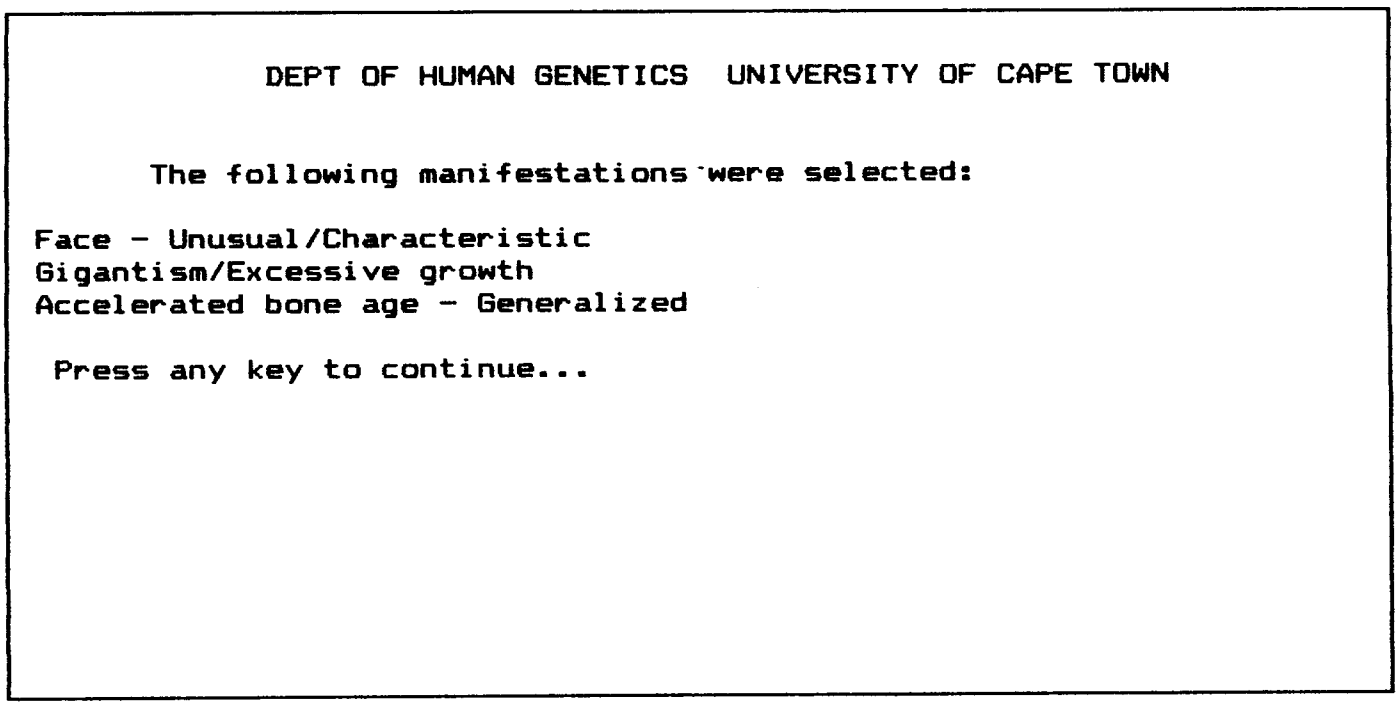

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN
The following manifestations were selected:
Face - Unusual/Characteristic
Gigantism/Excessive growth
Accelerated bone age - Generalized
Press any key to continue...

Fig 7-1

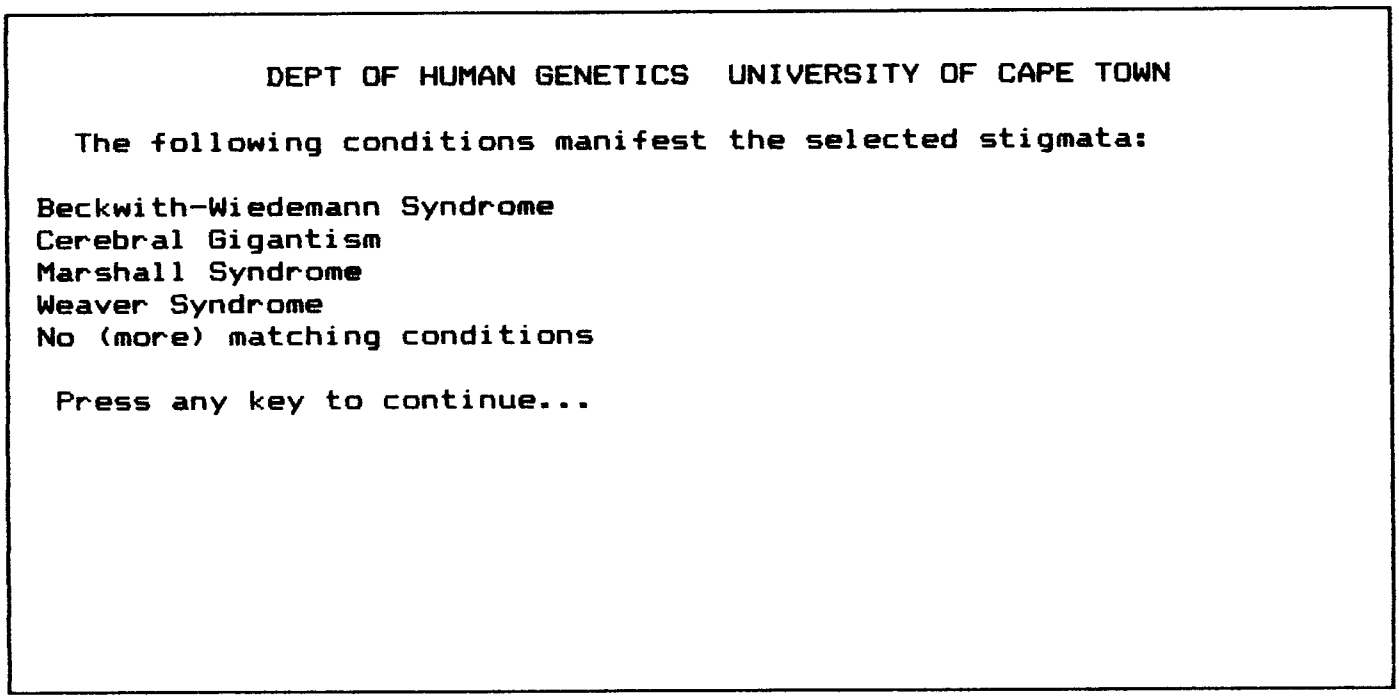

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN
The following conditions manifest the selected stigmata:
Beckwith-Wiedemann Syndrome
Cerebral Gigantism
Marshall Syndrome
Weaver Syndrome
No (more) matching conditions
Press any key to continue...

Fig 7-2

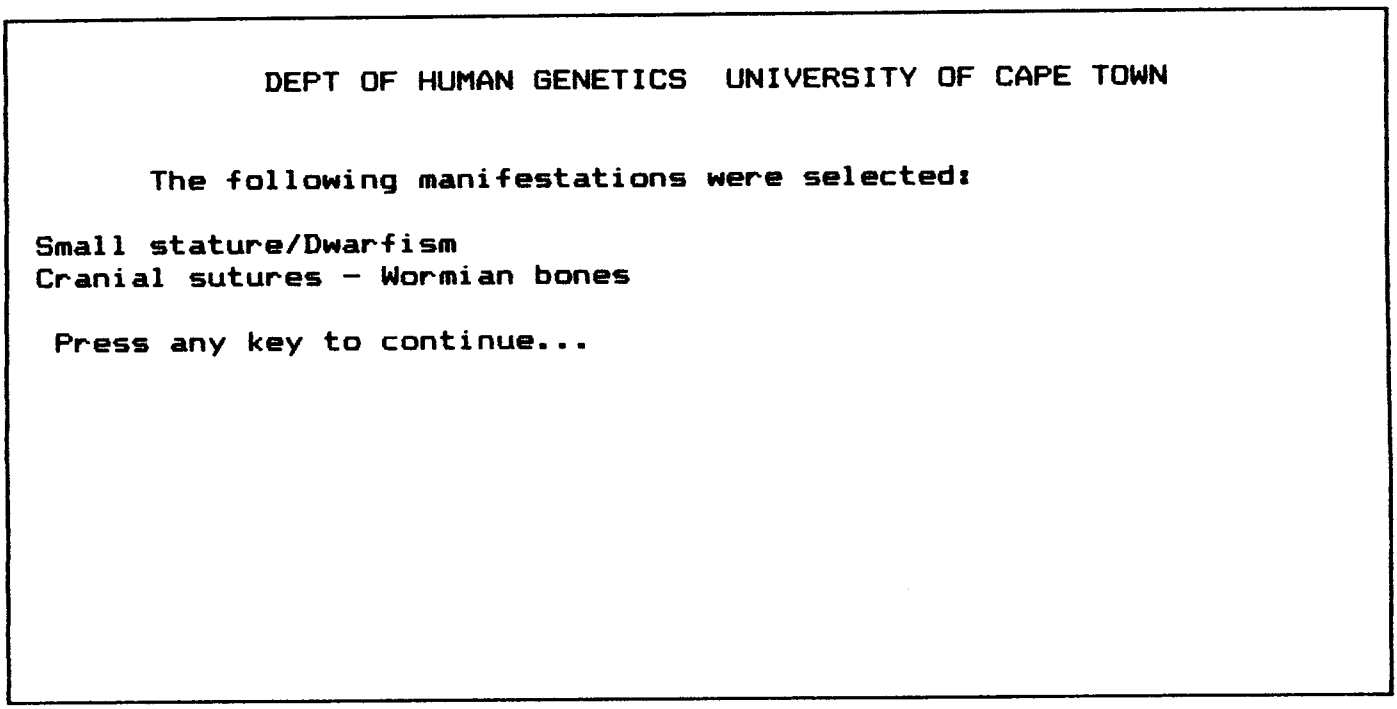

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN
The following manifestations were selected:
Small stature/Dwarfism
Cranial sutures - Wormian bones
Press any key to continue...

Fig 7-3

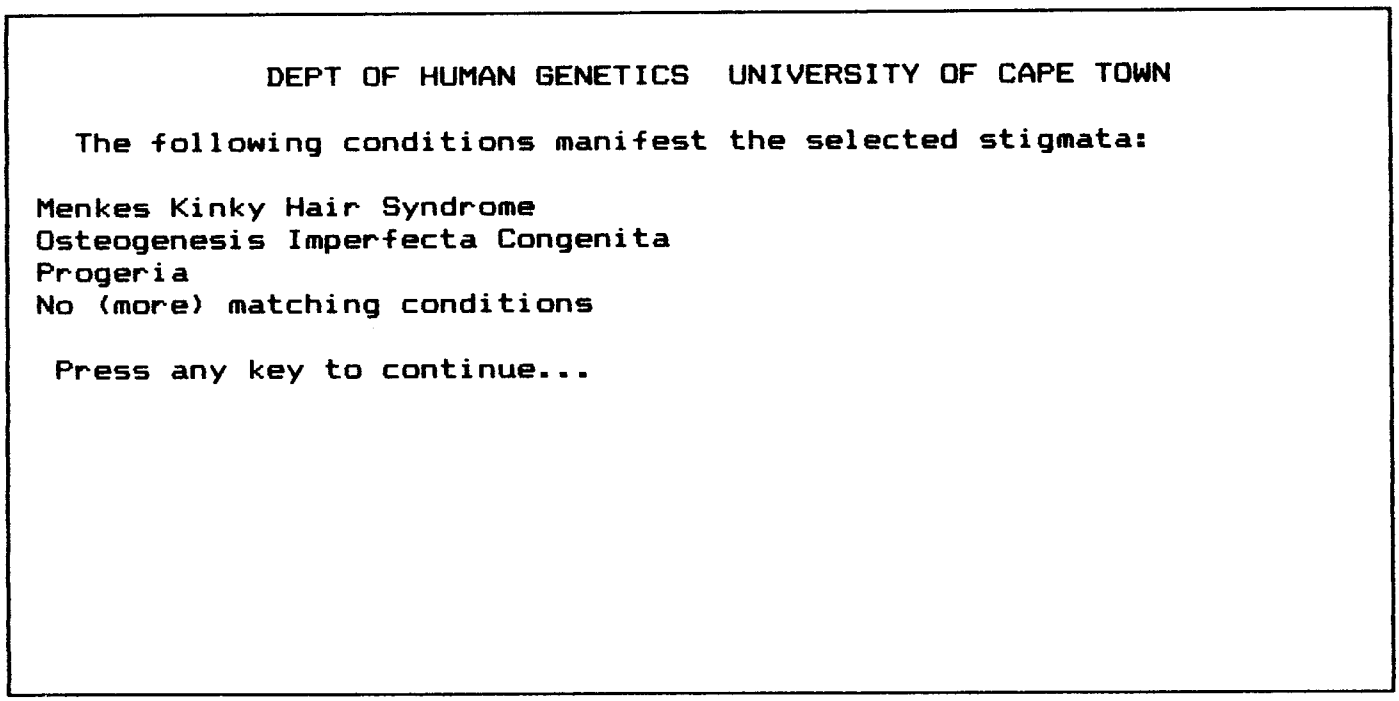

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN
The following conditions manifest the selected stigmata:
Menkes Kinky Hair Syndrome
Osteogenesis Imperfecta Congenita
Progeria
No (more) matching conditions
Press any key to continue...

Fig 7-4

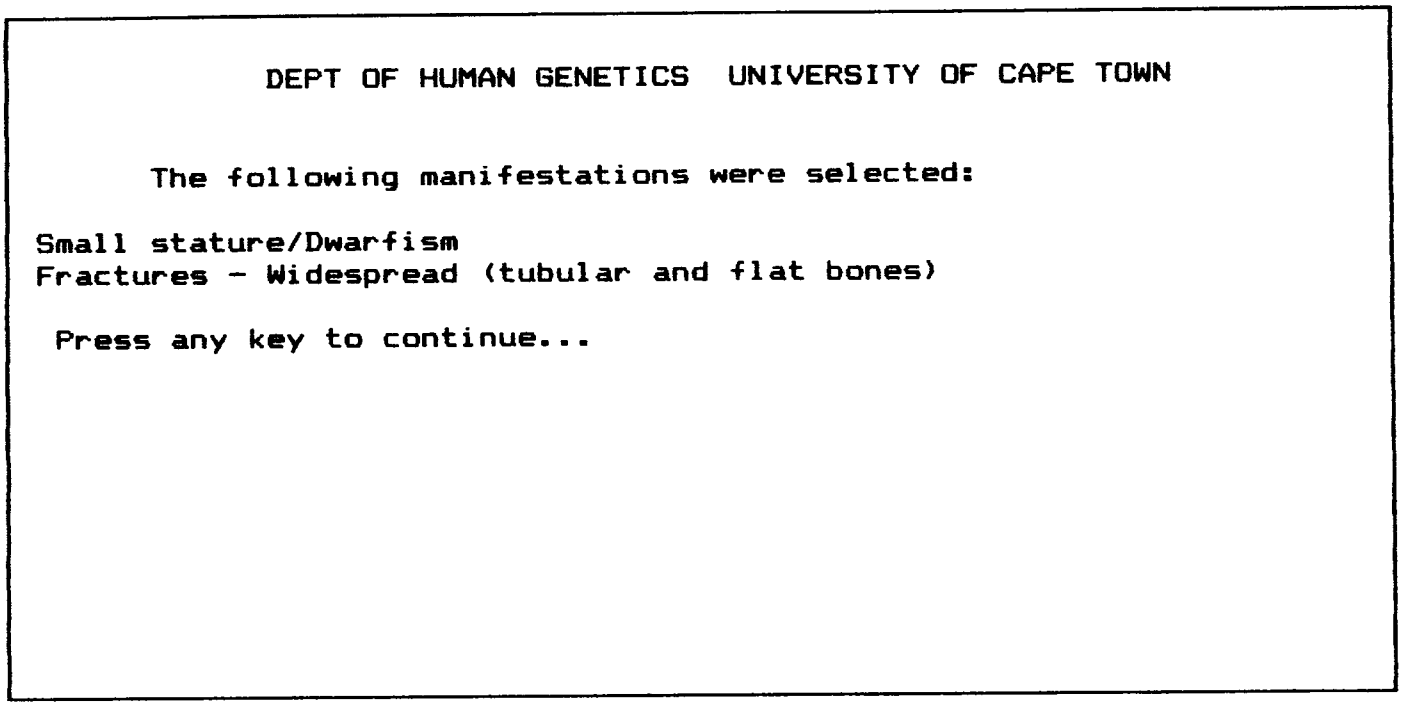

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN
The following manifestations were selected:
Small stature/Dwarfism
Fractures - Widespread (tubular and flat bones)
Press any key to continue...

Fig 7-5

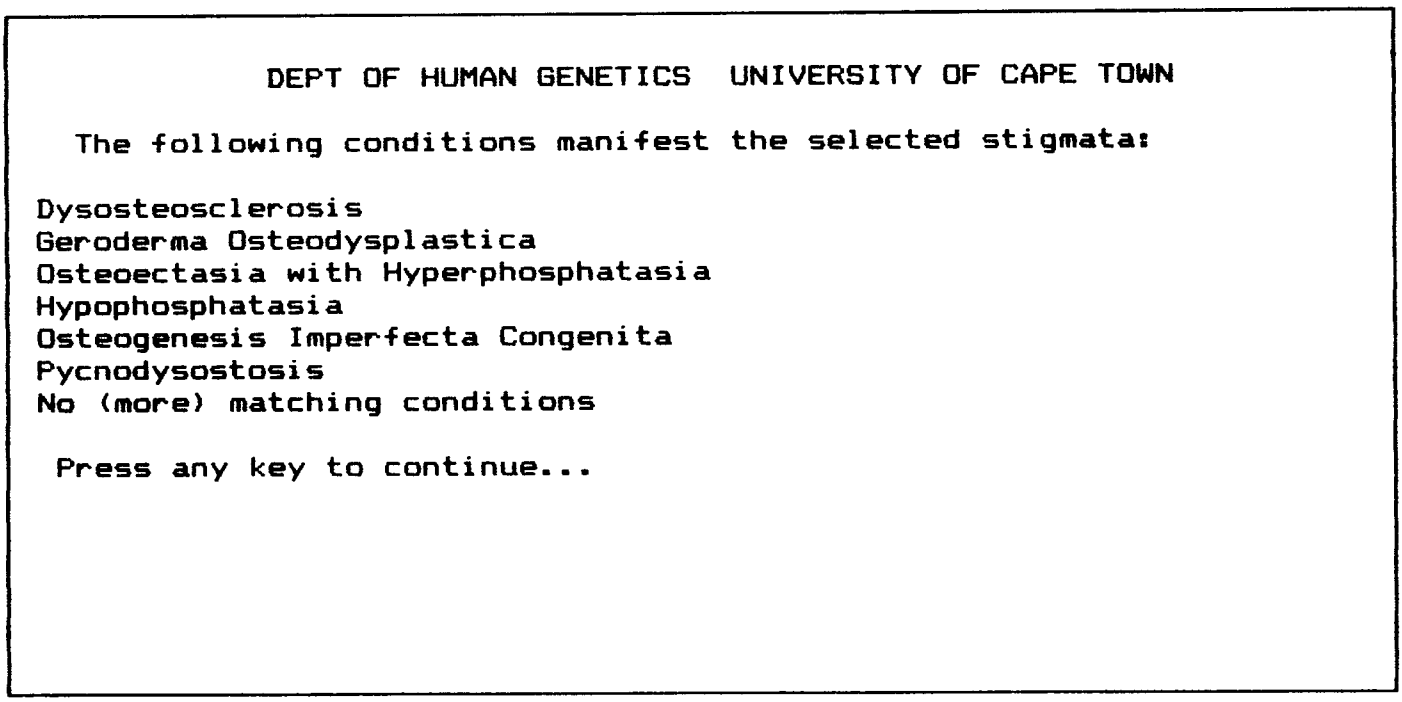

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN
The following conditions manifest the selected stigmata:
Dysosteosclerosis
Geroderma Osteodysplastica
Osteoectasia with Hyperphosphatasia
Hypophosphatasia
Osteogenesis Imperfecta Congenita
Pycnodysostosis
No (more) matching conditions
Press any key to continue...

Fig 7-6

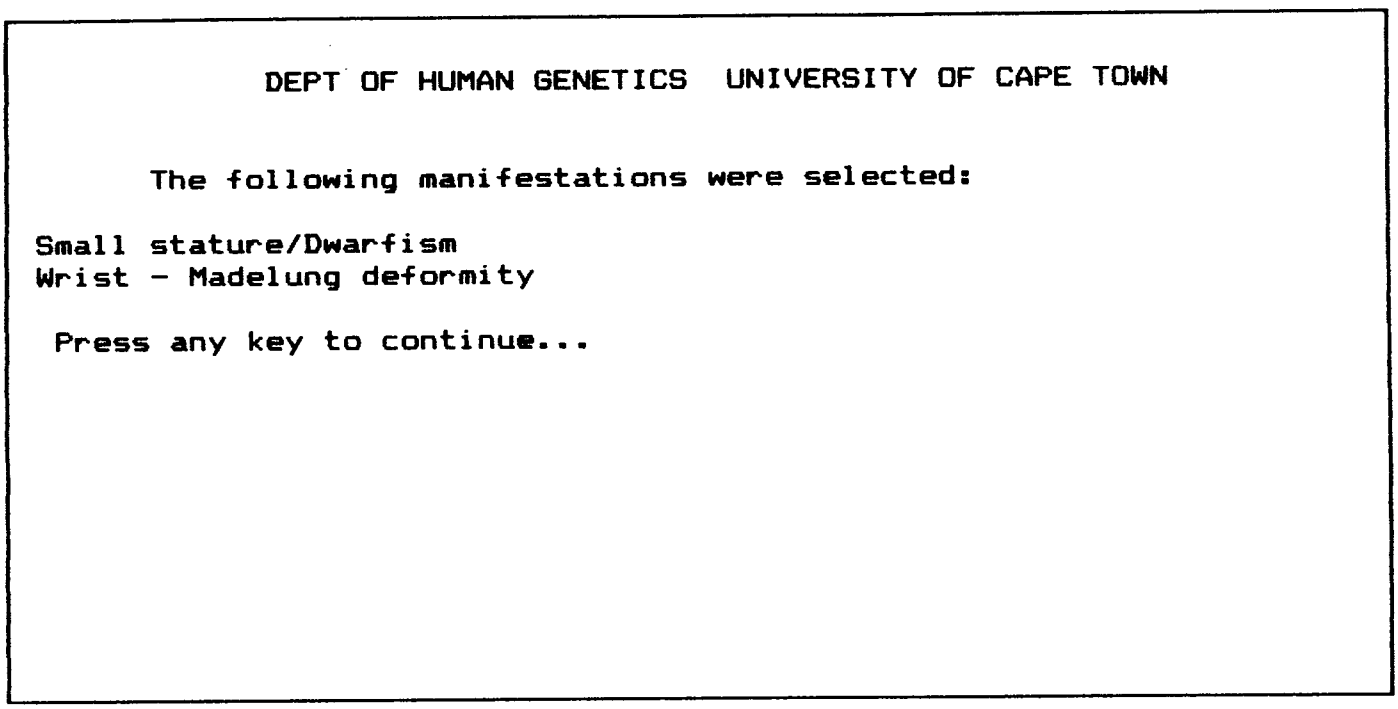

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN
The following manifestations were selected:
Small stature/Dwarfism
Wrist - Madelung deformity
Press any key to continue...

Fig 7-7

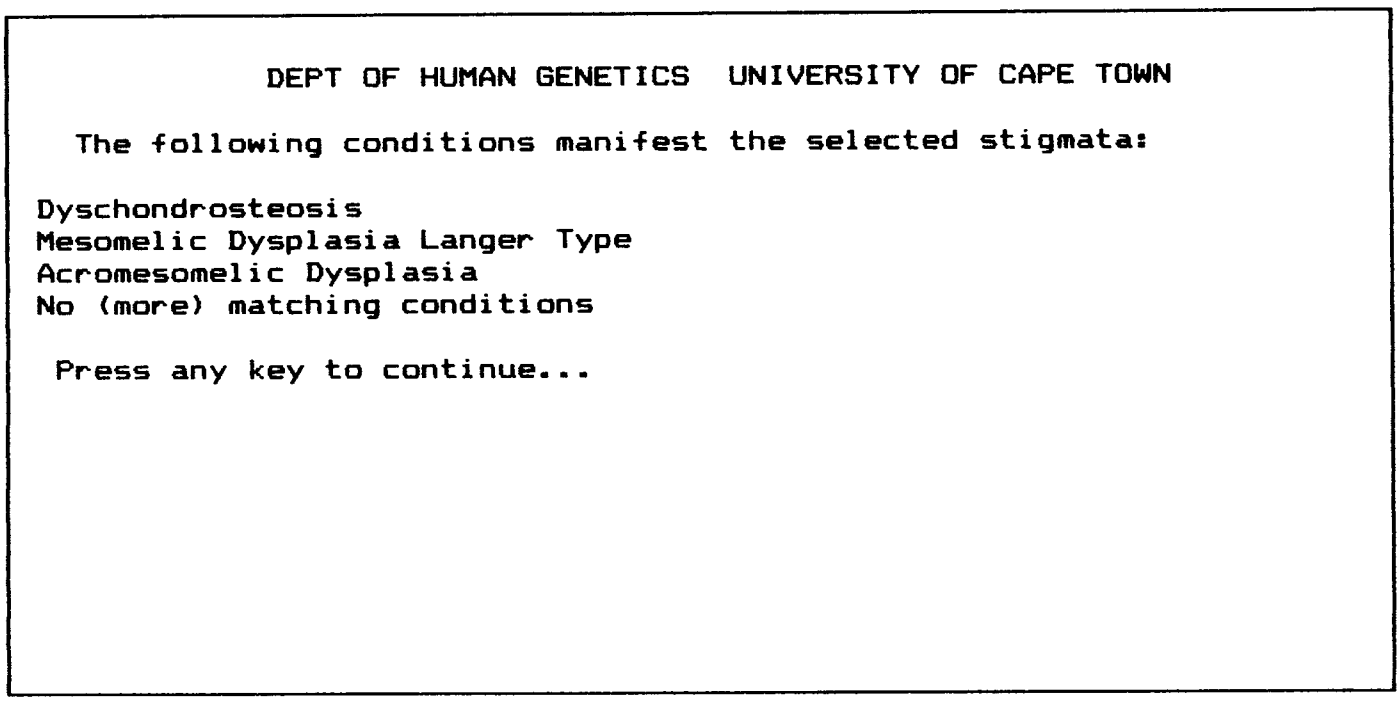

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN
The following conditions manifest the selected stigmata:
Dyschondrosteosis
Mesomelic Dysplasia Langer Type
Acromesomelic Dysplasia
No (more) matching conditions
Press any key to continue...

Fig 7-8

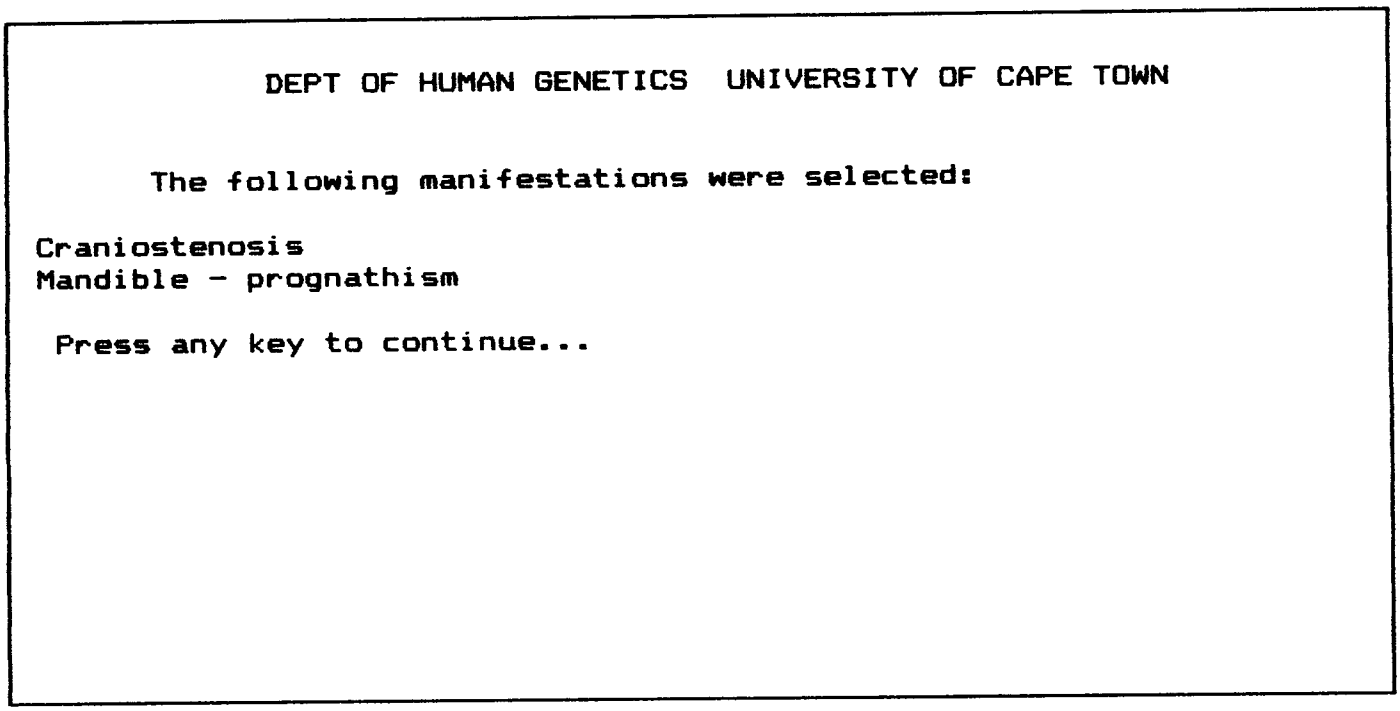


Fig 7-9

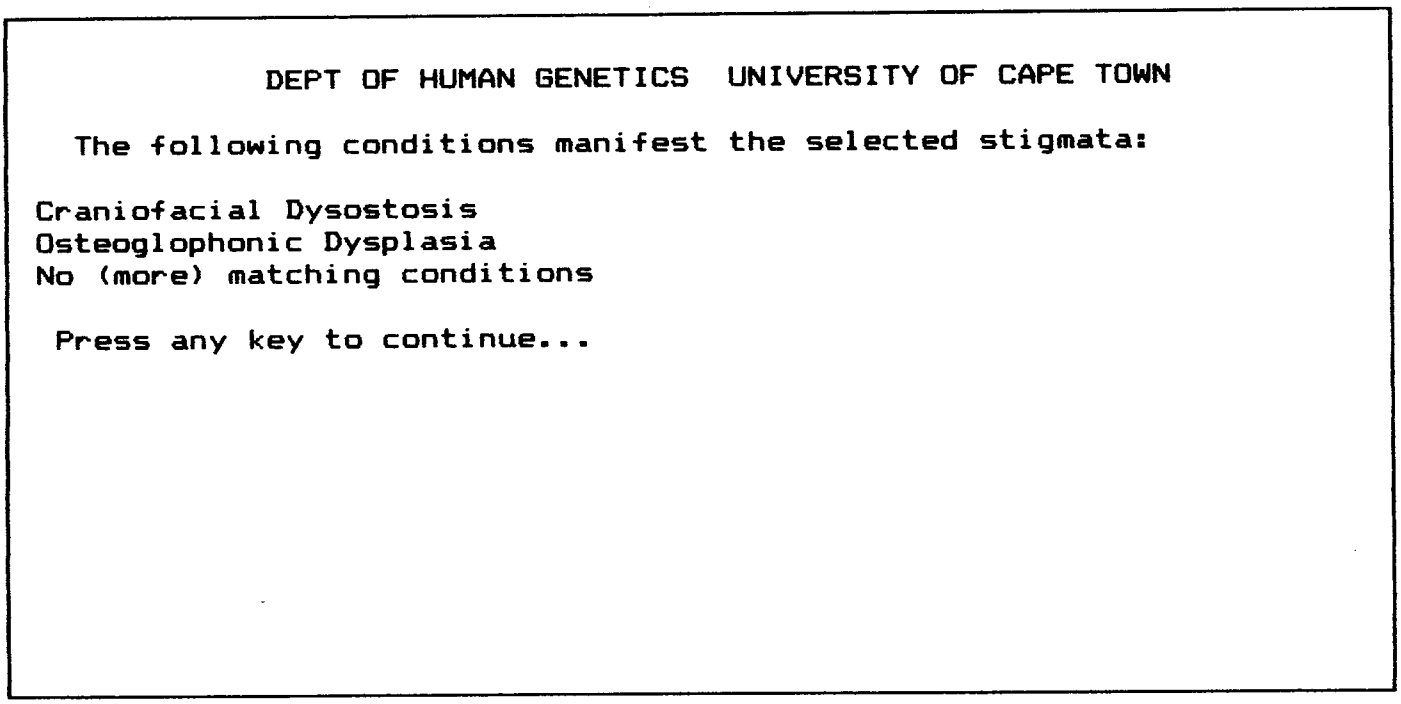


Fig 7-10

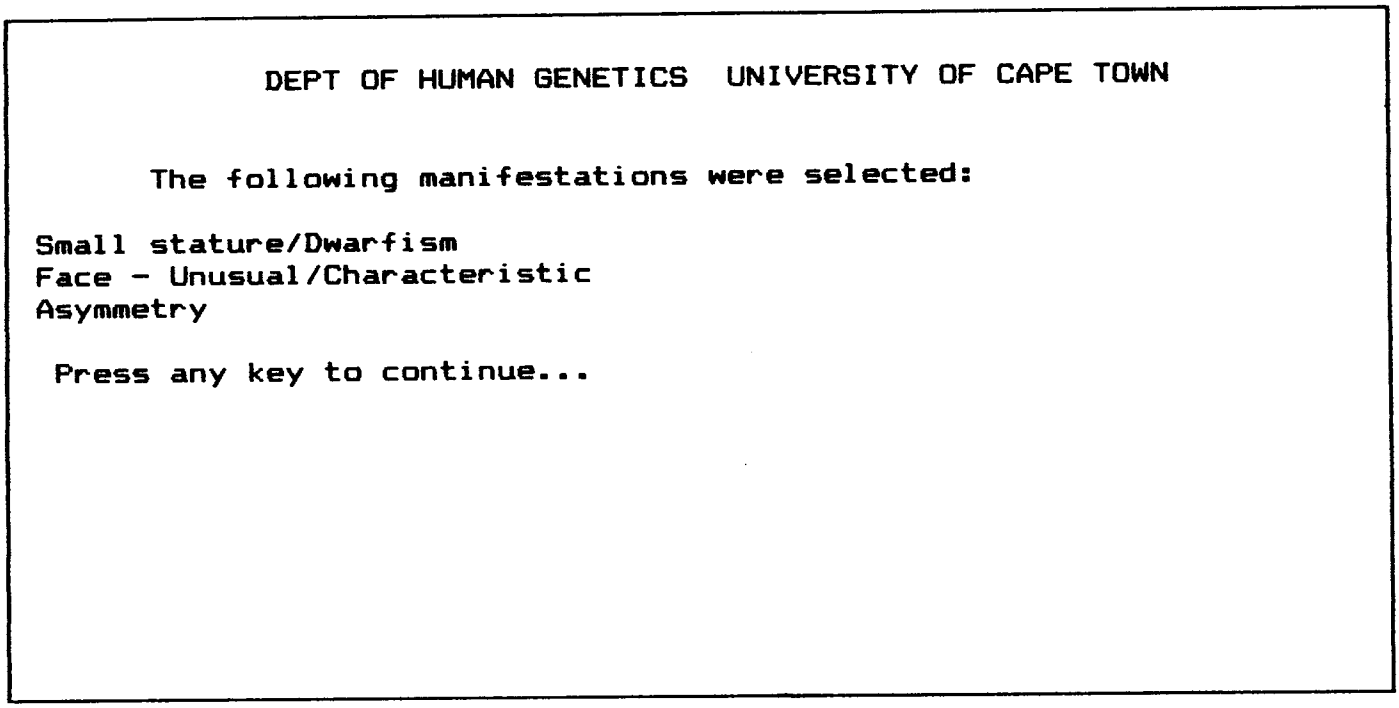


Fig 7-11

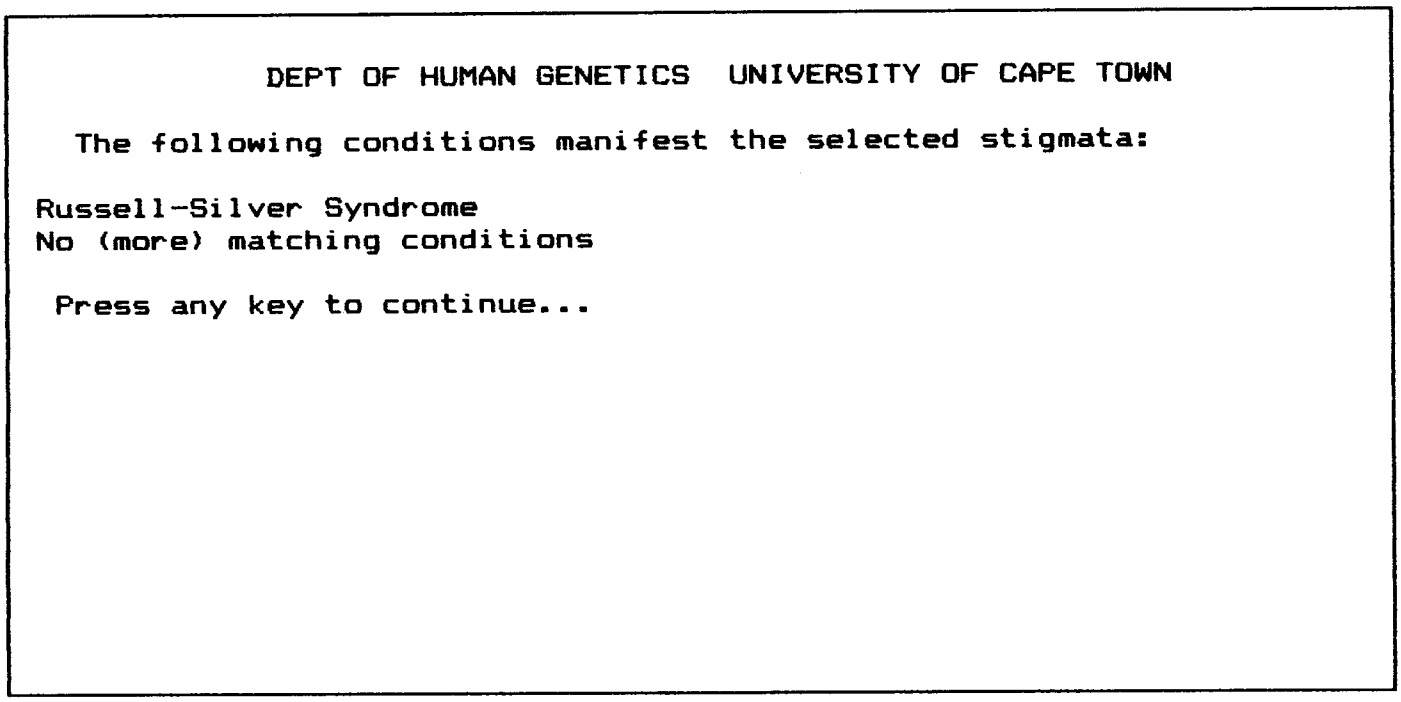


Fig 7-12

Page 7-i illustrates the response to the following combination:

- Unusual face
- Excessive growth
- Accelerated bone age

The computer's response consisting of Beckwith-Wiedemann syndrome, Cerebral gigantism, Marshall syndrome and Weaver syndrome, was considered to be correct and complete.

The initial response to the Dwarfism and Wormian bones combination on page 7-ii consisted only of Menkes Kinky Hair syndrome and Progeria. The absence of Osteogenesis Imperfecta Congenita was considered by the domain expert to be an important omission. The profile of OI congenita was subsequently amplified by the addition of the particular manifestation, illustrating once again the system's ability to "learn".

The examples on pages 7-iii to 7-vi were considered correct and complete. The comment was made that the disorders are not necessarily listed in order of importance as, for example, in the fairly extensive response, shown on page 7-iii, to the combination of dwarfing and widespread bone fractures. This can be explained by the fact that the program performs a sequential scan, listing applicable conditions as they are encountered. When "weighting" is implemented as a future system enhancement (see section 8), this aspect will be improved upon.

7.3.3.2 Computer Responses to specific Cases.

In order to assess the differential diagnostic ability of the application, some problem cases were presented to the system. In some the diagnosis had been established beforehand; consequently, a comparison could be made with the suggestions from the computer. Other cases were undiagnosed and were regarded as problematic. Some examples will now be presented. These patients were seen and examined by Dr Colin Wallis during a survey of genetic diseases on the island of Mauritius, undertaken in the early part of 1987.

7.3.3.2.1 Case No. 1 T.L

This 12 year old dwarfed female patient manifested long bone asymmetry and platyspondyly. Multiple enchondromata in the tubular bones, long and short, were a cardinal radiographic feature.

Subsidiary pages 7-vii to 7-xiv show the computer responses to the combination of manifestations shown in Fig 7-13 at the top of sub-page 7-vii. A discussion follows on page 7-6, immediately after these sub-pages.

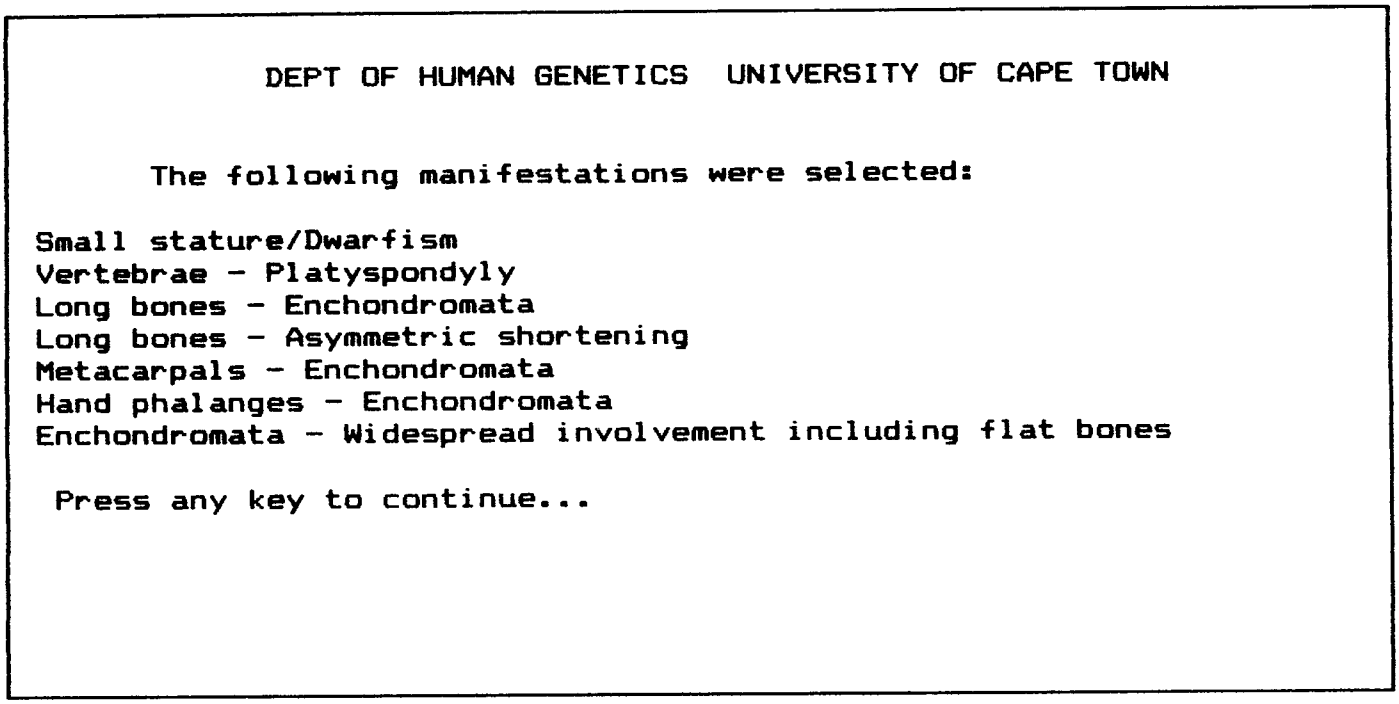


Fig 7-13

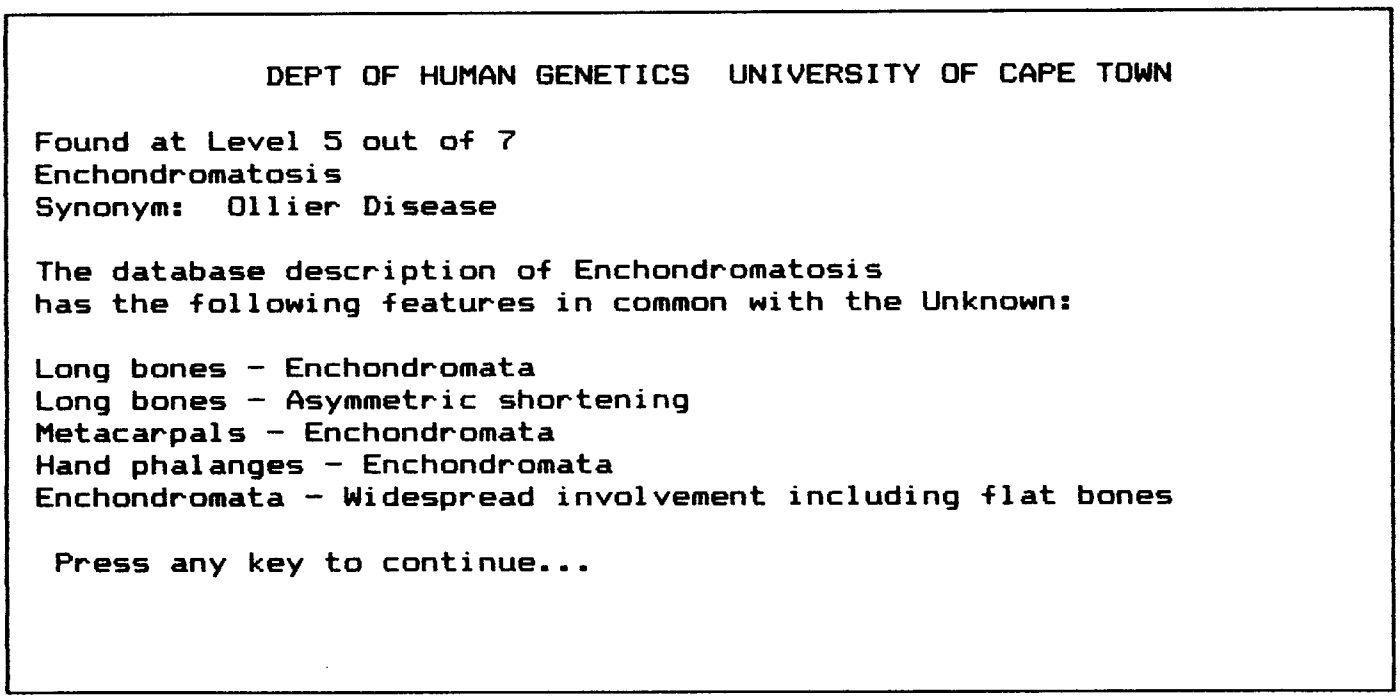


Fig 7-14

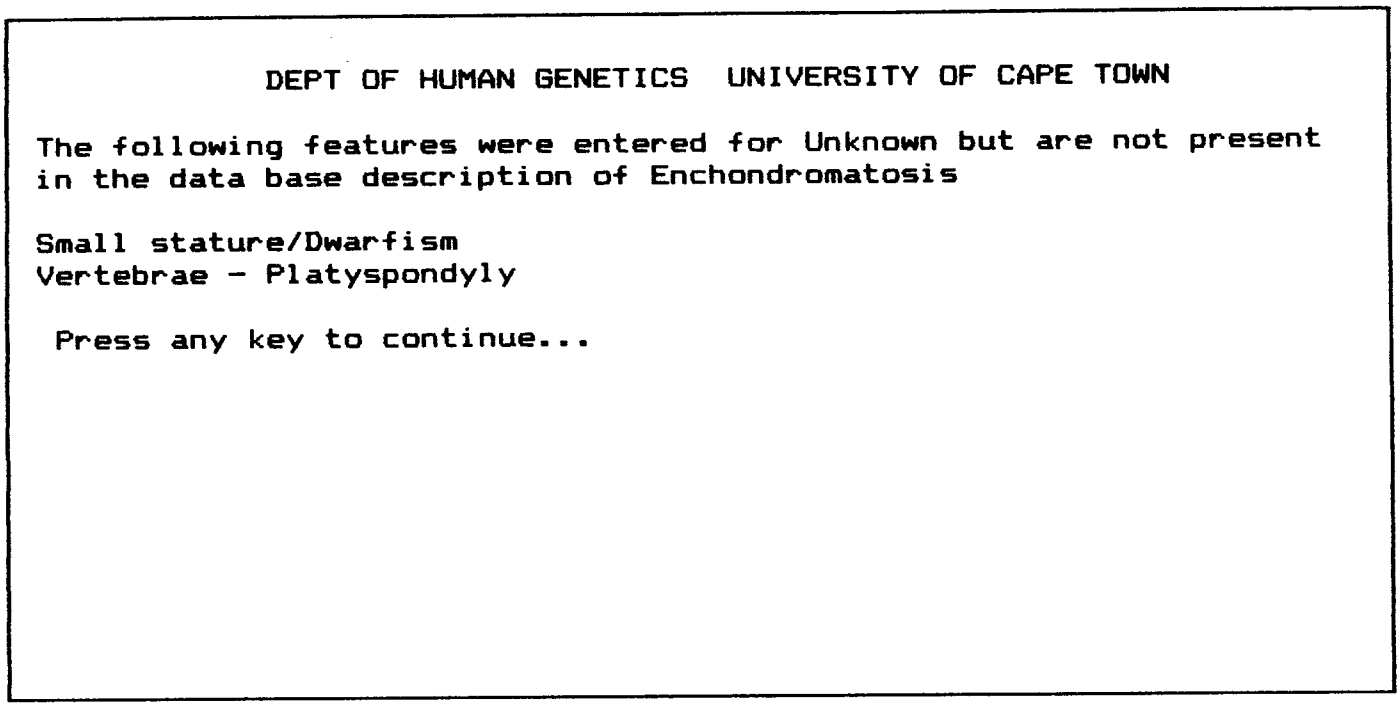


Fig 7-15

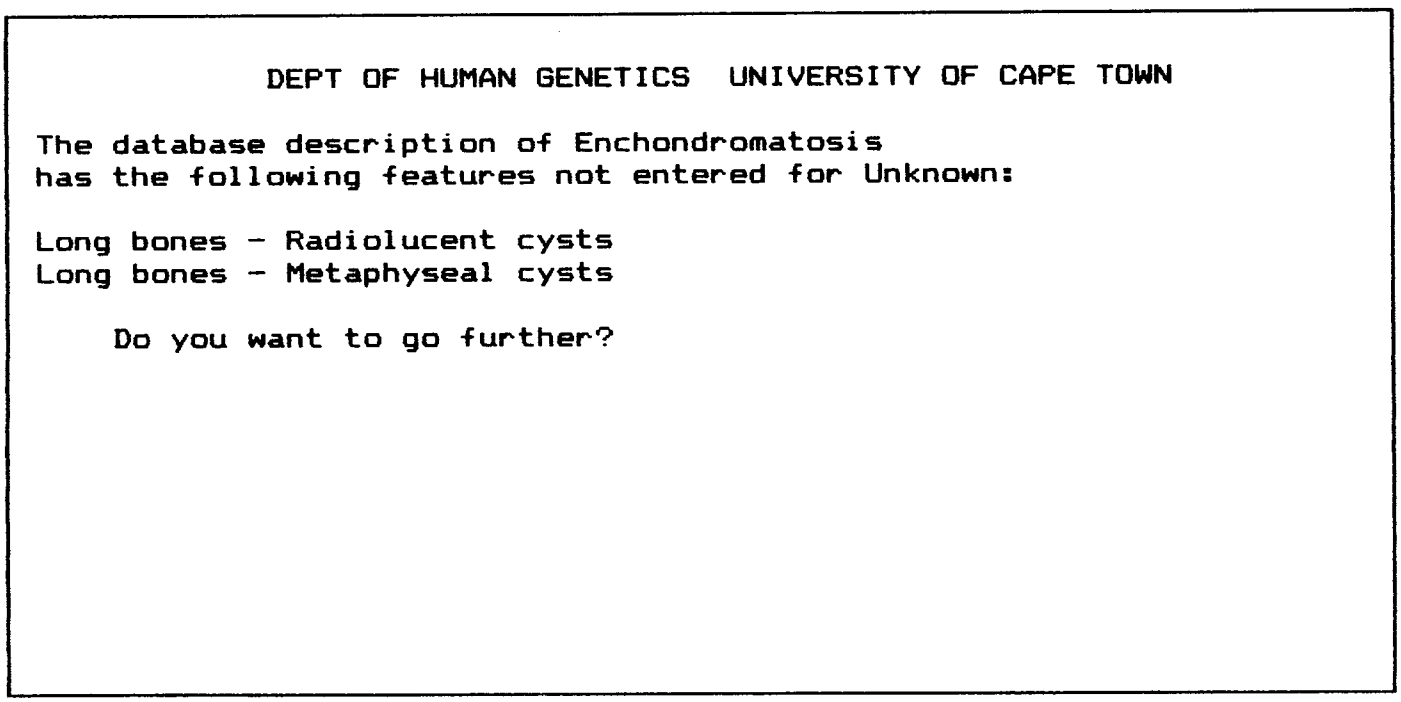


Fig 7-16

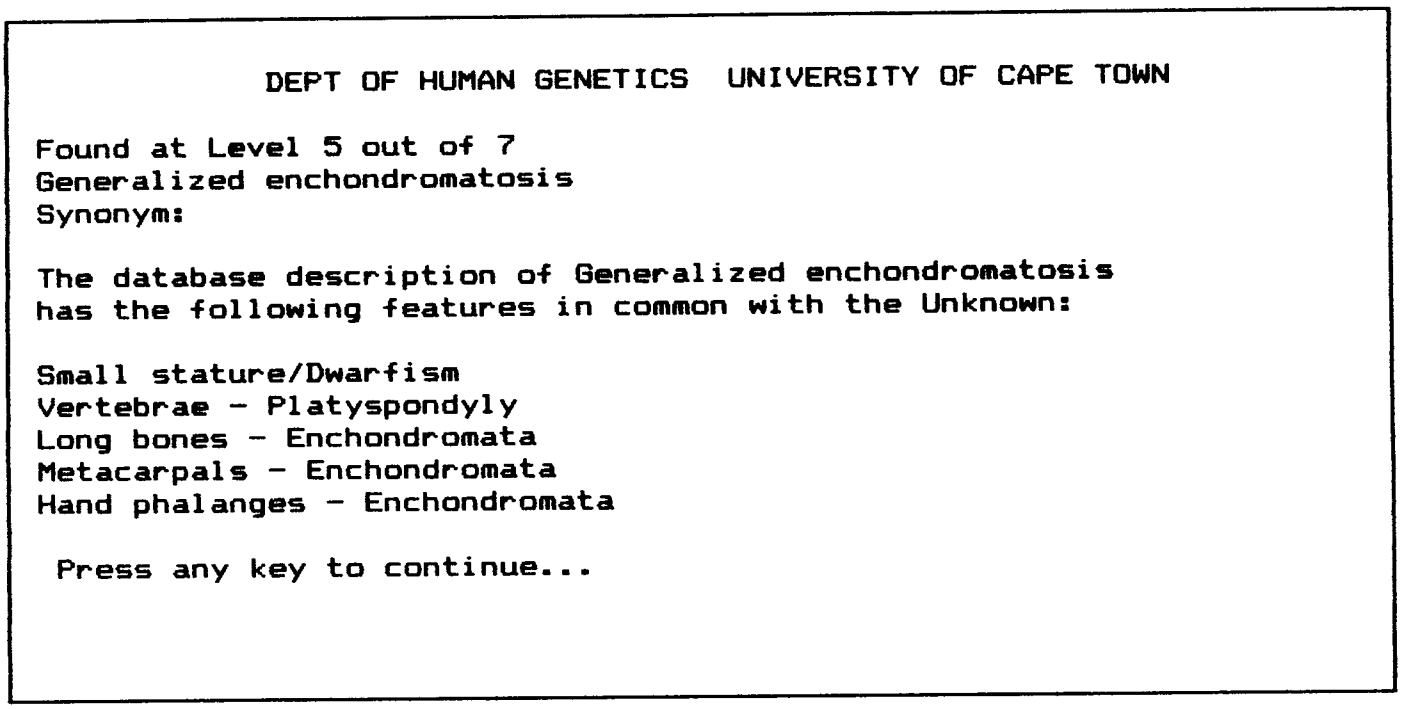


Fig 7-17

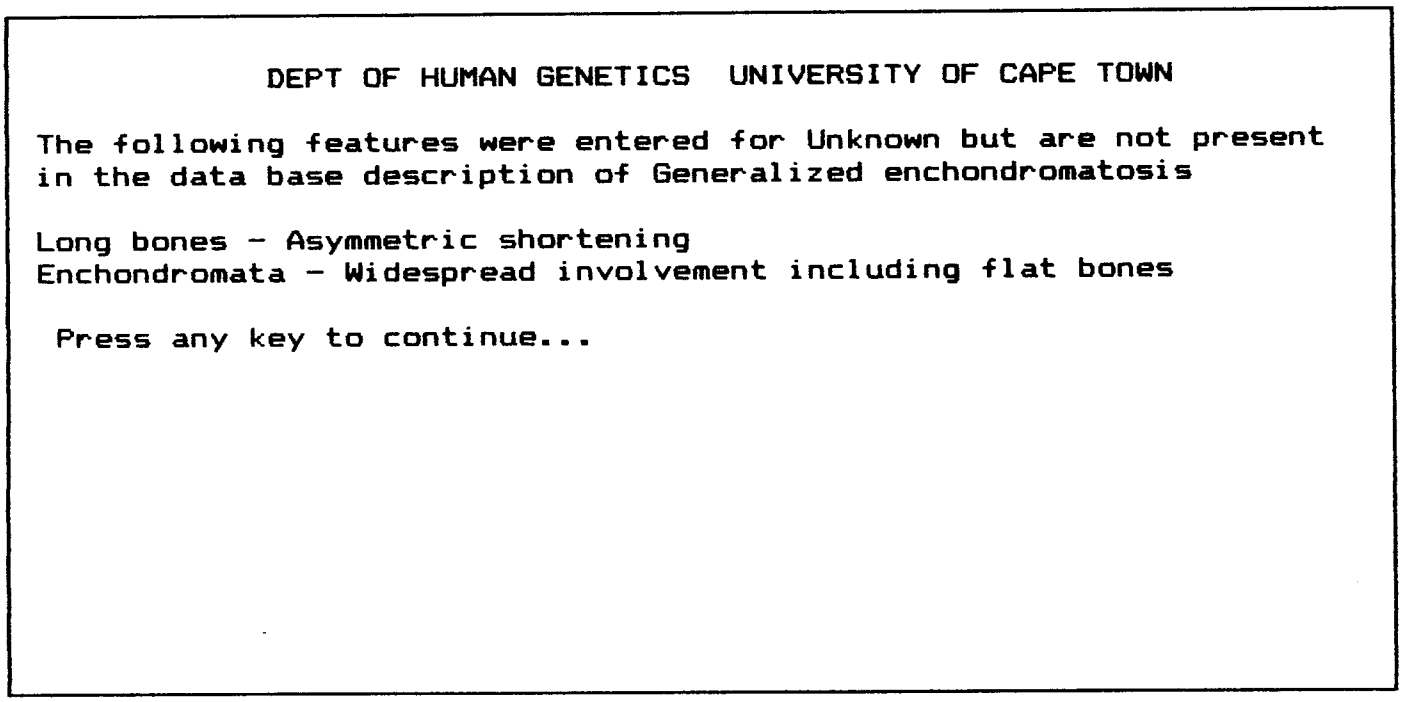


Fig 7-18

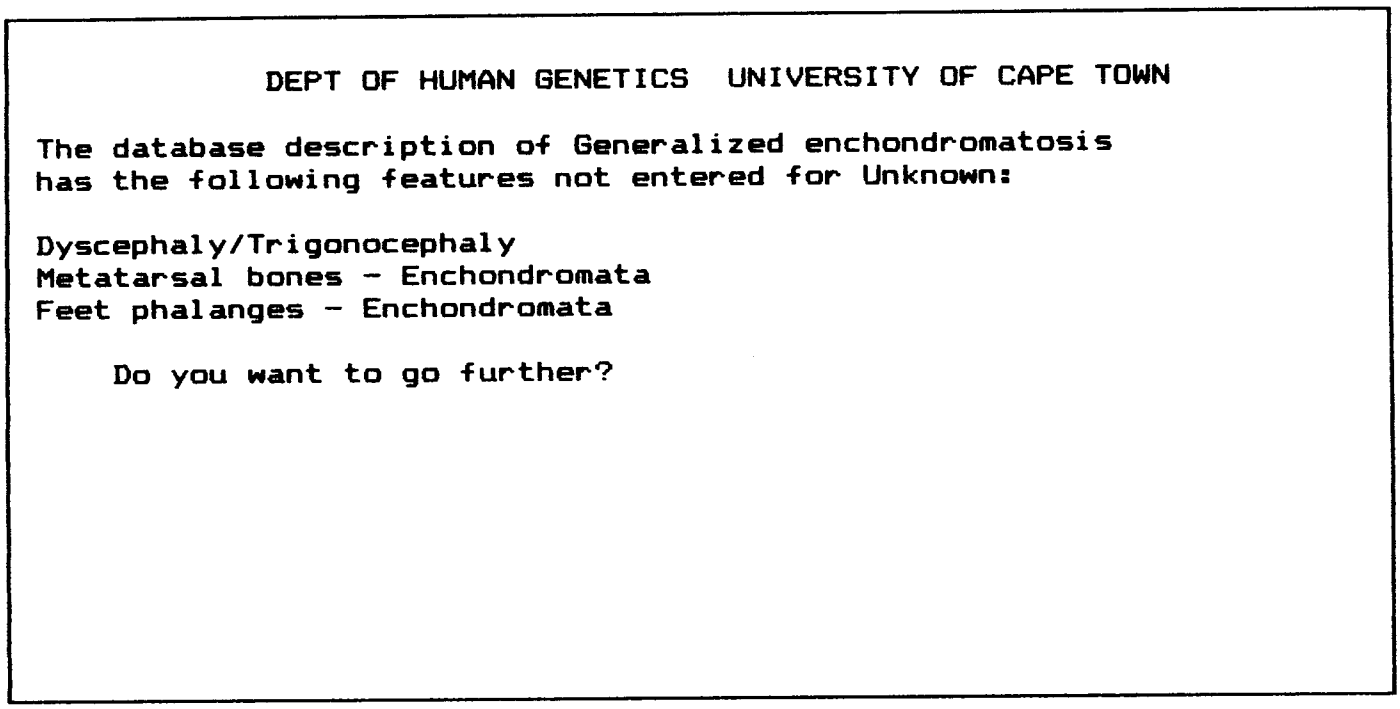

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN
The database description of Generalized enchondromatosis
has the following features not entered for Unknown:
Dyscephaly/Trigonocephaly
Metatarsal bones - Enchondromata
Feet phalanges - Enchondromata
Do you want to go further?

Fig 7-19

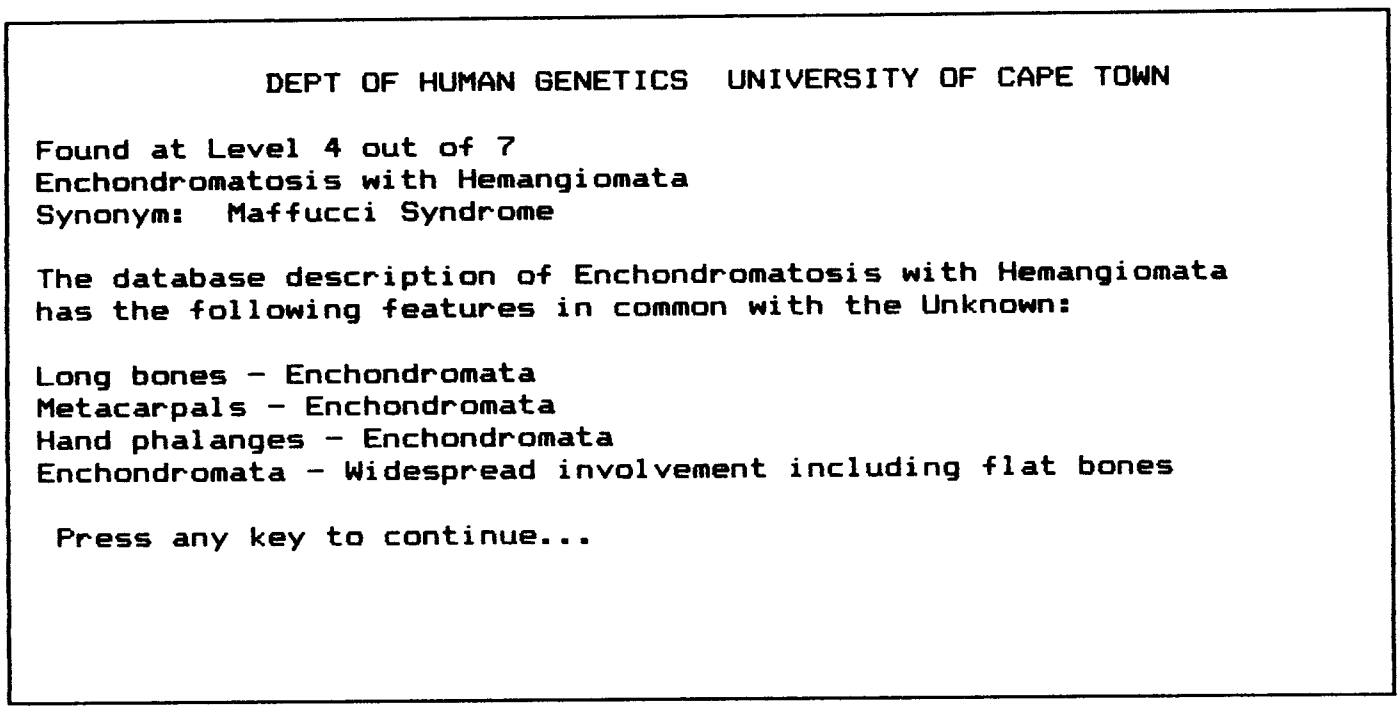


Fig 7-20

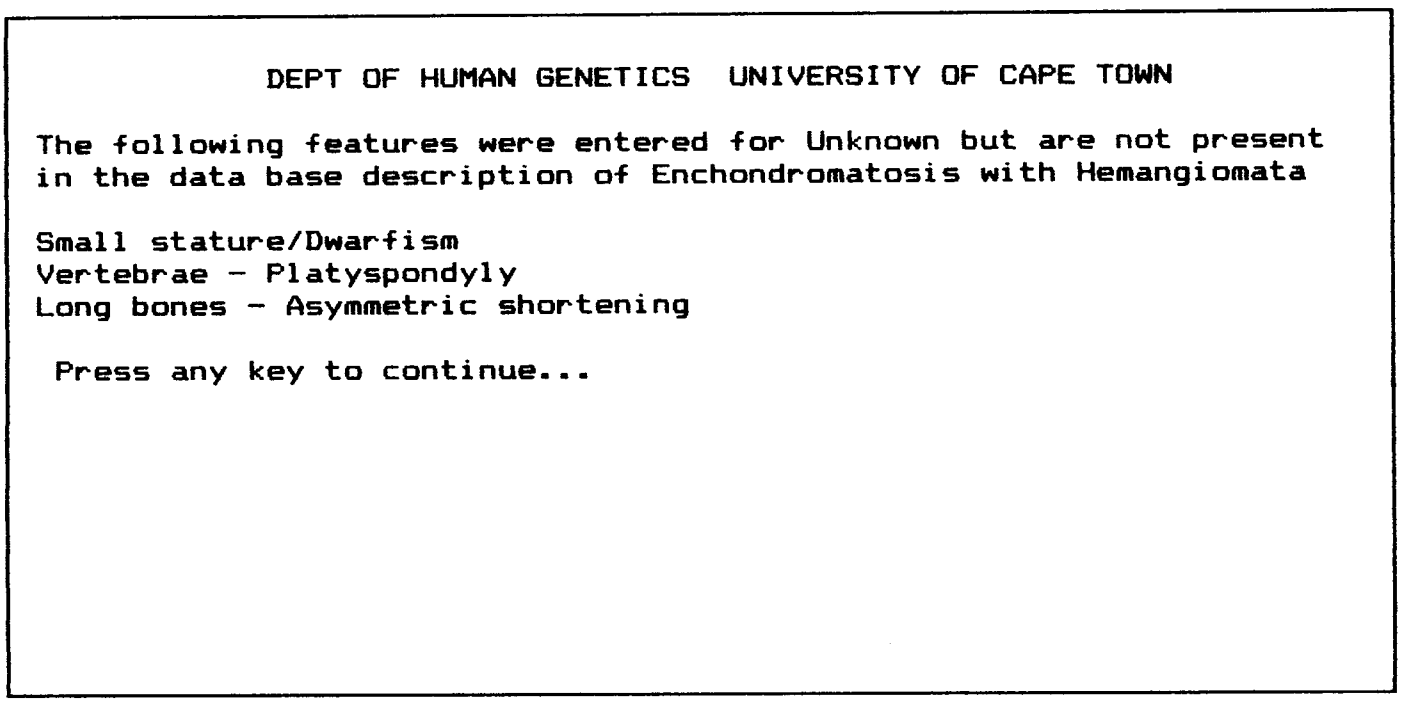


Fig 7-21

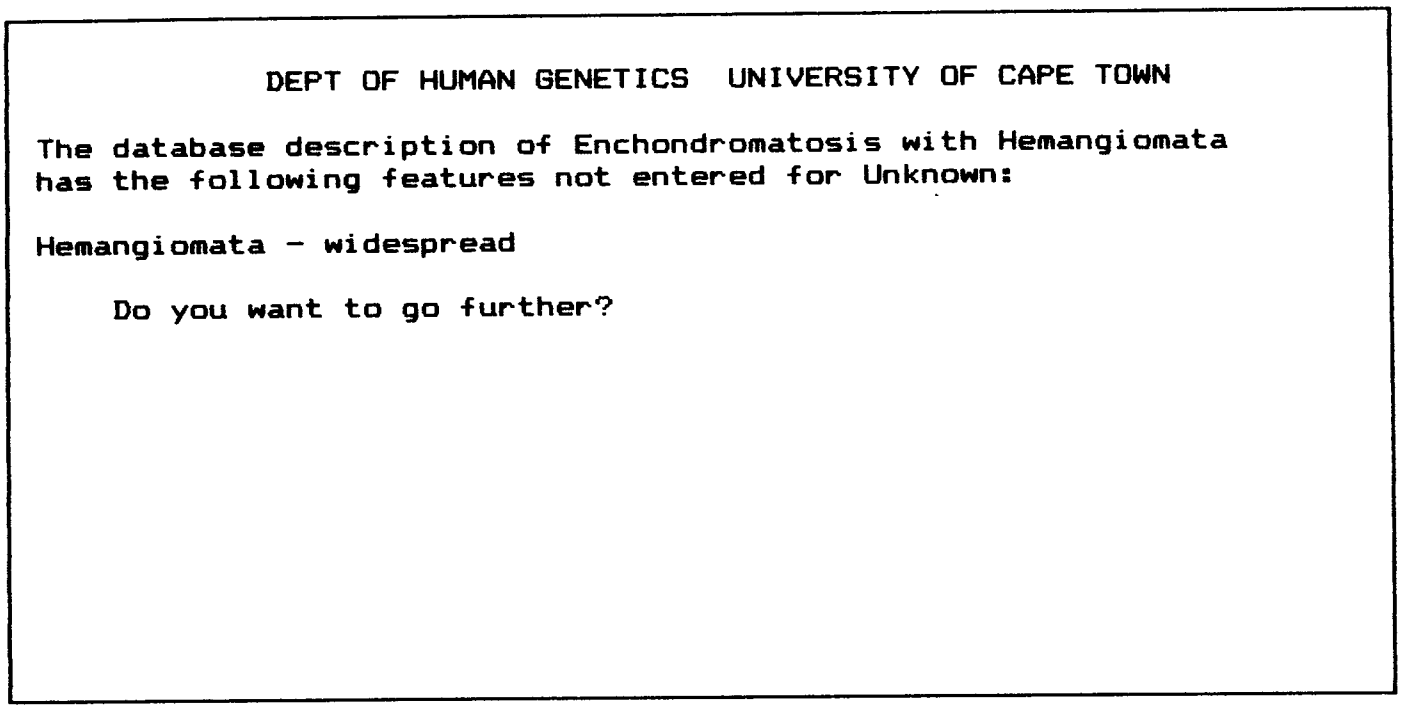

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN
The database description of Enchondromatosis with Hemangiomata has the following features not entered for Unknown:
Hemangiomata - widespread
Do you want to go further?

Fig 7-22

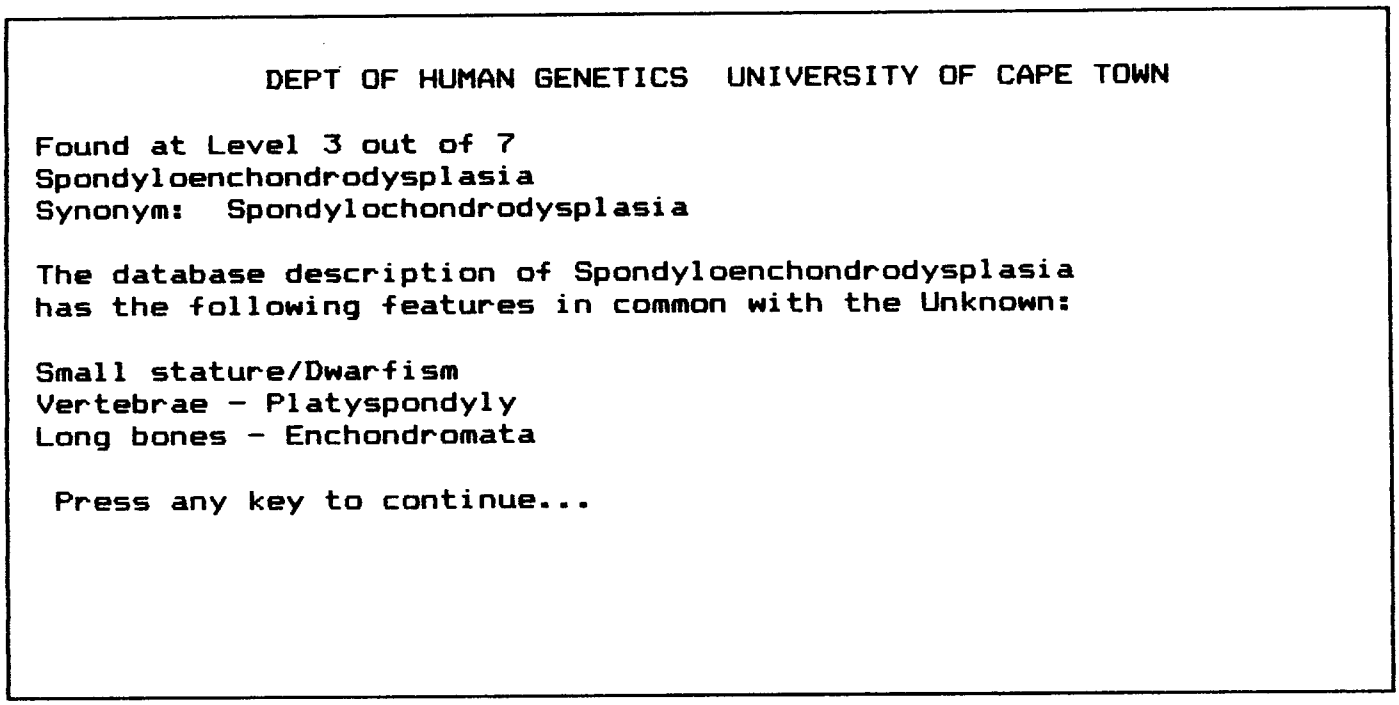


Fig 7-23

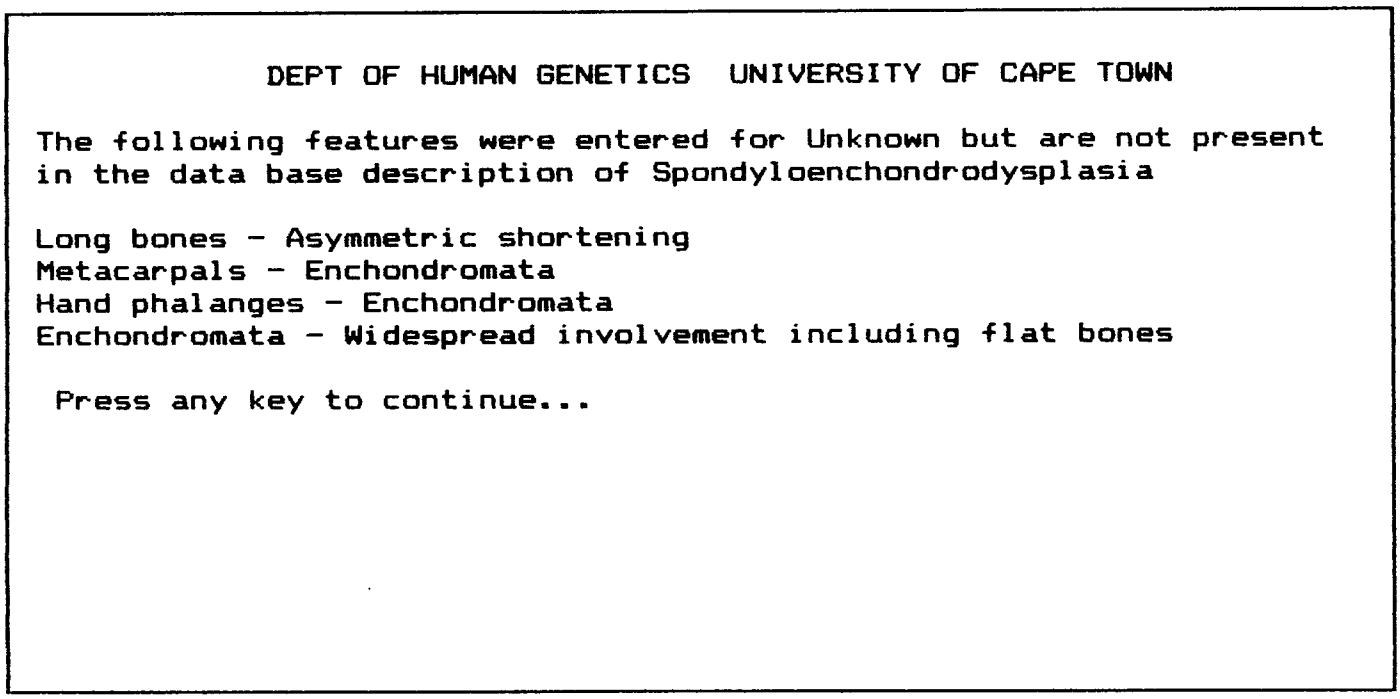


Fig 7-24

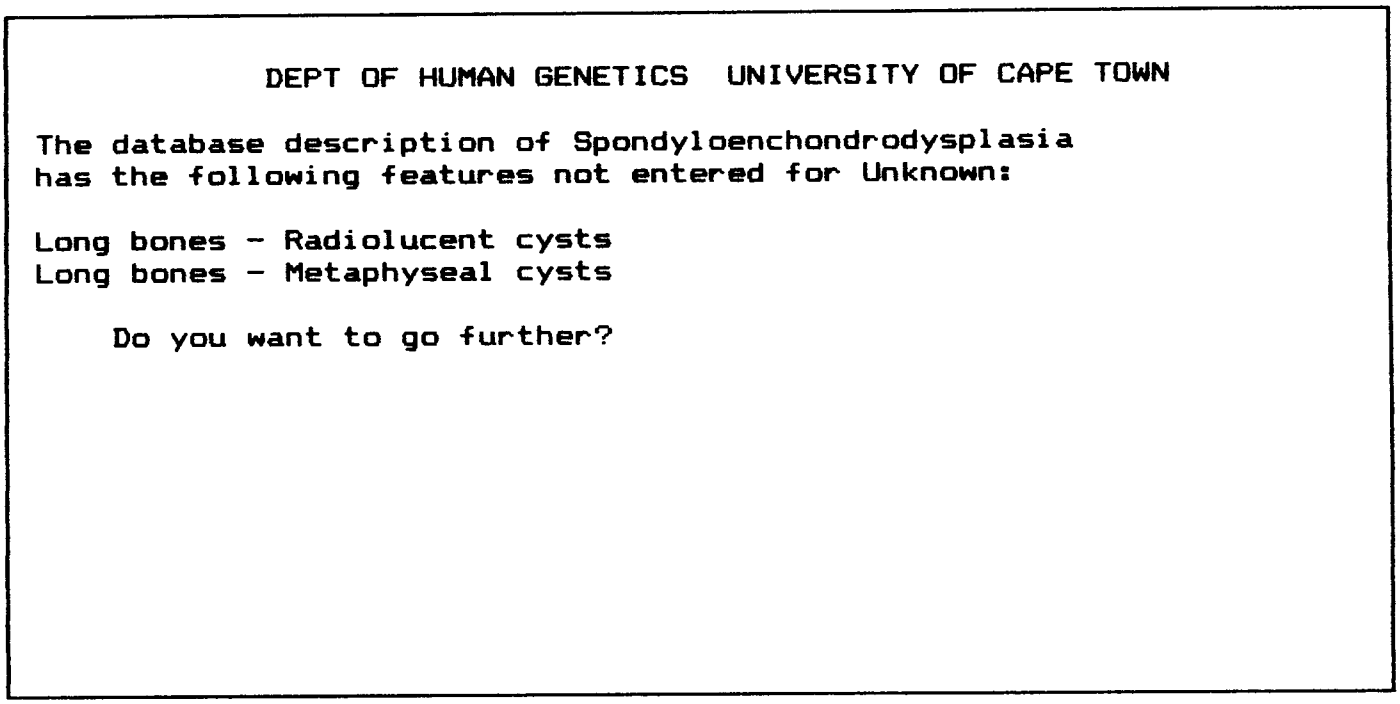


Fig 7-25

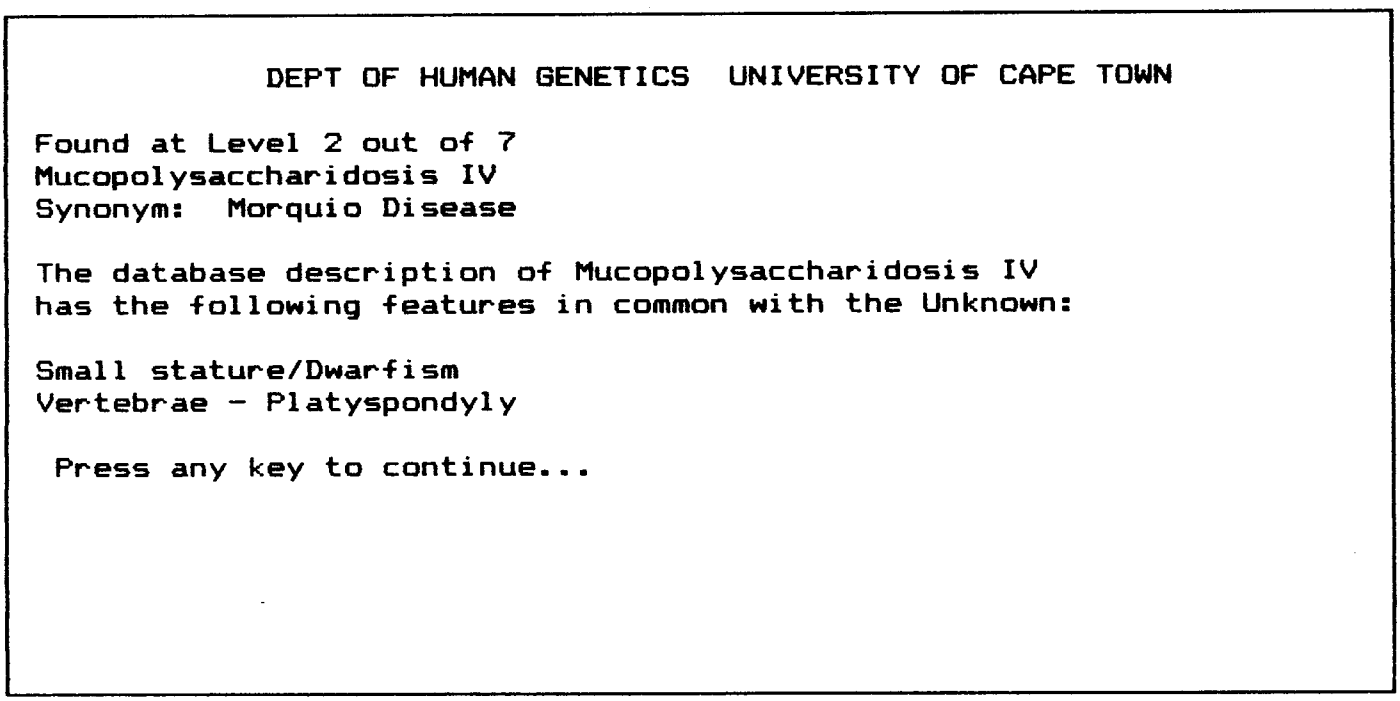


Fig 7-26

Discussion of Case No. 1.

The first "find" was that of Enchondromatosis (Ollier Disease). Five of the seven entered stigmata were known to the system to occur in this disorder (Fig 7-14). On sub-page 7-ix, Generalized Enchondromatosis is suggested at the same degree of congruence. According to domain expert Spranger, this condition is distinct from Ollier Disease. One of the distinguishing features is the presence of platyspondyly in Generalized Enchondromatosis.

The suggestion of Maffucci Syndrome, which appeared next (Fig 7- 20), could be ruled out by the absence of haemangiomata.

Spondyloenchondrodysplasia is offered next (Fig 7-23). This was considered to be a good suggestion by the domain experts. The presence of radiolucent cysts in this disorder as reported by the program result from the enchondromata and are therefore non- specific. Should this manifestation be entered as belonging to the unknown profile, Spondyloenchondrodysplasia will appear sooner in the differential diagnosis. This phenomenon, where the absence of secondary manifestations from the test profile delays the appearance of an apt suggestion, should be seen as a minor disadvantage resulting from the absence of weighting. This should improve once the latter facility is implemented (see Section 8).

The suggestion of MPS IV on sub-page 7-xiv at a level of 2 out of 7, suggested that the degree of specificity had reached a low level; consequently the process was discontinued. Numerous conditions manifest dwarfism and platyspondyly. MPS IV happened to be the first of these to be encountered in the knowledge base, hence the appearance of this disorder at this particular time.

It should be noted that the database contains one further disorder which may apply to the case under consideration, namely Enchondromatosis with irregular vertebral lesions (Spranger et al, 1978). The presence of platyspondyly and the absence of other vertebral lesions in the patient make this diagnosis unlikely.

7.3.3.2.2 Case No.2 B

This 20 year old dwarfed patient had widespread bone sclerosis with a tendency to fracturing, as well as a late open fontanelle. A brother was similarly affected. There was a history of parenteral consanguinity.

Subsidiary pages 7v to 7xii illustrate the computerized differential diagnosis which resulted from the combination of stigmata listed in Fig 7-27 at the top of sub-page 7-xv. The discussion follows on page 7-7, immediately after sub-page 7-xxii.

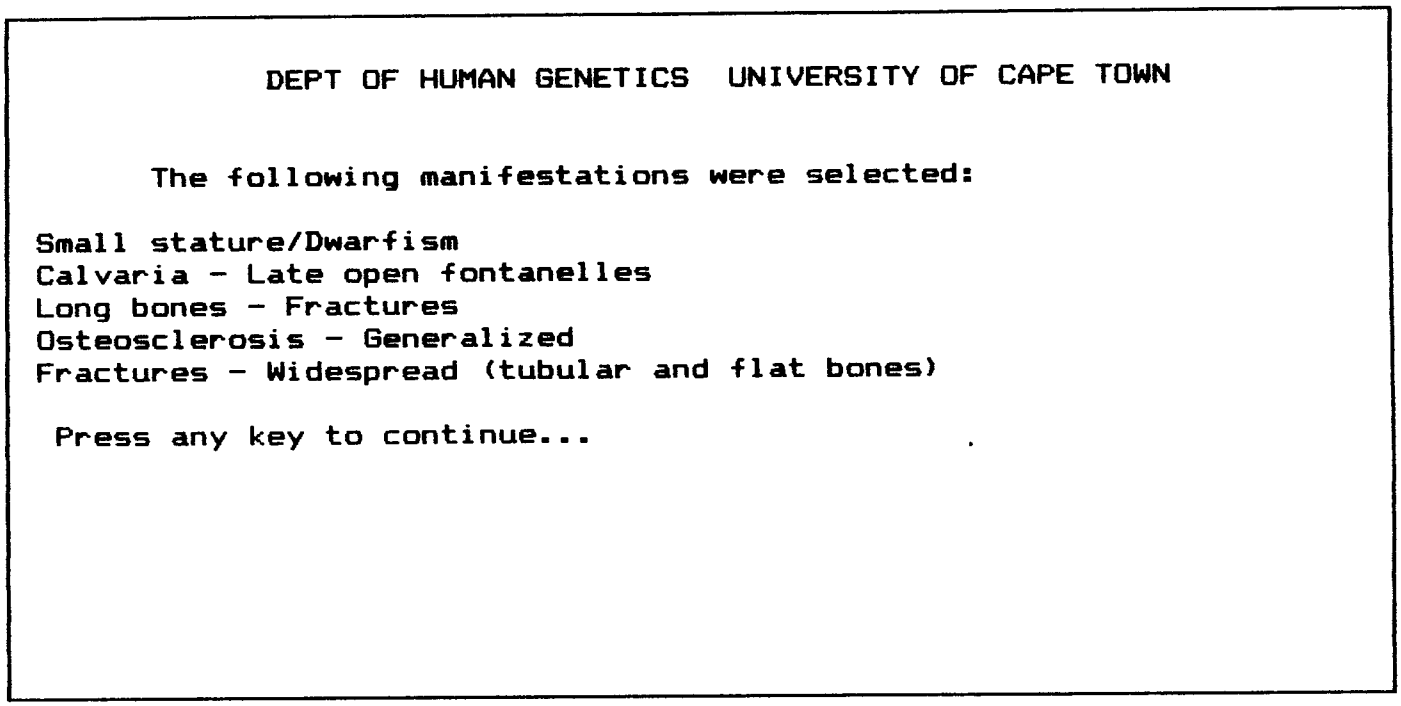


Fig 7-27

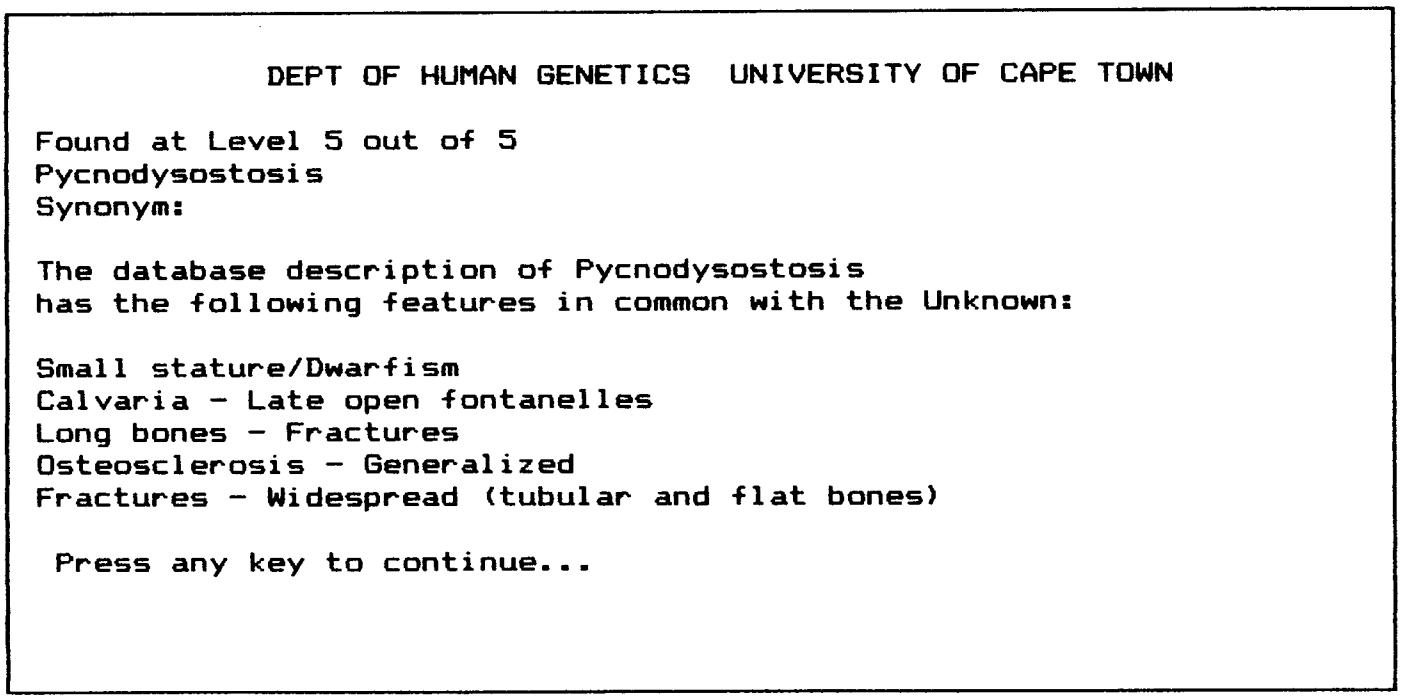


Fig 7-28

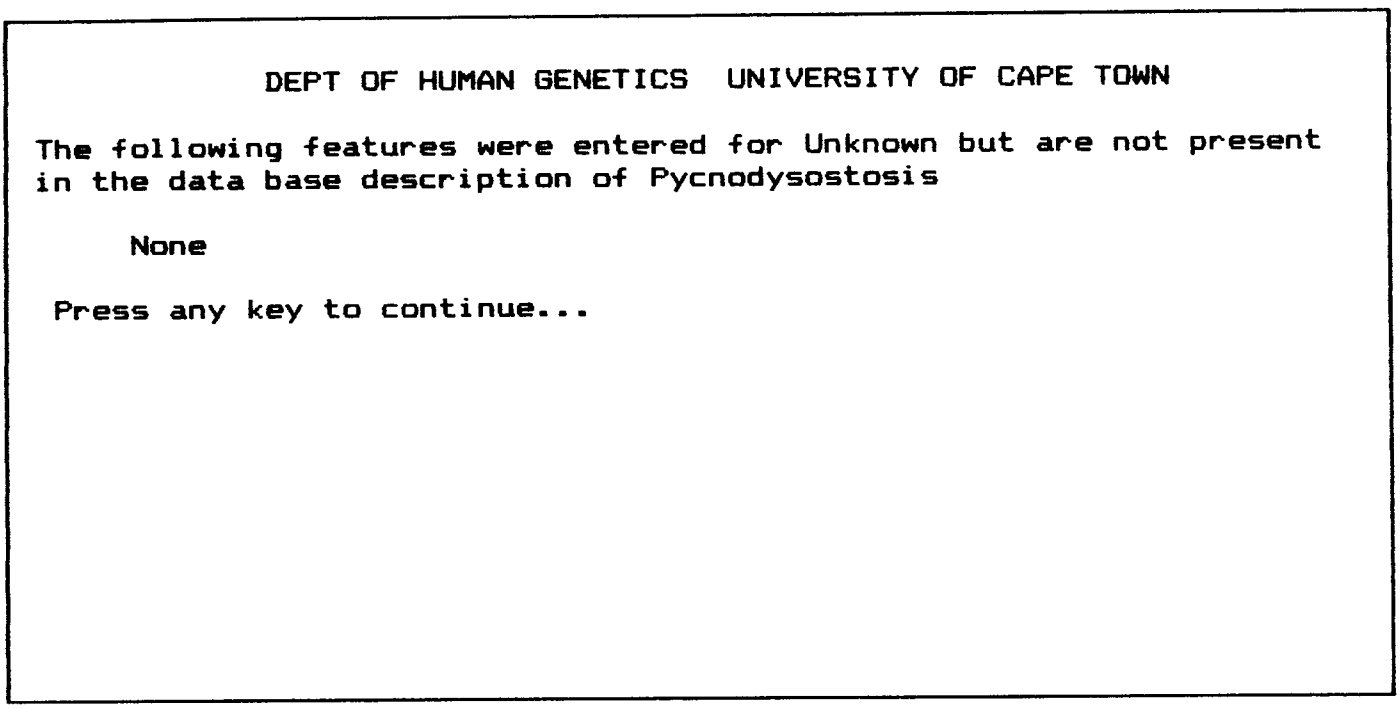


Fig 7-29

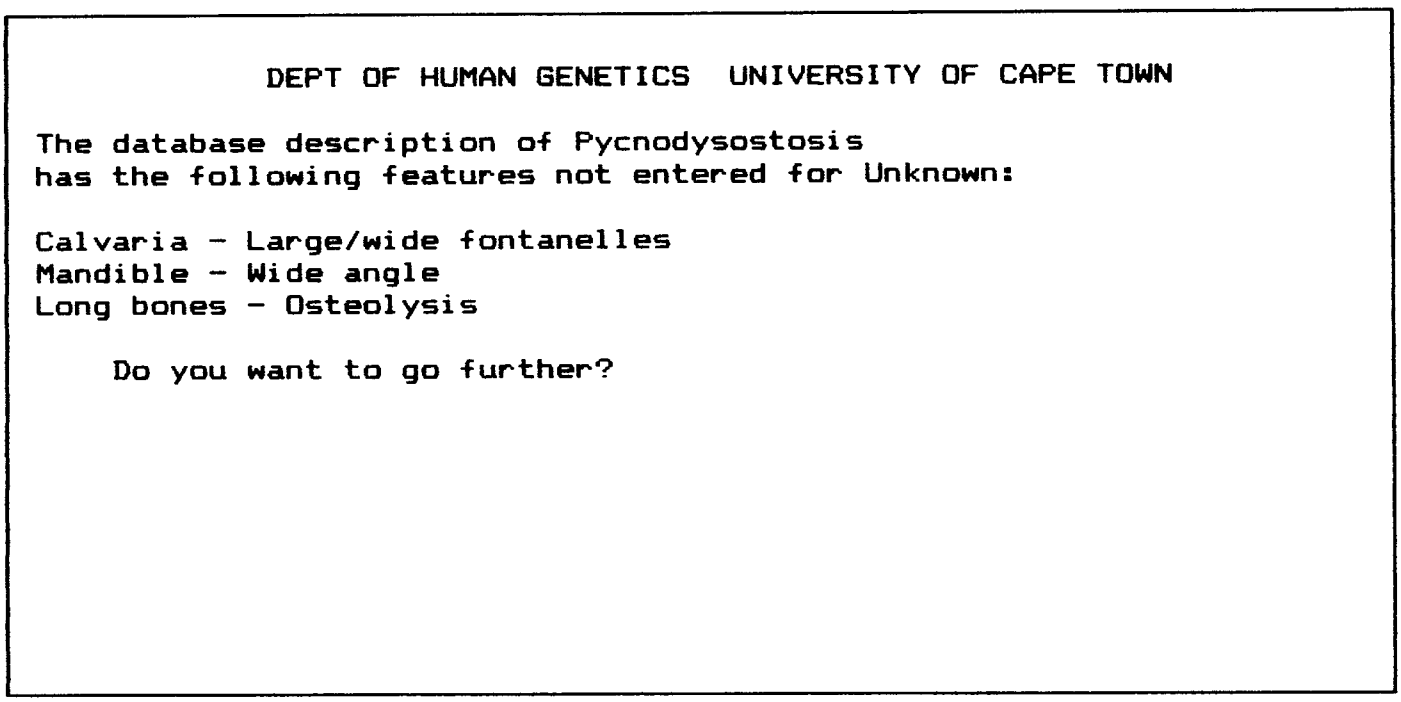


Fig 7-30

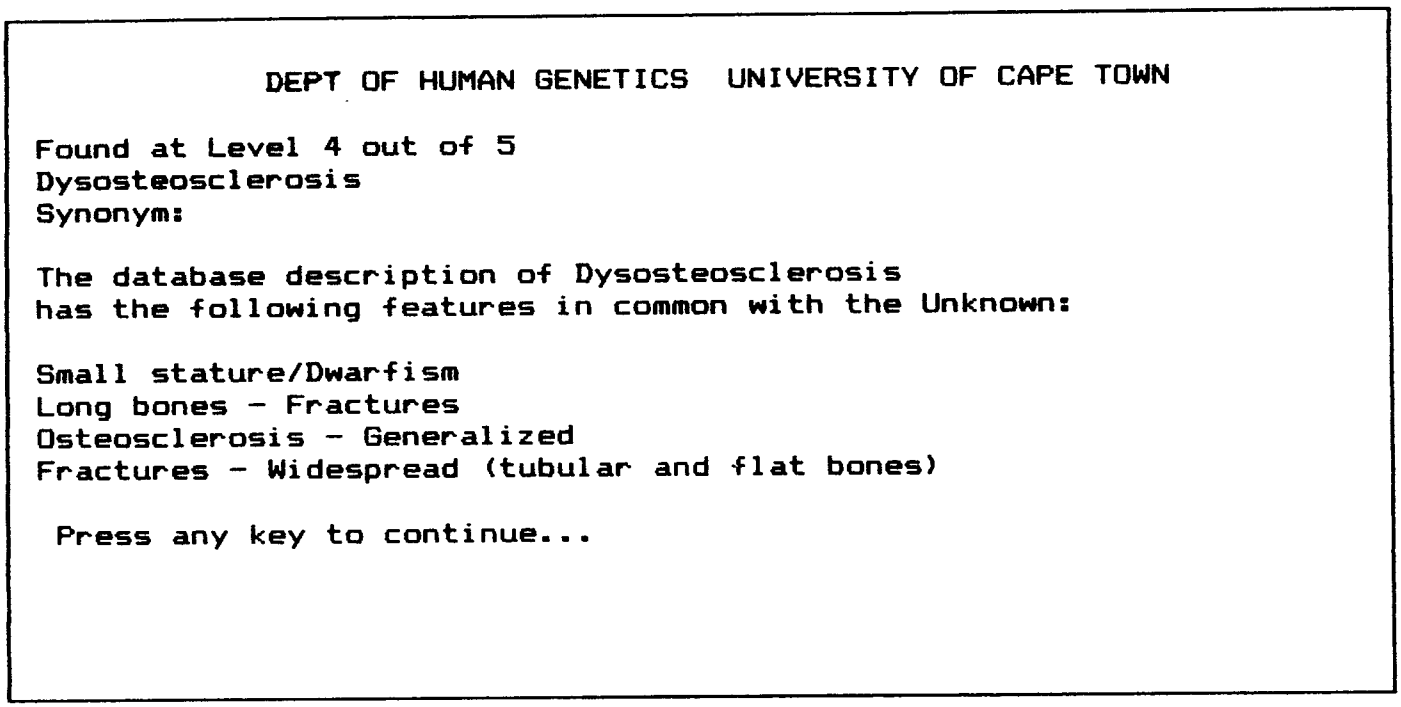


Fig 7-31

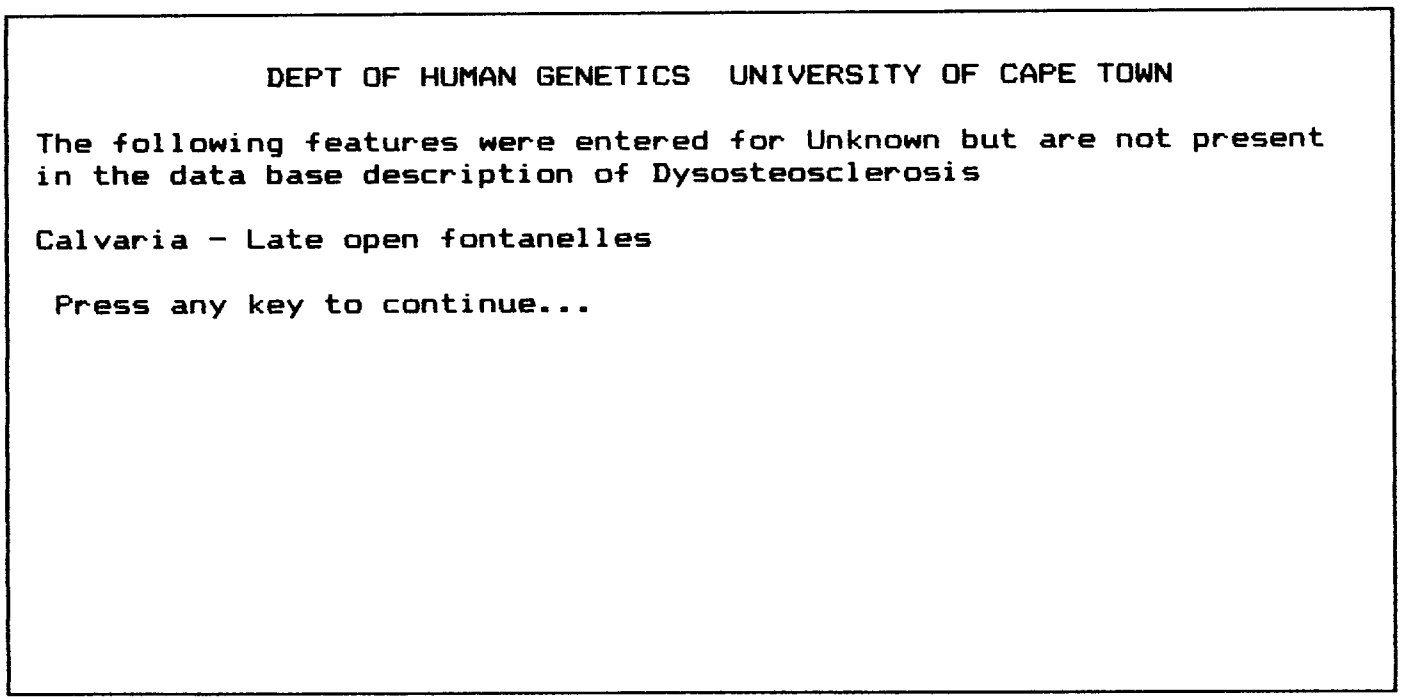


Fig 7-32

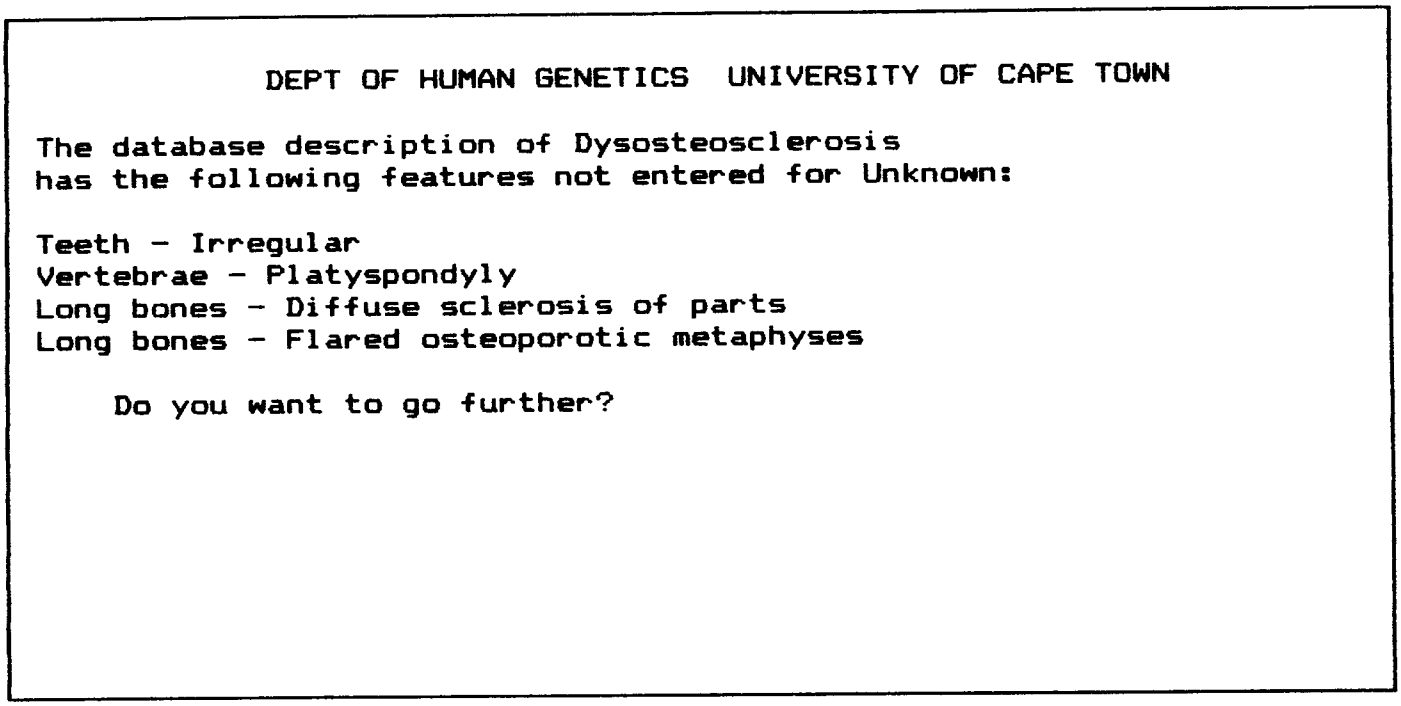

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN
The database description of Dysosteosclerosis
has the following features not entered for Unknown:
Teeth - Irregular
Vertebrae - Platyspondyly
Long bones - Diffuse sclerosis of parts
Long bones - Flared osteoporotic metaphyses
Do you want to go further?

Fig 7-33

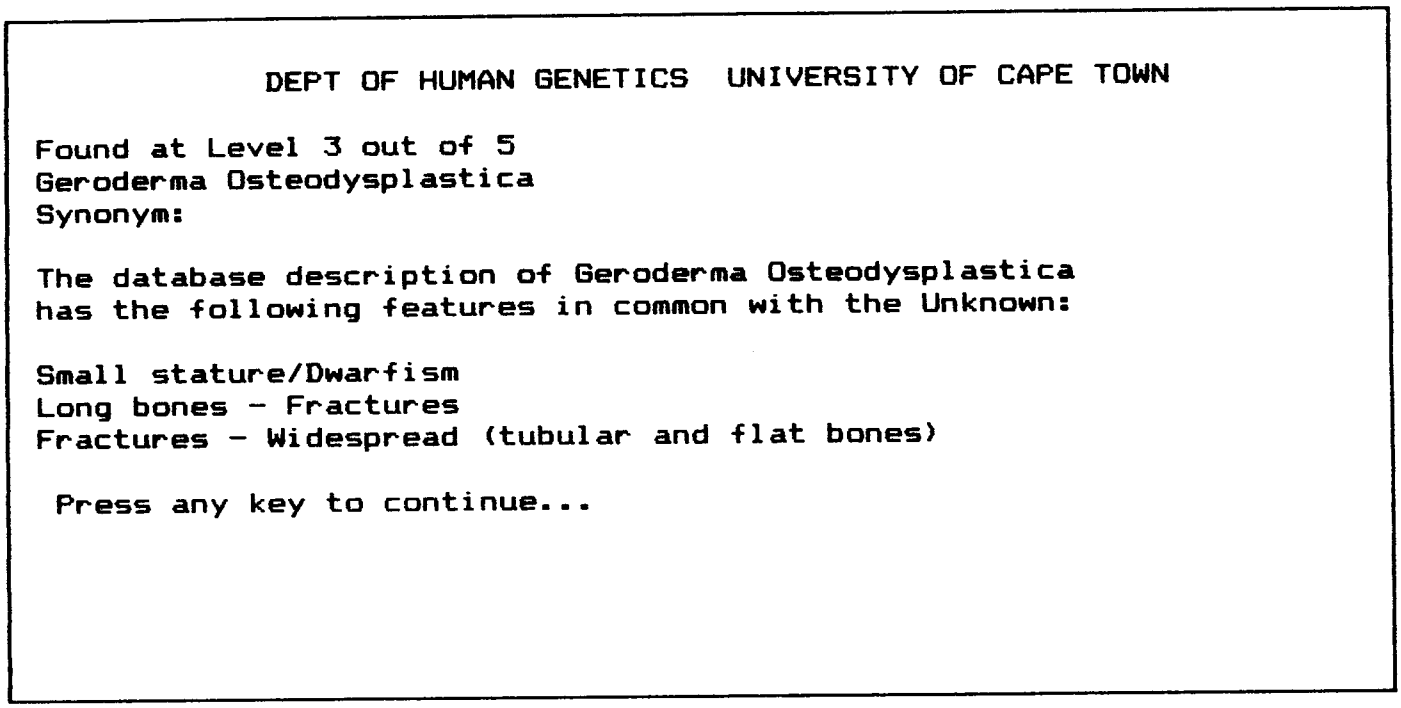


Fig 7-34

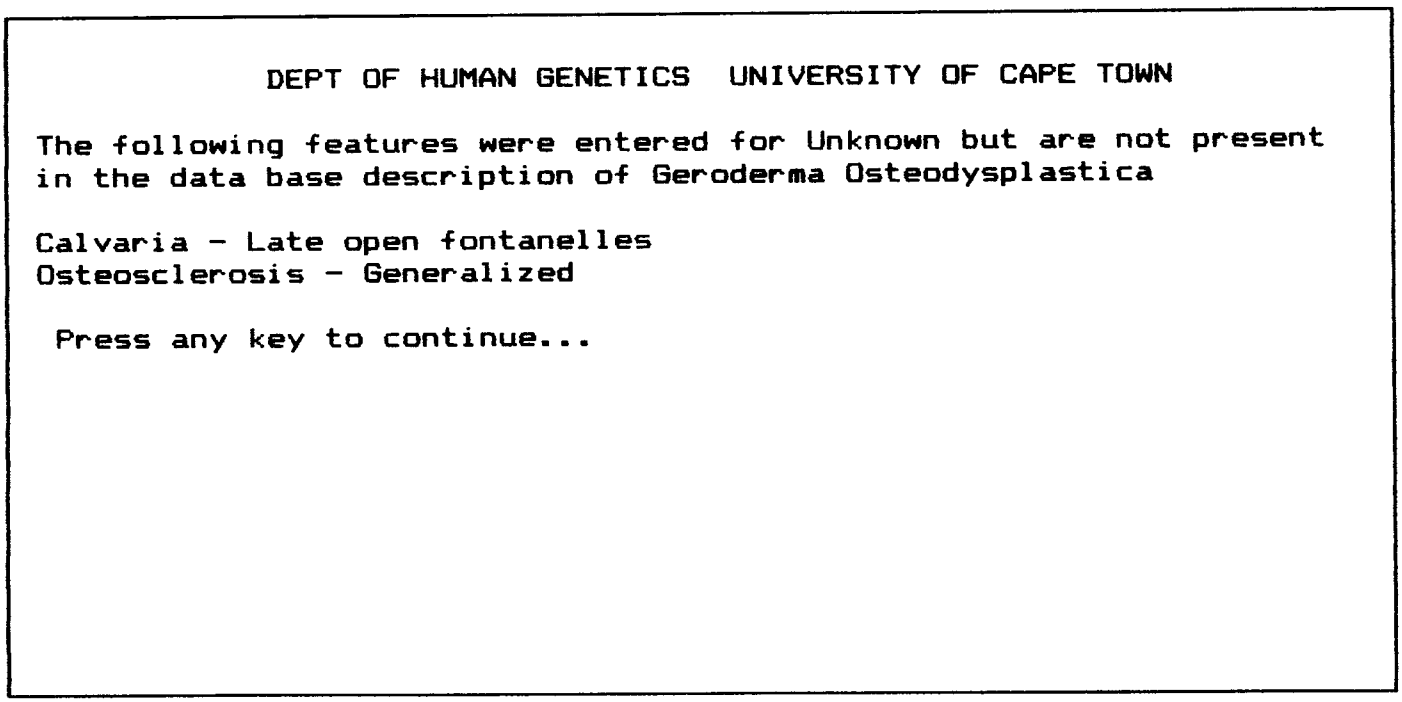


Fig 7-35

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN

The database description of Geroderma Osteodysplastica
has the following features not entered for Unknown:

Face - Unusual/Characteristic
Abnormal skin/pigmentation
Soft tissue - Hyperlaxity
Vertebral bodies - Osteopenia
Vertebrae - Platyspondyly
Joints - Generalized hypermobility
Hip - Dislocation

Do you want to go further?

Fig 7-36

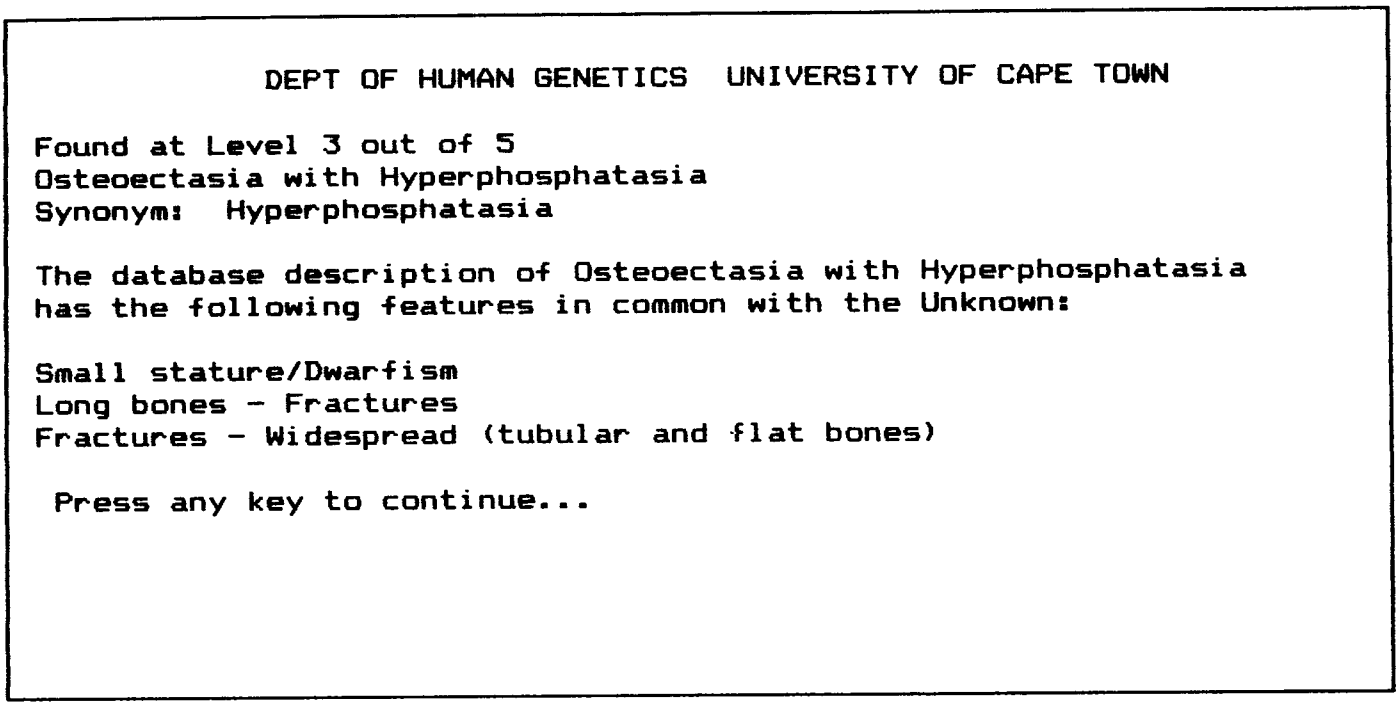


Fig 7-37

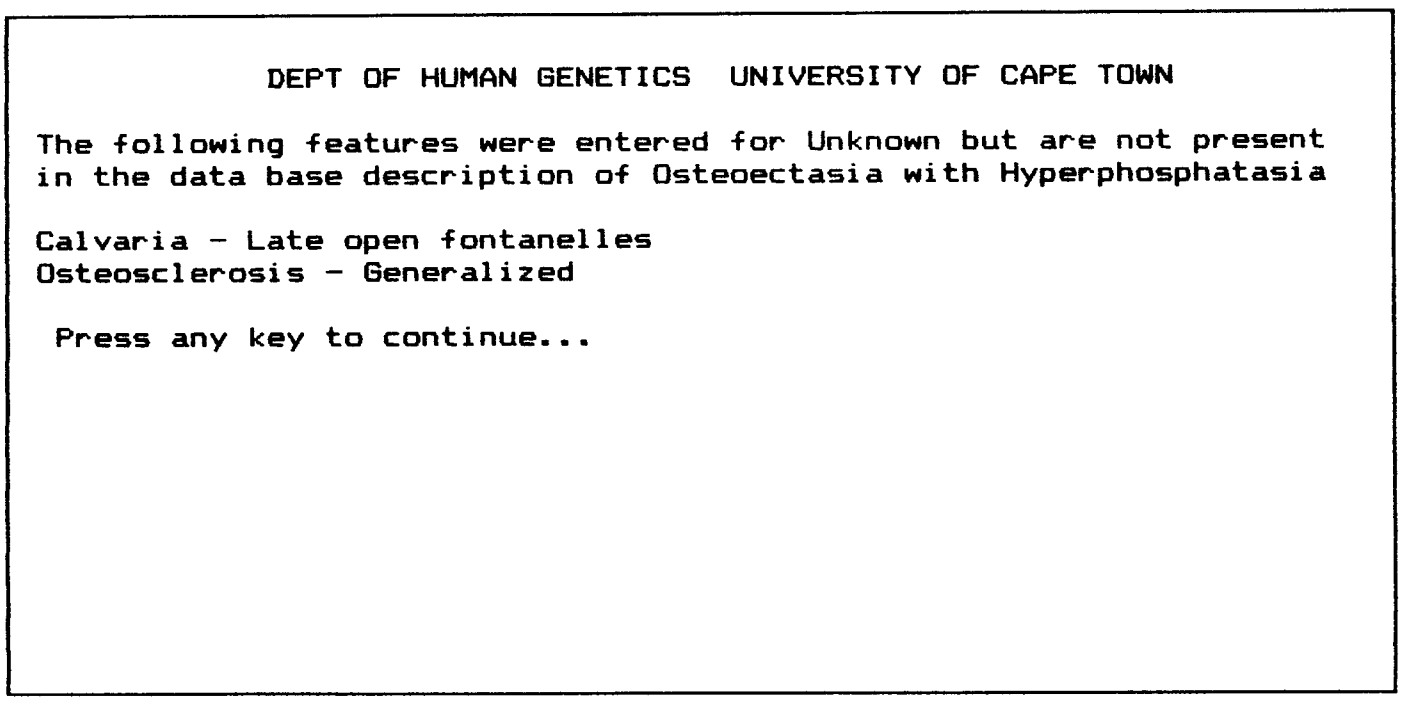


Fig 7-38

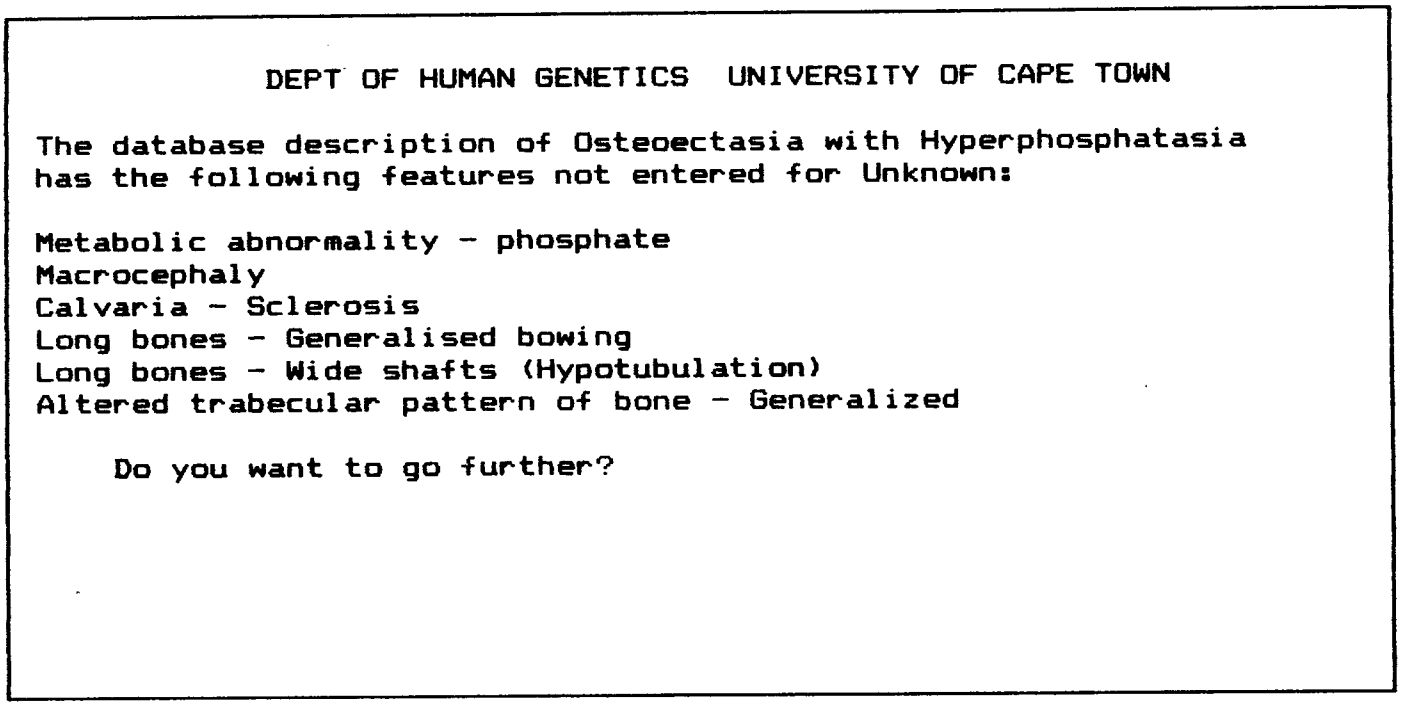


Fig 7-39

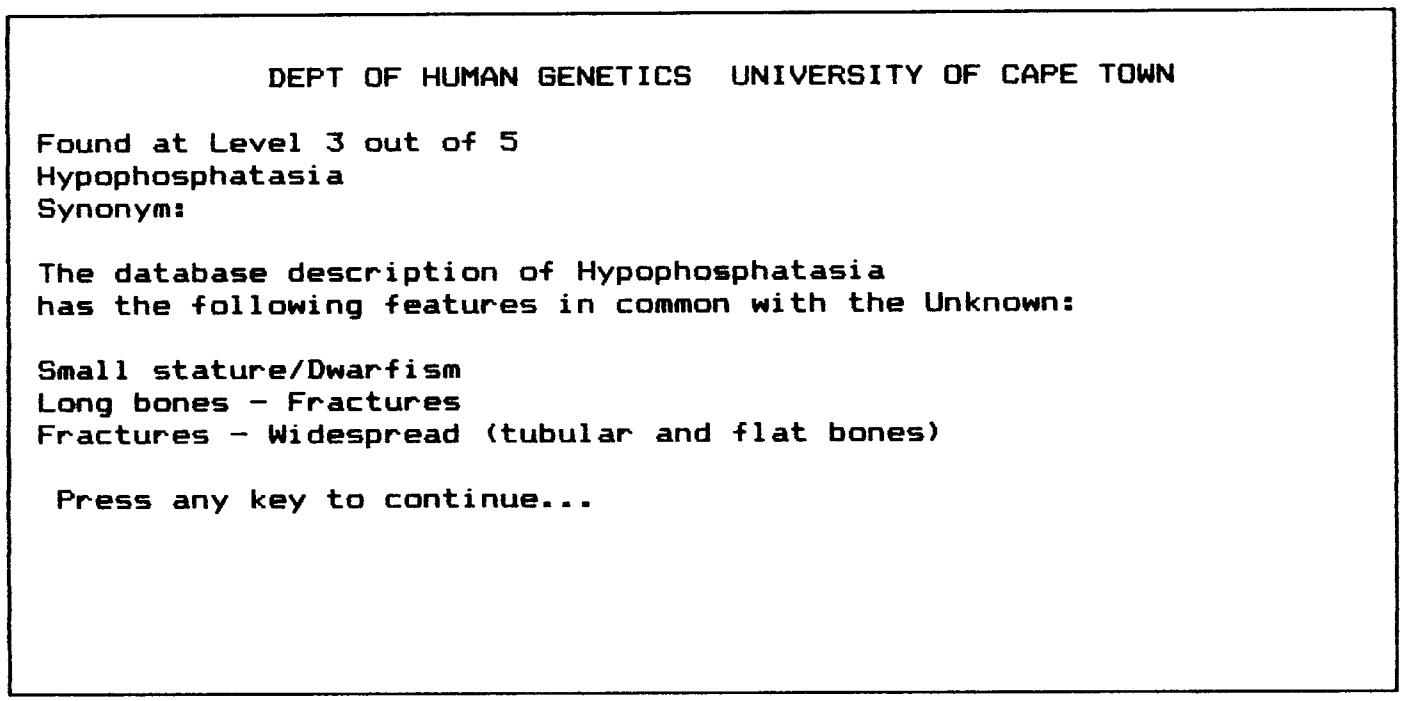


Fig 7-40

Discussion of case no.2.

Pycnodysostosis is suggested with a high level of certainty as shown in Fig 7-28. This diagnosis had been entertained before the computer run as being the most likely in this case. As a matter of interest some further suggestions were obtained from the computer. These are listed on sub-pages 7-xvii to 7-xxii. Although less likely, none of these were considered to be inappropriate.

7.3.3.2.3 Case No.3 G.

This 50 year old male was dwarfed, mentally retarded and displayed coarse facial features. In addition, radiological examination revealed brachydactyly, collapse of the femoral capital epiphyses, platyspondyly, thin ribs and a non-specific wavy configuration of the tibia.

The list of entered manifestations is shown in Fig 7-41 at the top of sub-page 7-xxiii. The fact that all the features could be selected from the database, is indicative of fairly high comprehensiveness at present. The computer responses are shown on sub-pages 7-xxiii to 7-xxviii. The discussion follows immediately after on page 7-8.

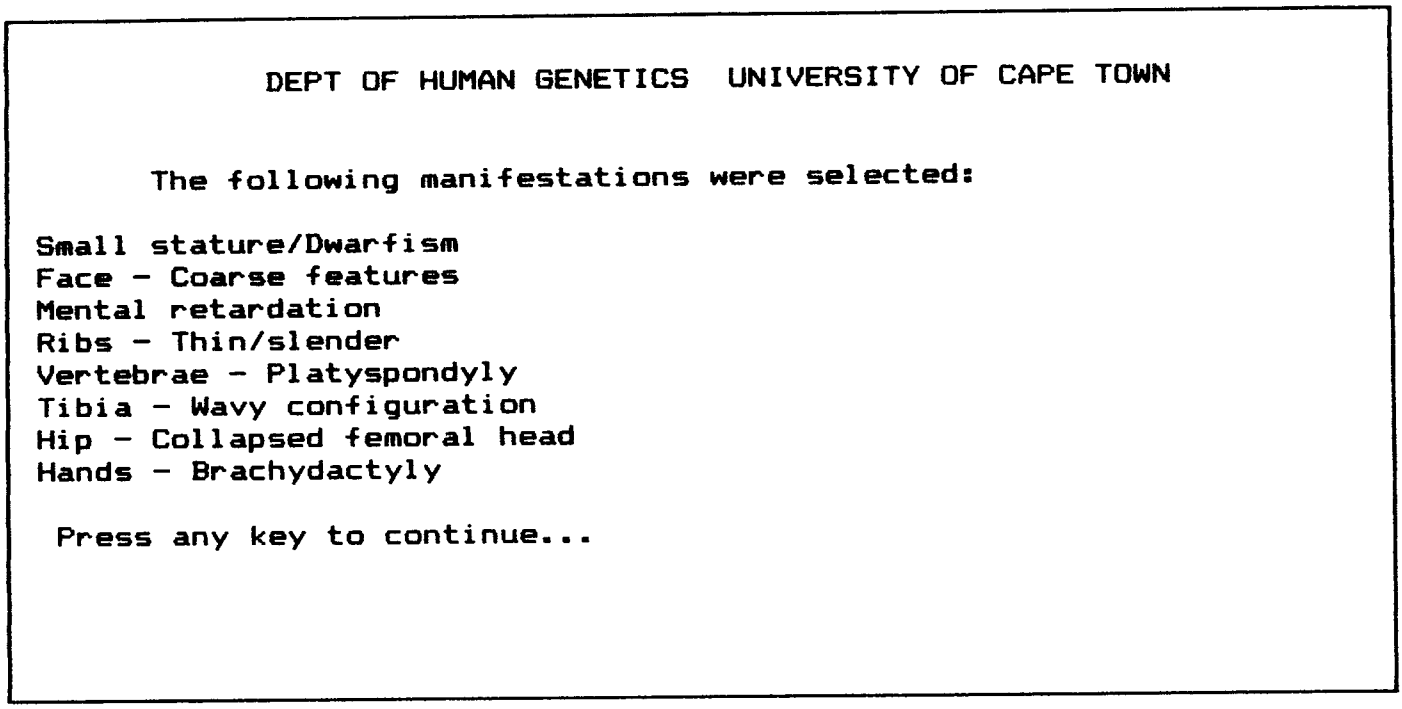


Fig 7-41

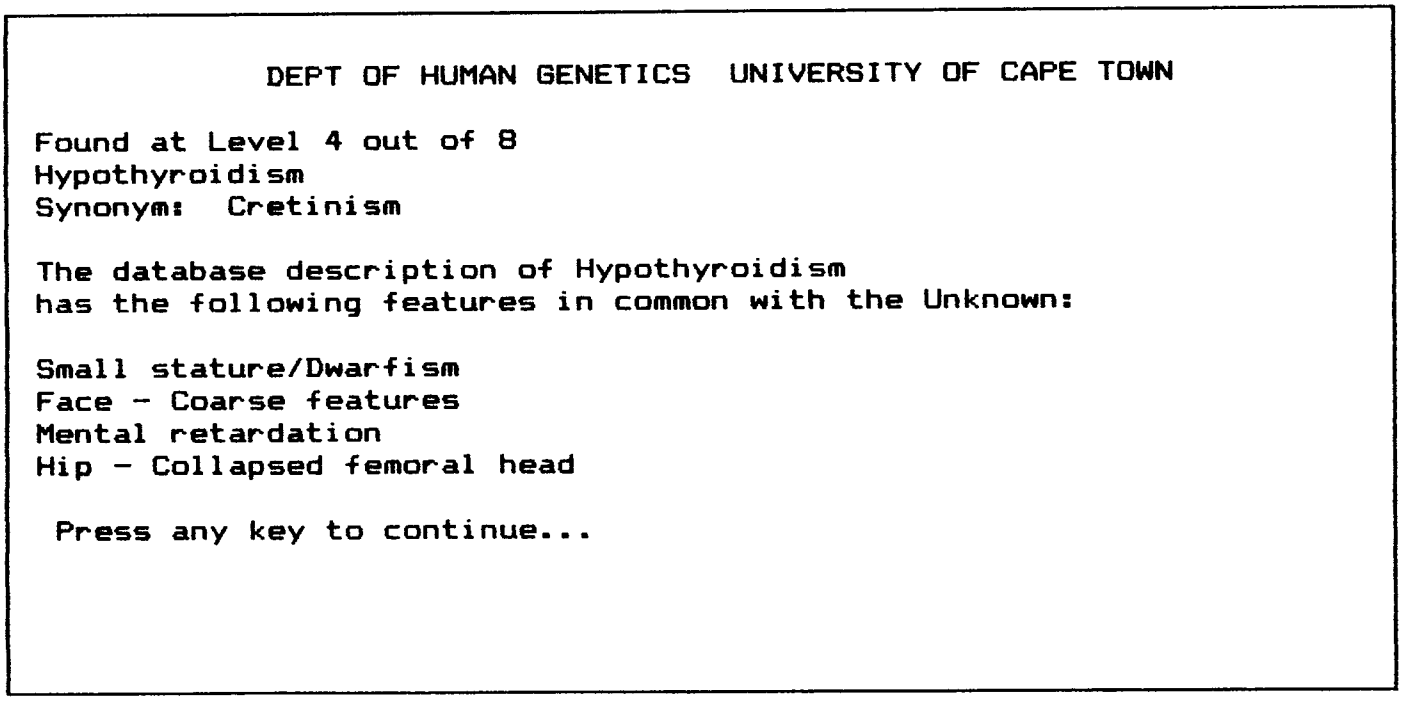


Fig 7-42

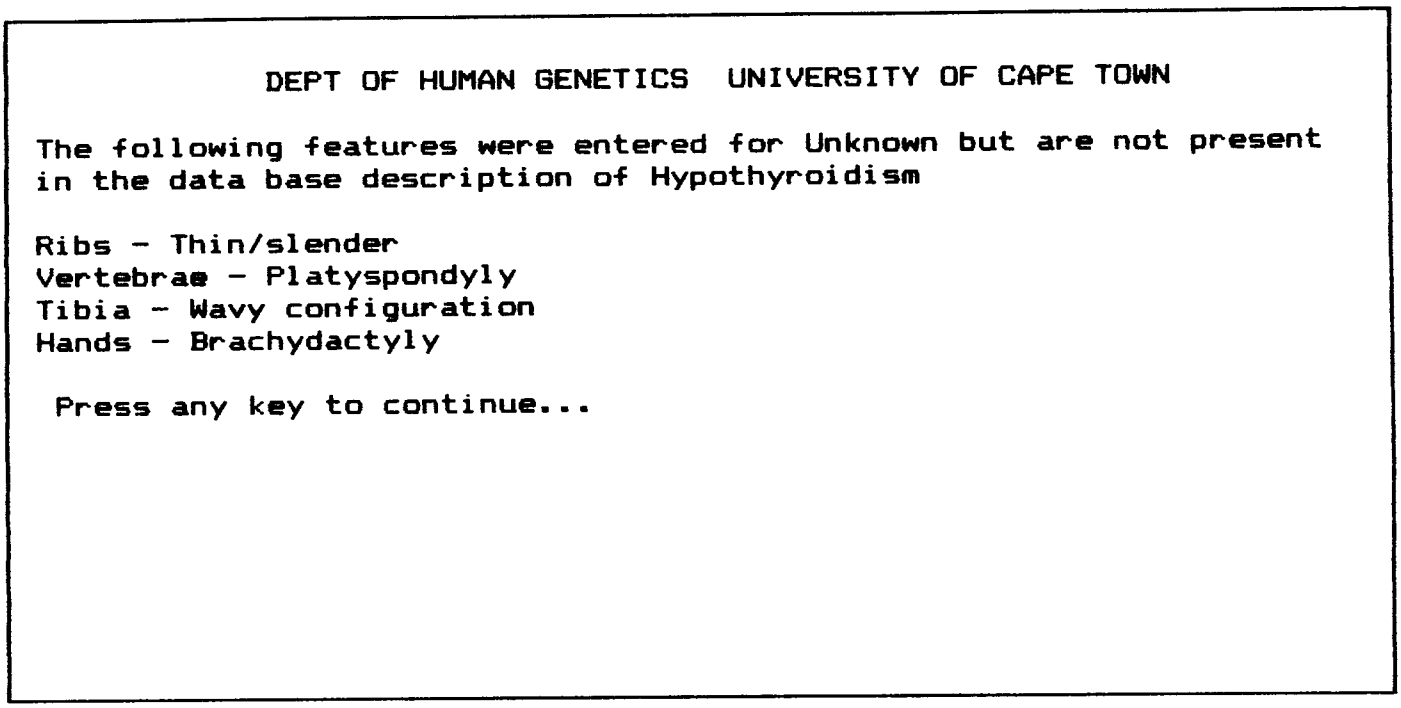


Fig 7-43

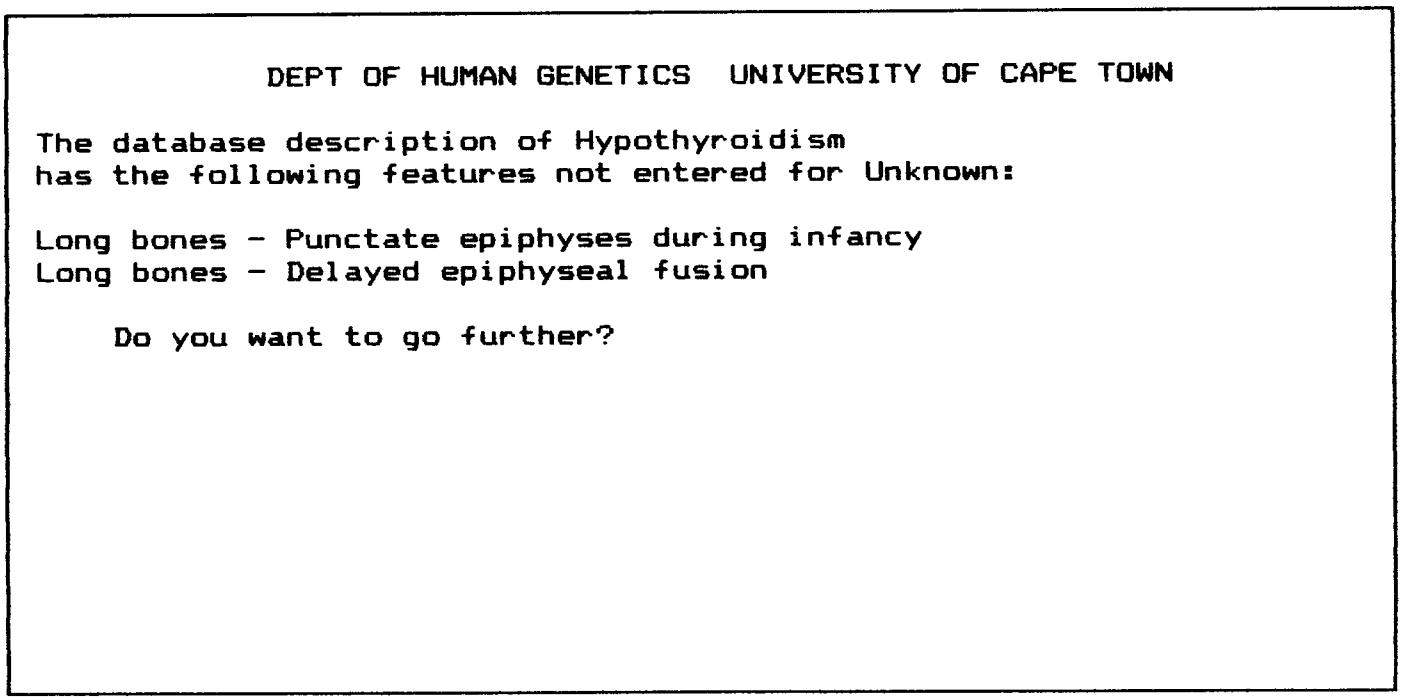


Fig 7-44

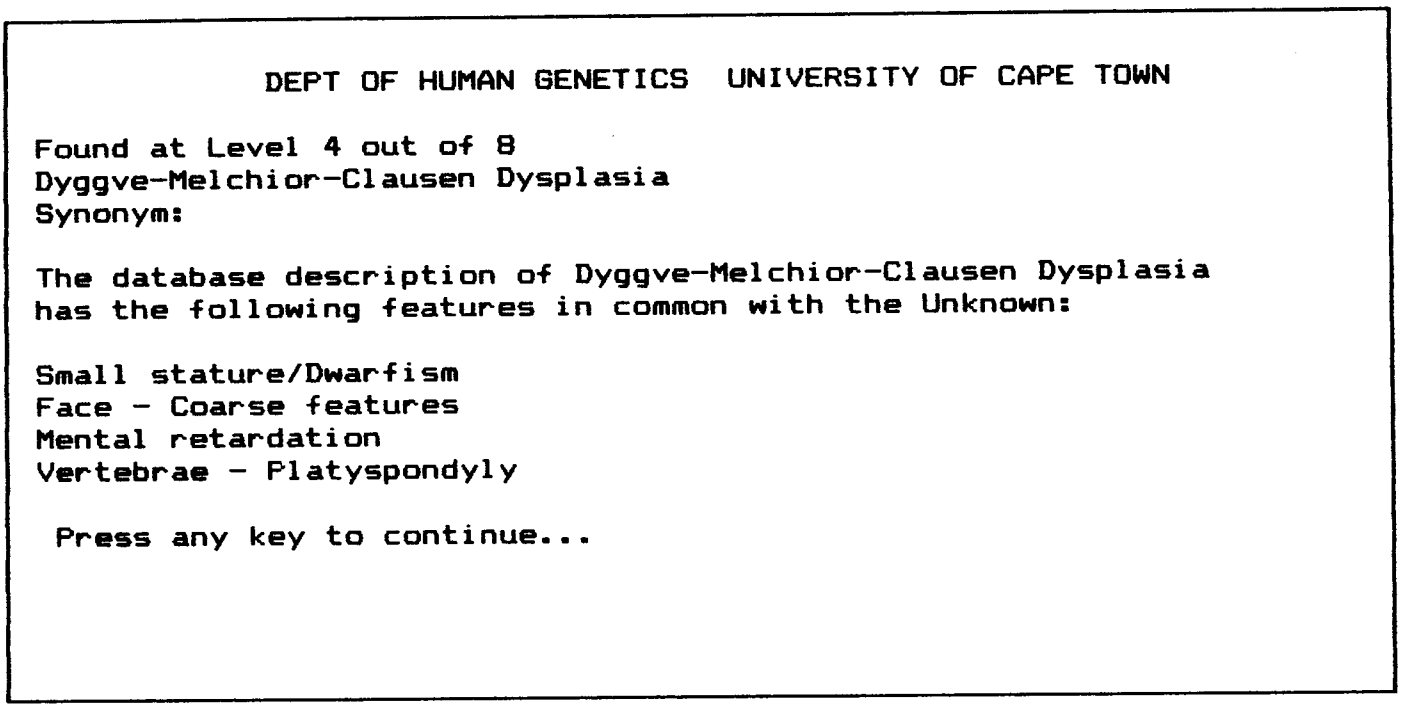


Fig 7-45

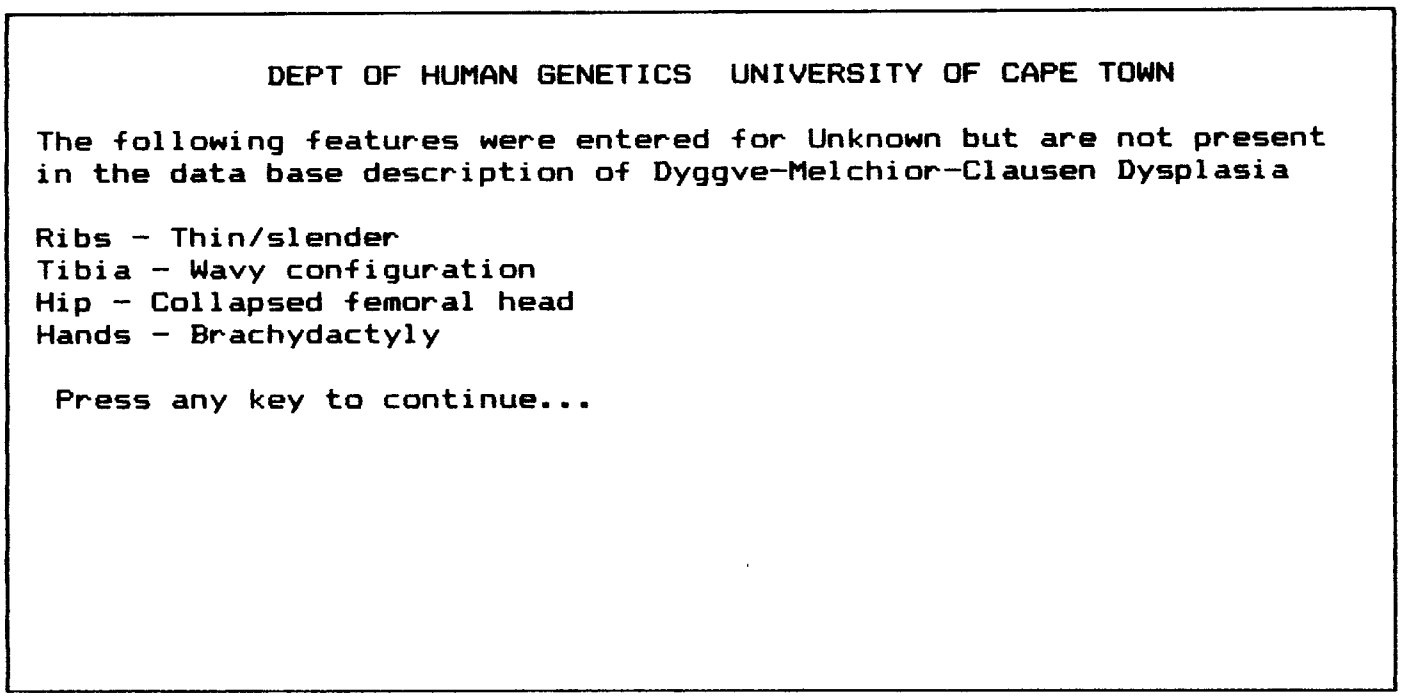


Fig 7-46

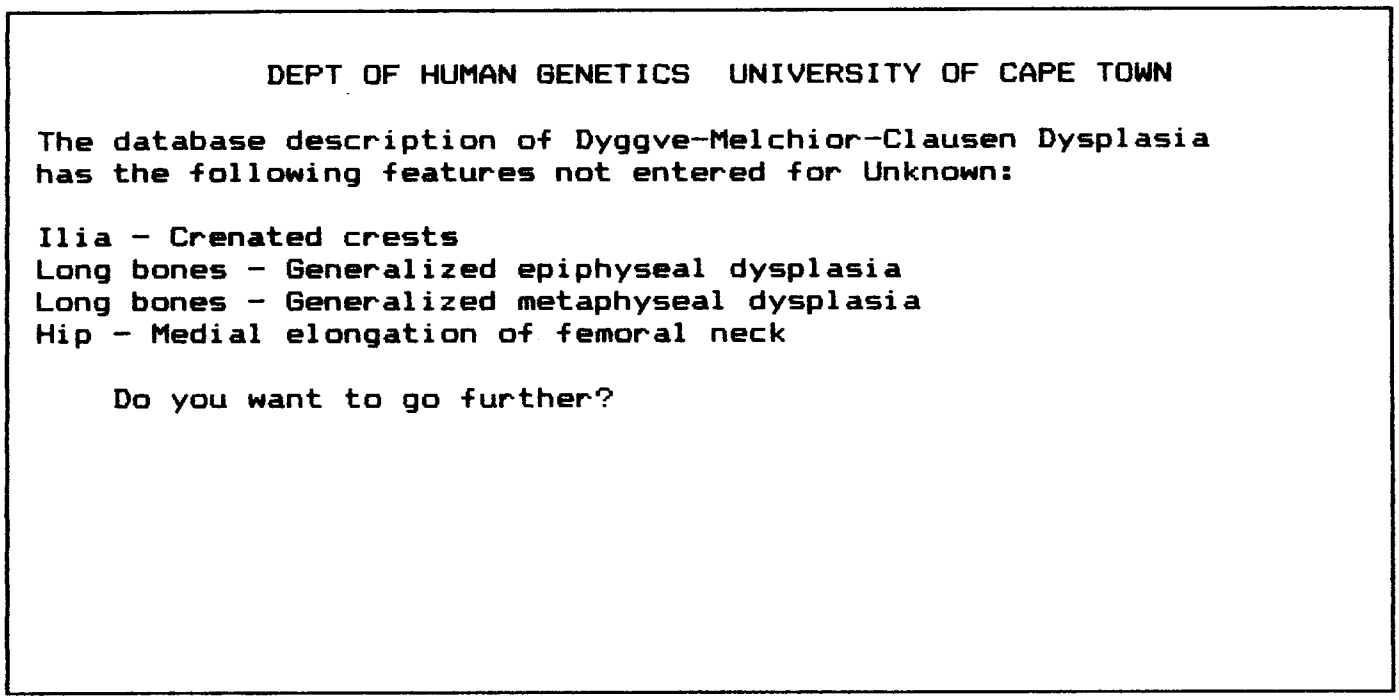


Fig 7-47

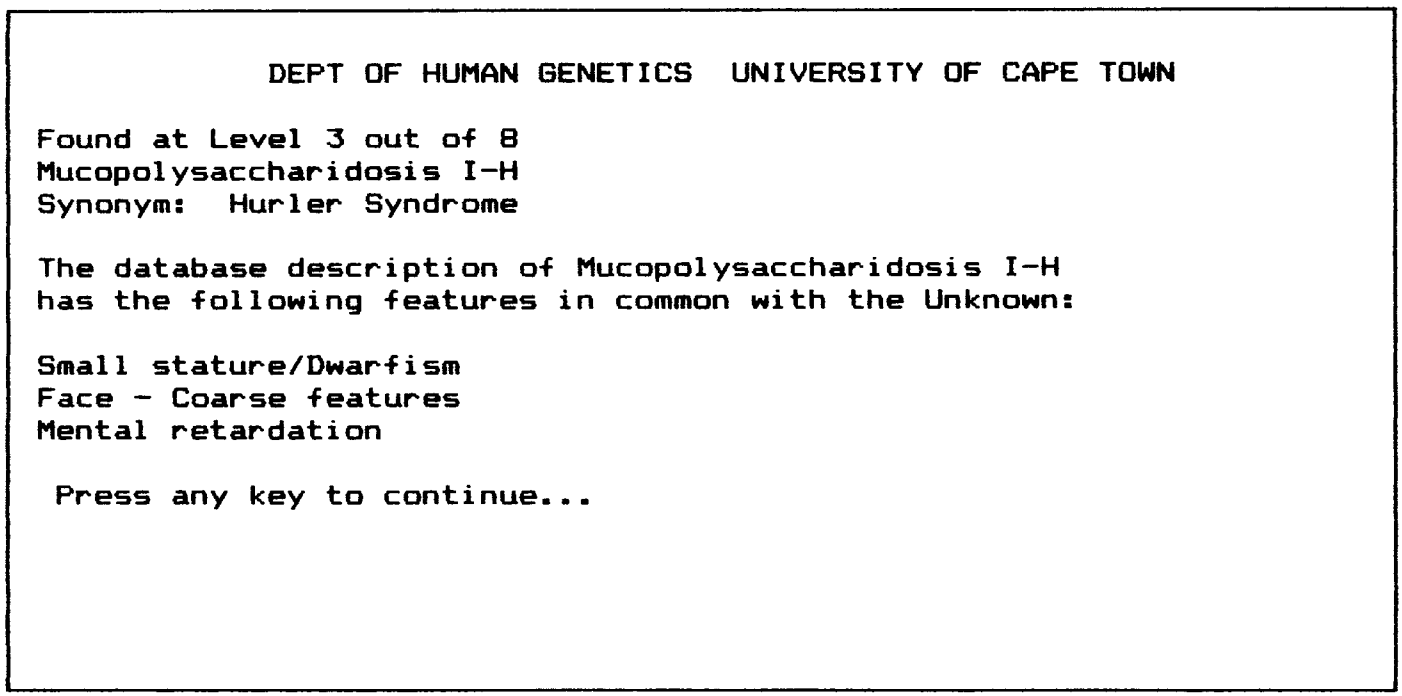


Fig 7-48

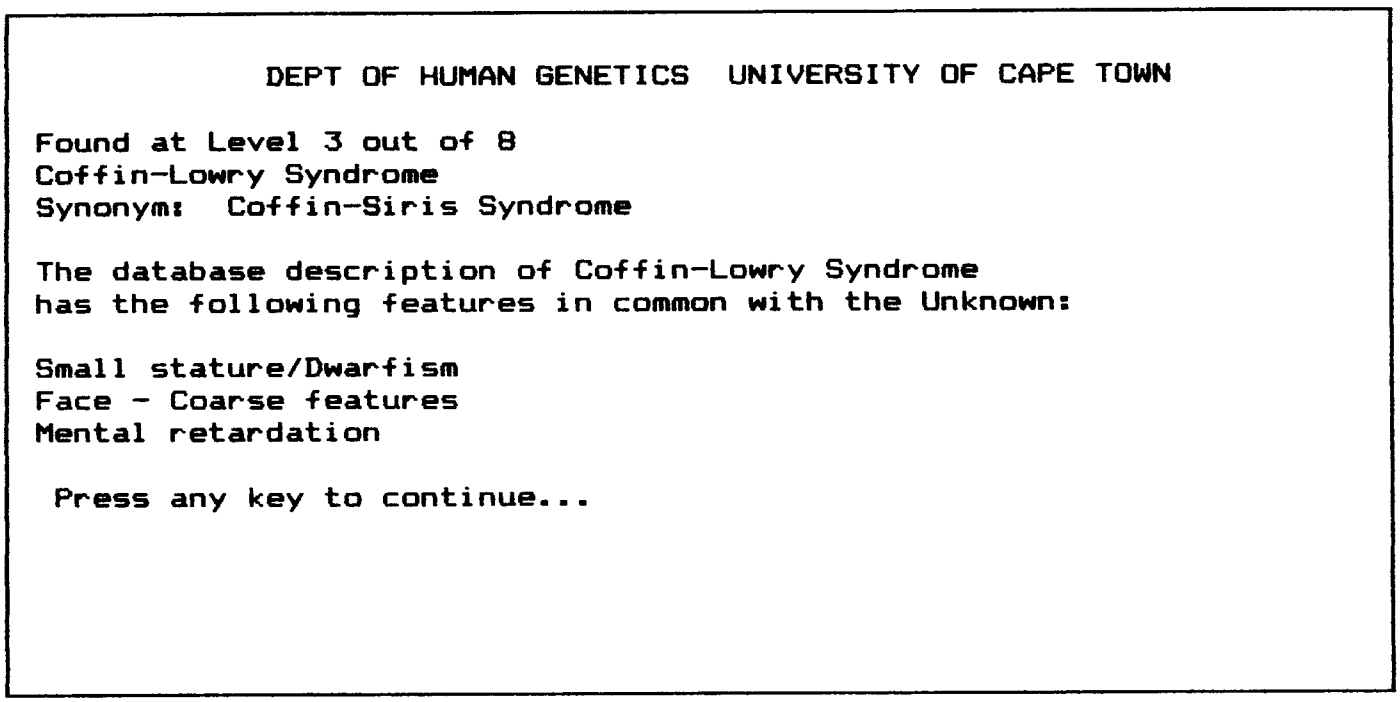


Fig 7-49

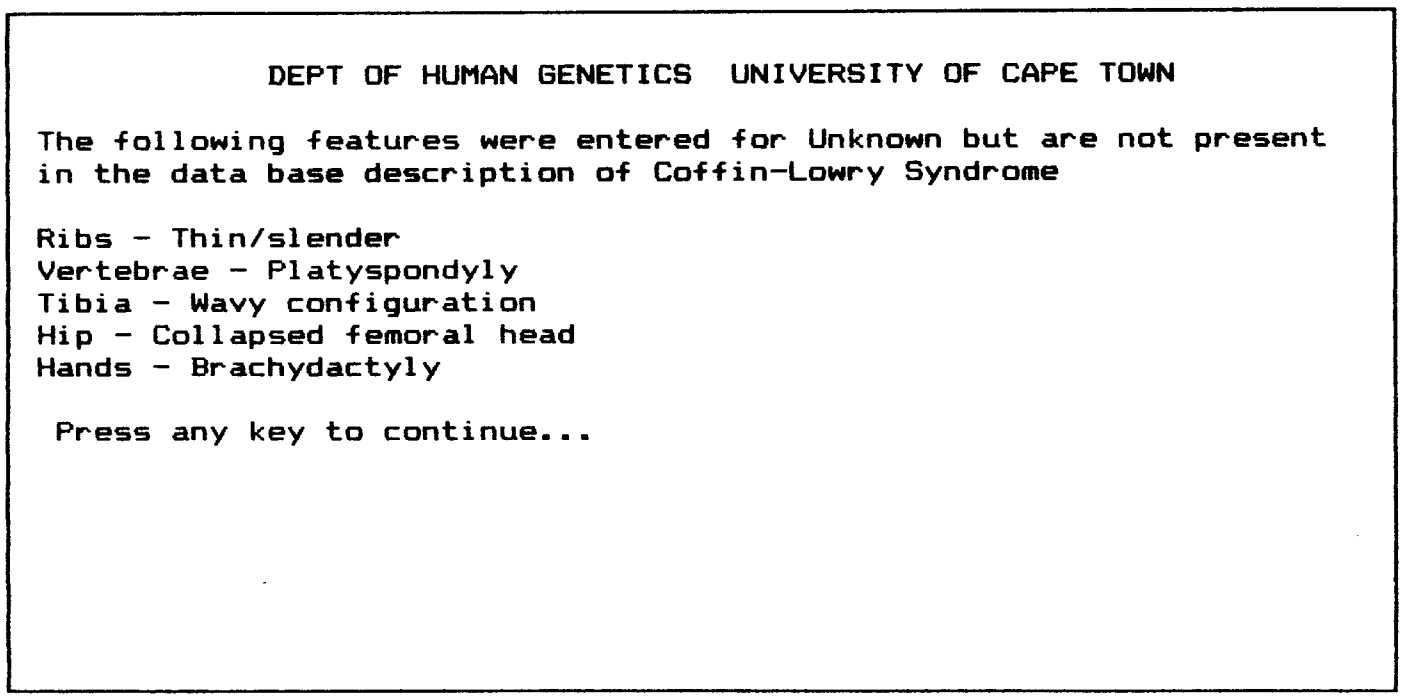


Fig 7-50

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN

The database description of Coffin-Lowry Syndrome
has the following features not entered for Unknown:

Mouth - Macrostomia
Nose - Abnormal shape
Philtrum - abnormal
Abnormal skin/pigmentation
Nail disorder
Soft tissue - Hyperlaxity
Hypertelorism
Calvaria - Hyperostosis
Frontal sinuses - Large
Frontal bones - Prominent/bossed
Pectus carinatum
Spine - Generalized Scoliosis/ Kyphoscoliosis
Knee - Absent/ hypoplastic patella
Hand phalanges - Hyperplastic distal tufts
Retarded bone age - Generalized

Do you want to go further?

Fig 7-51

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN

Found at Level 3 out of 8
Spondyloperipheral Dysplasia
Synonym:

The database description of Spondyloperipheral Dysplasia
has the following features in common with the Unknown:

Small stature/Dwarfism
Vertebrae - Platyspondyly
Hands - Brachydactyly

Press any key to continue...

Fig 7-52

Discussion of case No. 3.

Cretinism was the first disorder to be suggested by the computer on sub-page 7-xxiii. The domain expert (Beighton, 1987) regarded this as very appropriate, and at the time of writing, the matter was being followed up with biochemical investigations. Further suggestions follow on sub-pages 7-xxv to 7-xxviii. Dyggve- Melchior-Clausen syndrome can be ruled out by the absence of the pathognomonic crenated iliac crests and MPS I by the age of the patient. The suggestions of Coffin-Lowry syndrome and Spondyloperipheral dysplasia were considered unlikely but not inappropriate.

7.3.3.2.4 Case No. 4 D.

This 20 year old bilaterally deaf male manifested osteoarthritic degeneration of joints with fixed flexion deformities of the wrists, elbows, hips and knees. Multiple epiphyseal changes were present in the long bones. The vertebrae showed irregular endplates. The case was undiagnosed.

The manifestations were entered as shown in Fig 7-53 at the top of sub-page 7-xxix. The computer responses can be seen on sub- pages 7-xxix to 7-xxxiv. A discussion follows on page 7-9.

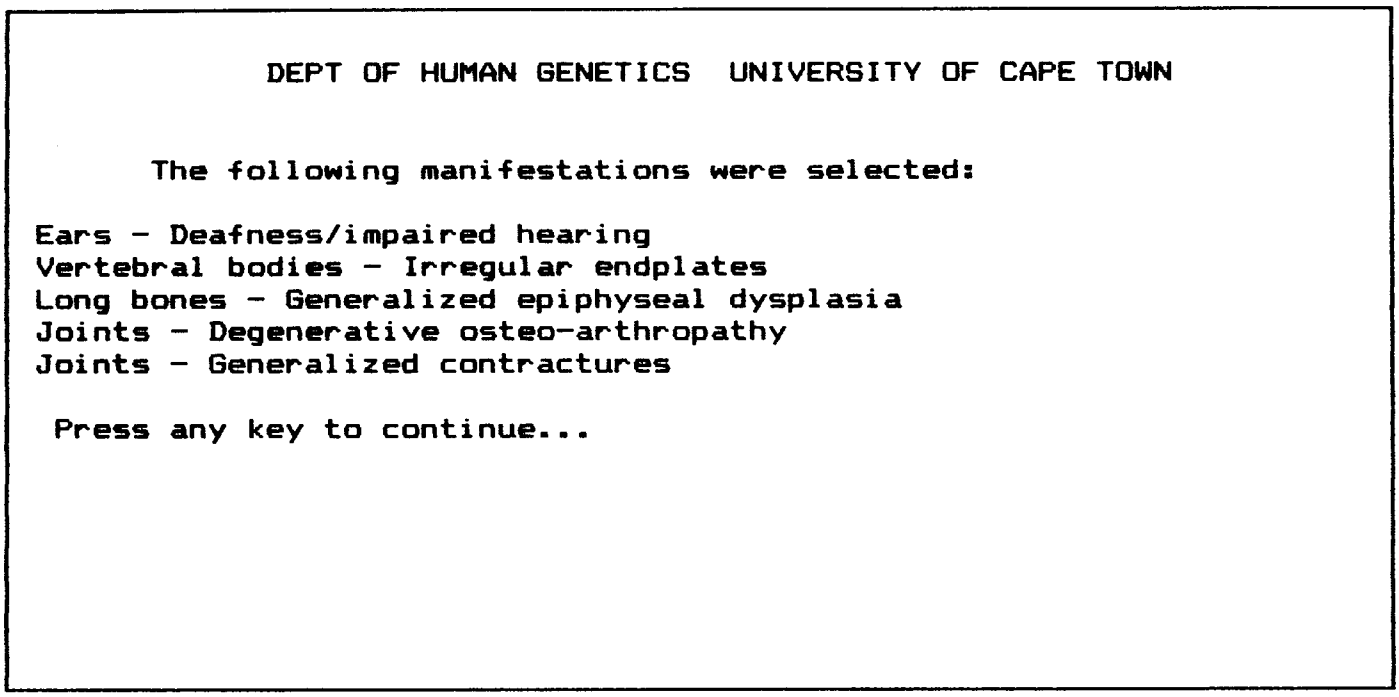


Fig 7-53

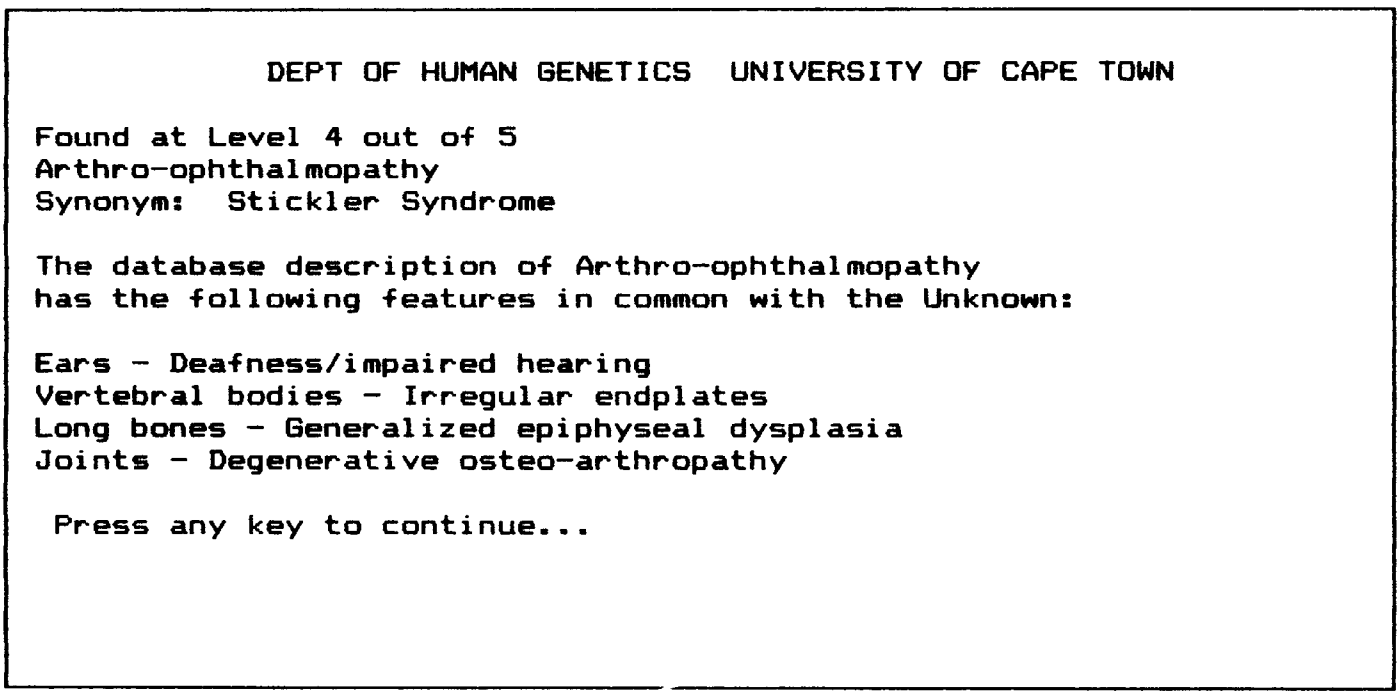


Fig 7-54

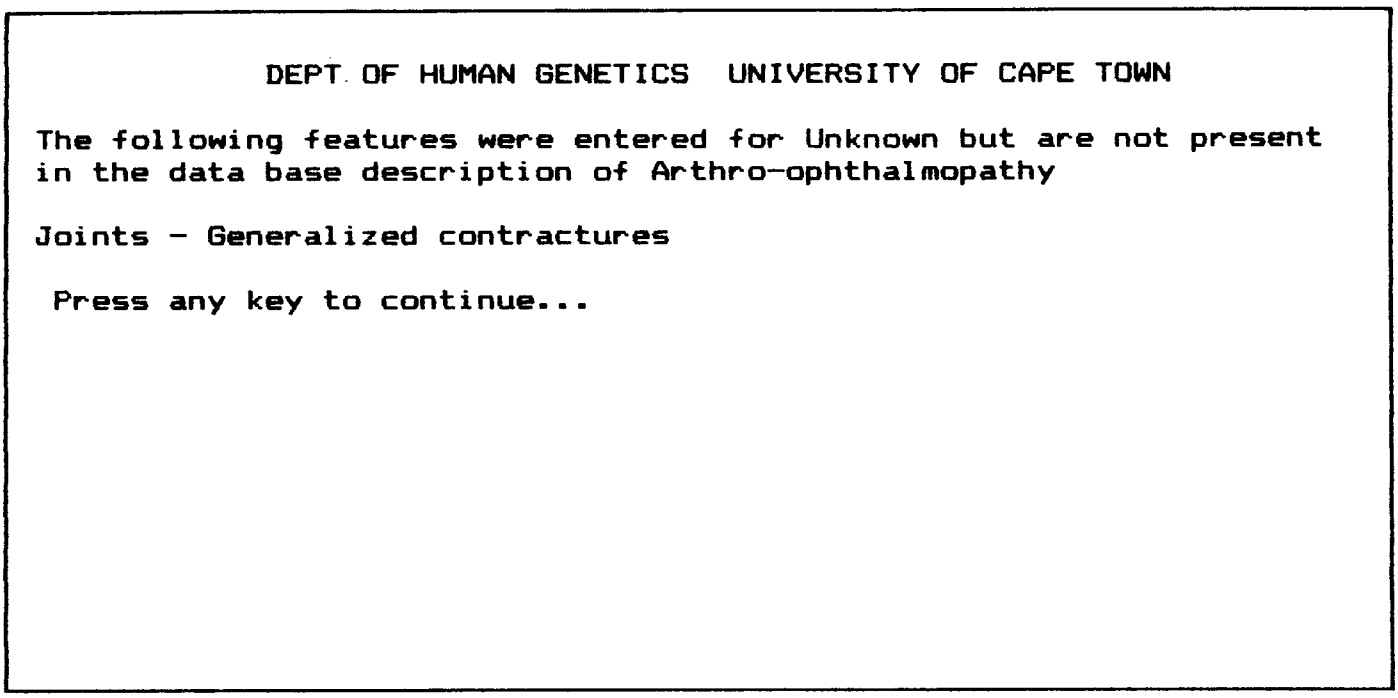


Fig 7-55

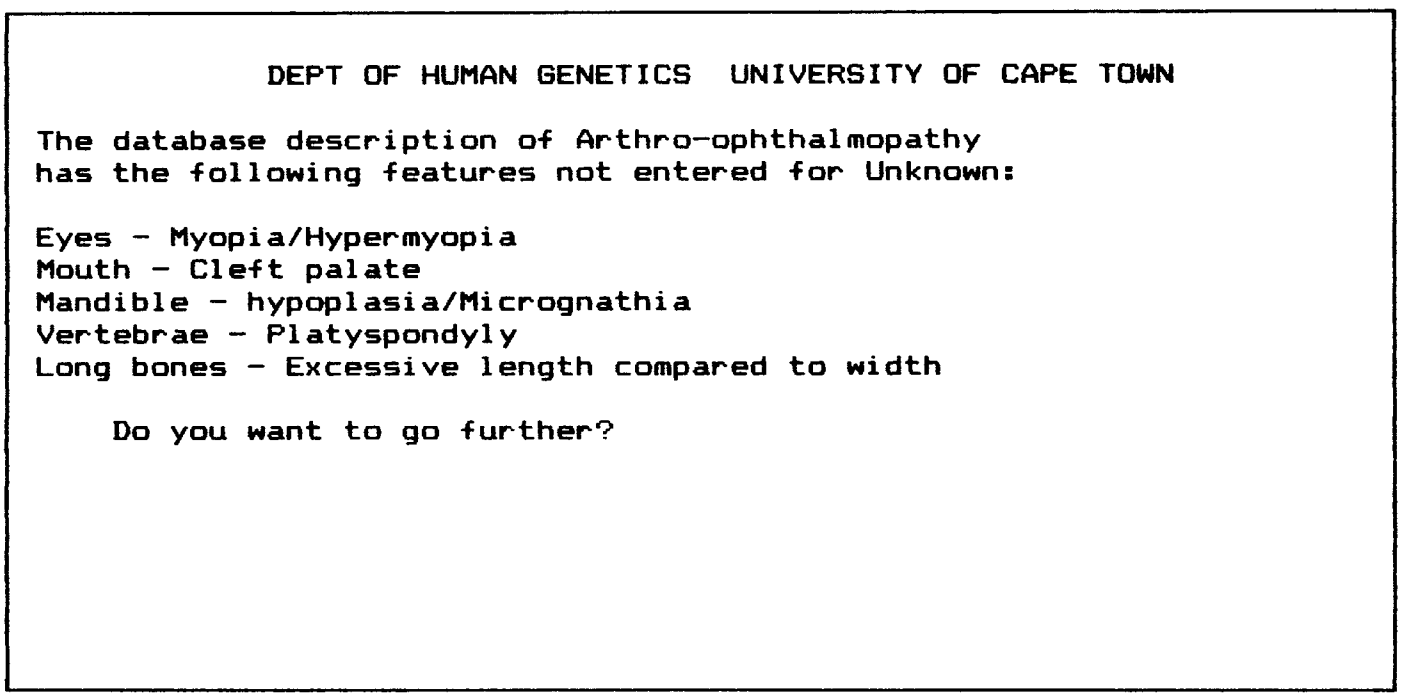


Fig 7-56

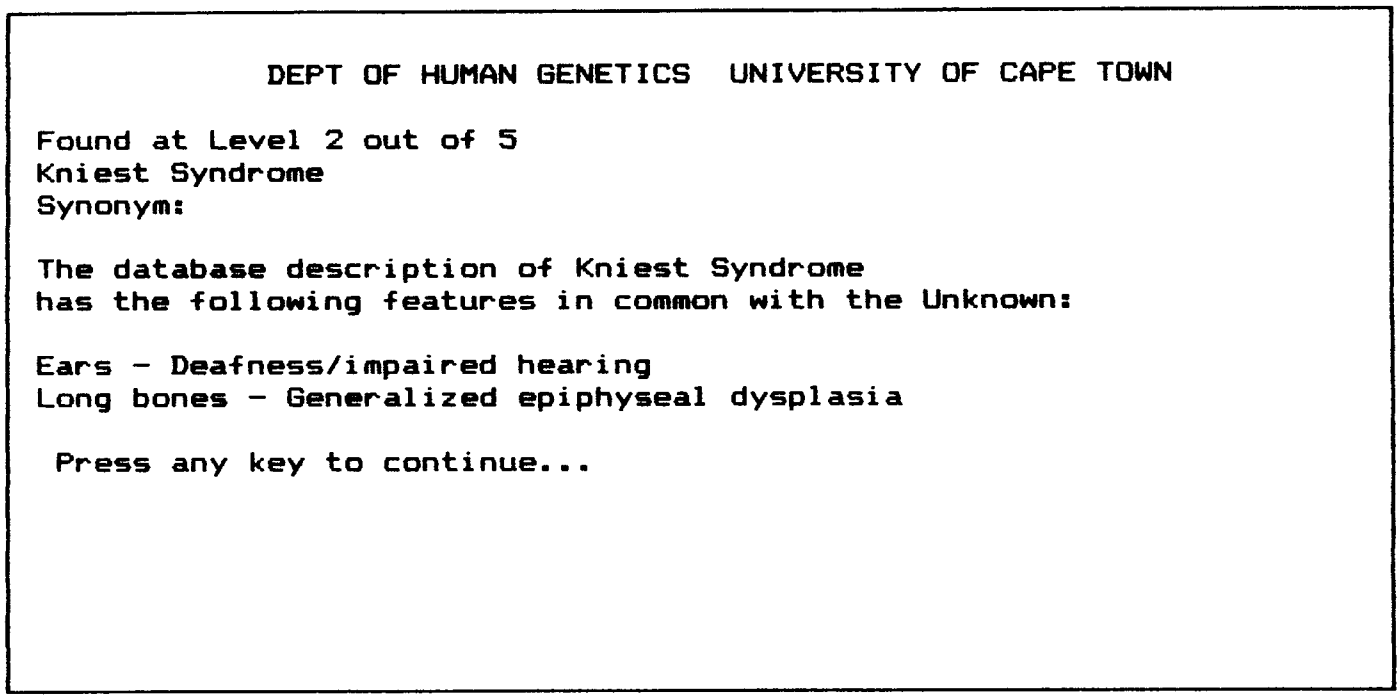


Fig 7-57

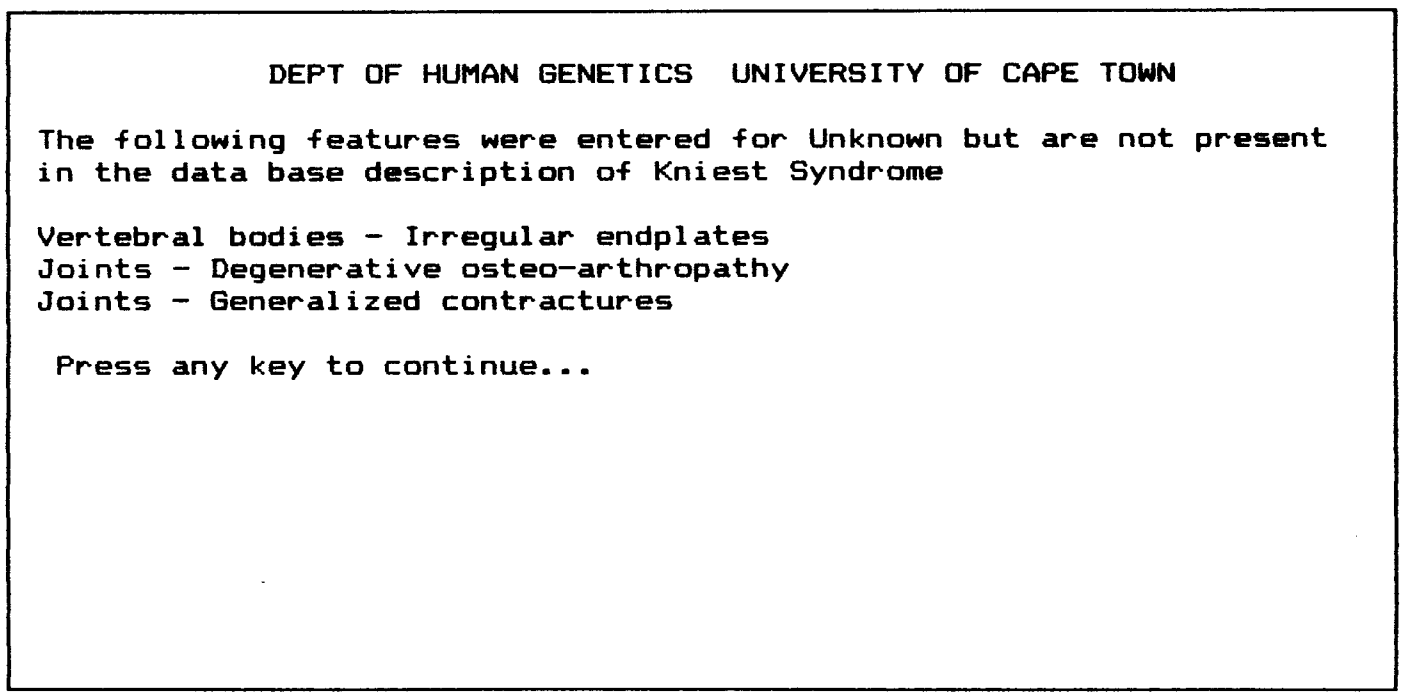


Fig 7-58

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN

The database description of Kniest Syndrome
has the following features not entered for Unknown:

Small stature/Dwarfism
Face - Unusual/Characteristic
Eyes - Non-specific defects
Spine - Generalized Scoliosis/ Kyphoscoliosis
Thoracic spine - Accentuated kyphosis/gibbus
Lumbar spine - Accentuated lordosis
Vertebrae - Platyspondyly
Long bones - Short shafts
Long bones - Expanded metaphyses
Joints - Generalized hypomobility
Hip - Delayed ossification of epiphyses
Hip - Short femoral neck
Hip - Broad/thick femoral neck
Metacarpals - Pseudoepiphyses

Do you want to go further?

Fig 7-59

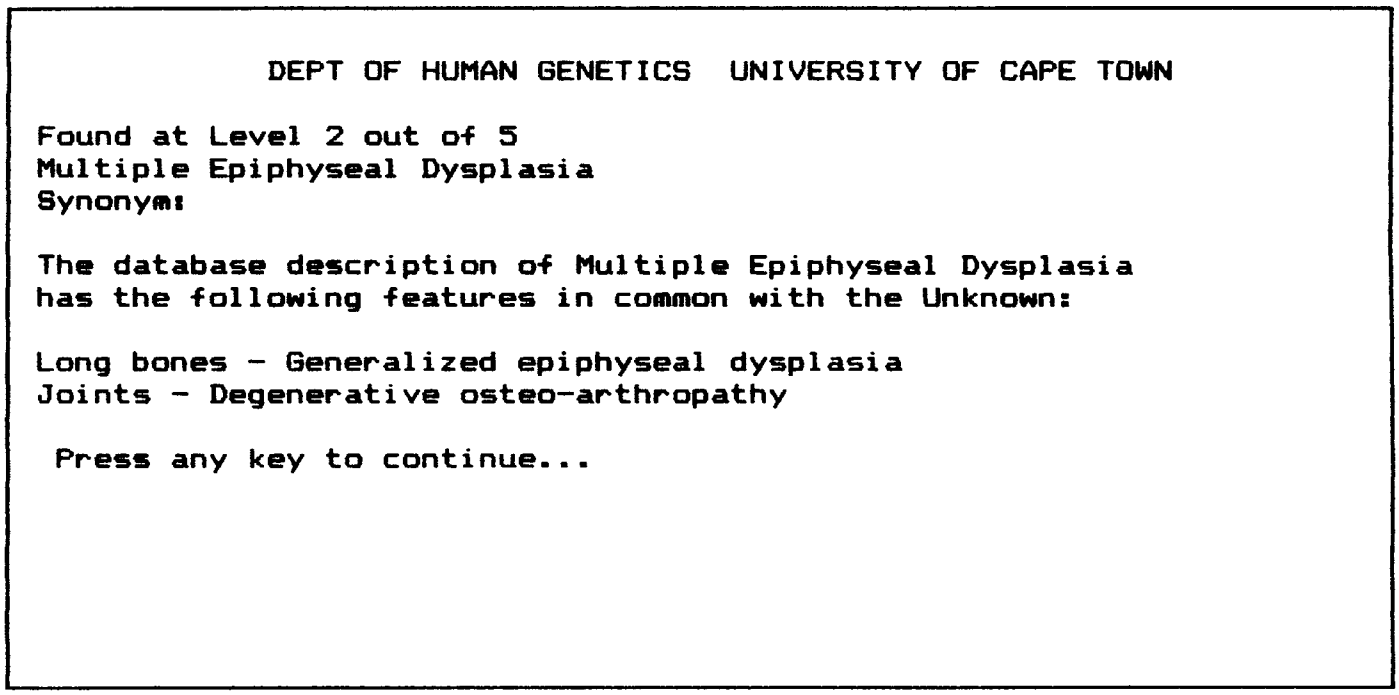


Fig 7-60

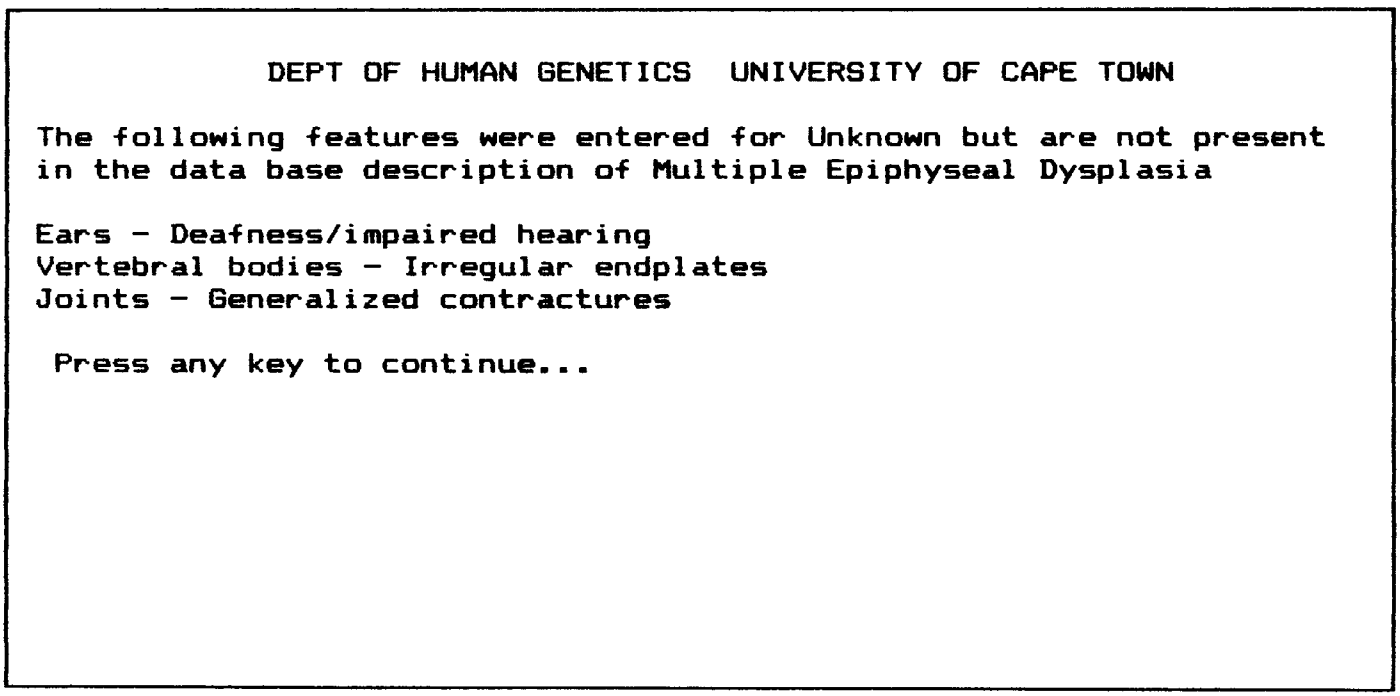


Fig 7-61

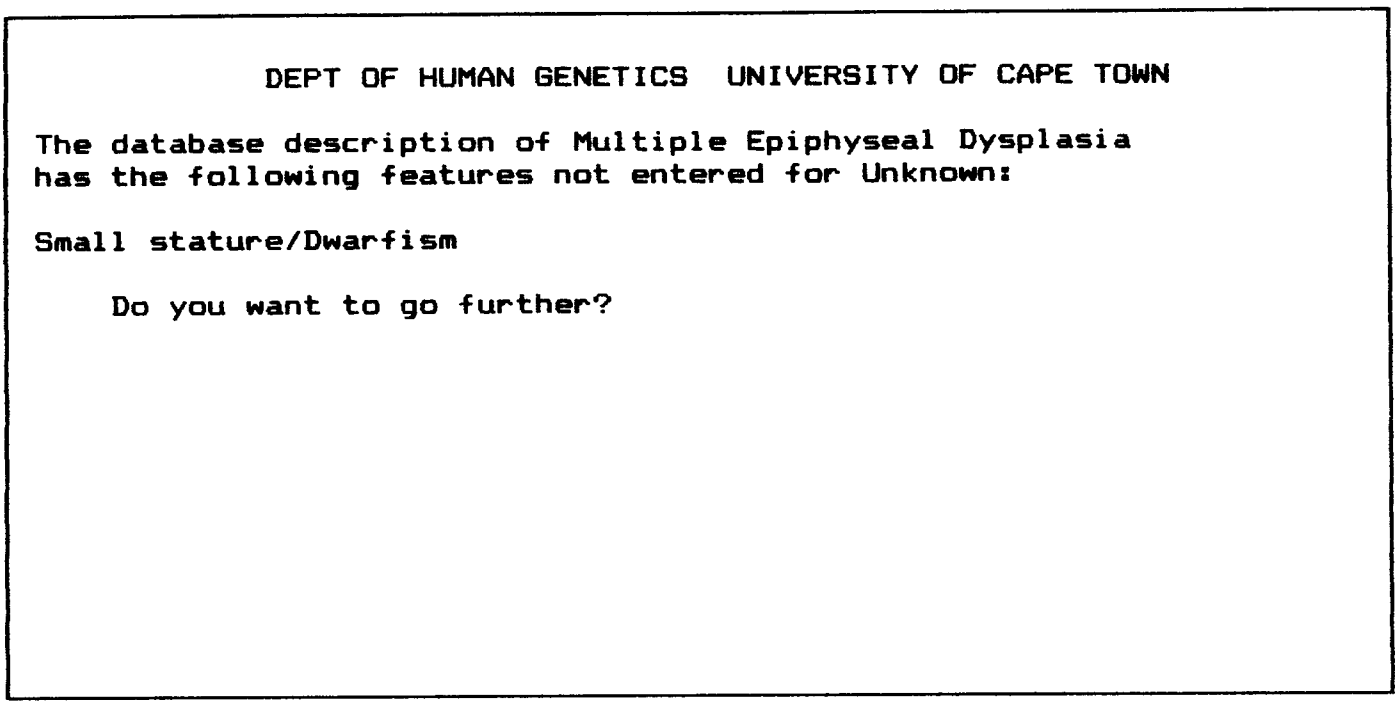

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN
The database description of Multiple Epiphyseal Dysplasia
has the following features not entered for Unknown:
Small stature/Dwarfism
Do you want to go further?

Fig 7-62

Discussion of case No. 4.

The first suggestion from the computer, shown in Fig 7-54 is that of the Stickler syndrome. This diagnosis was, at the time of writing, considered as the most likely in the absence of better alternatives. The other computer suggestions, namely Kniest syndrome and Multiple Epiphyseal Dysplasia (sub-pages 7-xxxi to 7-xxxiv) were thought to be acceptable but less likely. The possibility also exists that this patient has an undelineated syndrome.

7.3.3.3 Value as a Diagnostic Aid.

With case no. 1 the value of the computer as a diagnostic aid lay in the ability to suggest conditions that might not normally be considered. The suggestion of Generalized Enchondromatosis after Enchondromatosis may come as a surprise to users who are unaware of the distinction between the two disorders. When profile material was collected from the literature for Ollier disease, Generaized Enchondromatosis was encountered as well as the two other disorders with spinal involvement, namely Spondyloenchondrodysplasia and Enchondromatosis with irregular vertebral lesions (Spraner et al, 1978). These conditions were promptly entered into the knowledge base at the time. During recent discussions with domain expert Spranger (1987), the existence of these separate entities was confirmed.

The responses to case no. 2 tended to confirm the clinical diagnosis that had already been made.

The suggestion of Cretinism for case no. 3 was significant. Domain expert Beighton had entertained this possibility, but less expert users would probably not consider this diagnosis in the initial stages. For geographic reasons the diagnosis had not been confirmed or otherwise at the time of writing, but was considered to be a strong possibility. Brachydactyly, platyspondyly and tibial changes are not usually encountered in cretinism, and the possibility exists that this patient had some other undelineated entity in addition.

Case no. 4 had been undiagnosed before the computer suggestion of Stickler syndrome. This diagnosis was subsequently considered to be the most likely.

7.3.3.4 Conclusion.

The responses to the combinations of manifestations as well as to the five problem cases, indicate that the microcomputer system may well be a useful aid in the differential diagnosis of skeletal disorders.

SECTION EIGHT.

PROTOTYPE REVISION - A FUTURE PROJECTION

CHAPTER 8.1
INTRODUCTION.

In section 7 it was shown that the computer application which formed the subject of this thesis proved to be a successful and valuable aid in the diagnosis of the inherited diseases of the skeleton. In this section some possible developments for the future will be outlined and the maintenance features discussed.

CHAPTER 8.2
MAINTENANCE FEATURES.

Maintenance of this application will of necessity have to be twofold since both computer and medical aspects are involved.

8.2.1 The Medical Aspects.

New syndromes are described from time to time. To be able to enter such new profiles promptly into the knowledge base, will necessitate a close scrutiny of the professional literature on a regular basis. In chapter 3.3 it was shown that the ease with which a computer database can be updated as opposed to a book version, constitutes a major advantage. If an efficient updating routine can be maintained, the knowledge base developed in this project could prove to be a valuable repository for newly delineated genetic skeletal disorders.

"Older" syndromes are sometimes modified by the addition of new manifestations or the removal of stigmata that might have been erroneously ascribed. Sometimes it is ascertained that a particular syndrome that had been regarded as autonomous, is in fact a variation of some other condition. Such factors will also demand continual monitoring of the relevant journals in order to keep the knowledge base current.

8.2.2 The Computer Aspects.

Rapid developments are taking place in the microcomputer field. Better and more extensive devices for secondary storage are becoming available. The microprocessors themselves are becoming more sophisticated with ever more rapid execution times, and it would seem that microcomputers soon will have capabilities comparable to those of smaller mainframe equipment. Since operating systems as well as programming languages are updated rather frequently, it may very well become necessary to rewrite and adapt applications (such as the one under discussion in this thesis) in order for them to remain useful in a changing data processing environment.

Ever since its conception, this project has been intended for the "single user." The nature of the domain is such that it would probably not find application in a multi-user environment. Should the ideas developed in this project be extended to larger disciplines, however, this viewpoint

may have to be reviewed. A multi-user environment such as typically provided by the Unix operating system, may then be considered more suitable. This would probably necessitate the rewriting of the programs in a language more appropriate to the particular operating system. Should, for instance, Unix be decided upon, the C programming language, which is closely associated with it, probably will have to be employed for the sake of maximum efficiency. Such a conversion will present a formidable task, although no serious practical difficulties should be encountered. The C language, for instance, lends itself admirably to bit manipulation (Schildt, 1985). This capability may even obviate the need for routines written in assembly language.

8.2.3 User support.

It is very likely that the application developed with this thesis will be used by medical personnel rather than by computer staff. In order to make the programs as "user friendly" as possible, as much support as possible must be provided. At present utilisation is straightforward being "menu-driven", and it would appear that users should be able to operate by means of the text provided in the various programs. Additional help facilities in the form of text files which could be consulted at strategic places during execution of the programs, should enhance user friendliness. Some users will probably consult the knowledge base on an occasional basis only. Such infrequent exposure is known to hamper the development of familiarity with any application, in which case the operation may have to be relearned every time the computer is employed. Help files are very effective in lessening or even obviating the necessity of such time-consuming retraining. Auxiliary files for this project have not been prepared as yet, but are being planned for the future. A user manual as well as a proper data dictionary as defined for larger databases (Cardenas, 1984) are also envisaged.

CHAPTER 8.3
POSSIBLE FUTURE DEVELOPMENTS.

8.3.1 Interface with the Gamut Index.

The Microcomputer Gamut Index was referred to in chapter 3.4. At present the gamut program presents consecutive menus from which applicable options may be selected. It ends with a list of likely conditions which should be considered in the diagnosis of a problem disease under consideration.

In the book version the various disease profile descriptions which constitute the final differential diagnostic list, may be looked up individually. This facility has not yet been developed in the computer application of the gamut index. With the development of a successful profile storage strategy as well as effective search and presentation routines for the diagnostic aid package, interface with the MGI becomes feasible. This is planned as a future undertaking, together with the incorporation of section 2 of the gamut index into the computer version.

8.3.2 Computer Graphics Application.

The present application has no graphics capabilities. Since the field of computer graphics is developing rapidly, it may very well become possible to develop diagrammatic presentations to facilitate the diagnosis of skeletal dysplasia. With the present state of the art, high resolution graphics are very demanding on computer memory and are not yet considered feasible for a microcomputer application such as this. The diagrams resulting from the use of lower resolution programs are not considered satisfactory, especially where small bones such as the carpals and tarsals are concerned.

8.3.3 Video Disk Applications.

The prospect of retrieving by computer, photographs and X-ray pictures stored on video disk is exciting. Such storage is already being used by the BDIS application in the USA. Economic factors precluded the application of this development to the present project. As a future prospect, it should rate high on the list of priorities.

8.3.4 Relative Importance of Manifestations (Weighting).

It was shown in chapter 2.3 that genetic manifestations may often be represented in a binary fashion, ie. a particular feature is either present or absent. Not infrequently, however, stigmata may vary in "intensity" from one disease to another, and the degree of severity becomes an important distinguishing factor. In addition, the diagnostic significance of different stigmata in any specific entity has very different weighting. It should therefore be useful to be able to assign weight factors to the various features that make up a disease profile. The necessity for this may become more acute with applications involving broader disciplines, or those where descriptions tend to be less clearcut.

The present project does not provide for weighting, but the following approach has been conceived by the author and could be implemented as a future project:

8.3.4.1 The Orthogonal Byte Concept.

Further bytes could be computed for the profile fields of the disease records. Pairs of bits in these extra bytes could be used to qualify the bits in the tier2 bytes which are set to indicate the presence of manifestations. The following scheme could be followed:

Binary weight	Decimal weight	Meaning
11	3	Cardinal feature, presence imperative
10	2	Feature often present
01	1	Feature sometimes present
00	0	Imperative absence.

By imperative absence is meant that the stigmata must be absent from the profile

in order to distinguish the particular disorder from other similar afflictions which, in turn manifest the features under consideration. In this context, the absent features which serve to distinguish between Pseudoachondroplasia and Achondroplasia (characteristic facies, for instance), may be cited as examples. However, with this approach, the imperatively absent manifestations will have to be denoted as "present" (with set bits) at tier2 level, but with qualifying bits of zero value. This will require careful adaptation of the application programs.

It was decided to designate the new type of bytes which house the qualifying bits as "orthogonal", since, conceptually the bits may be regarded as being "at right angles" to the feature bits which they qualify. This may be represented as follows:

```
0 1 0 1 0 0 0 1 0 0 0 1
  0   1       1       0
  1   0       1       0

  1   2       3       0
```

sometimes present *often present* *cardinal feature* *imperative absence*

In a recent publication Frey (1986) describes a different approach. He uses a model from architecture which could be considered for application in the medical field.

With Frey's approach uncompressed bit maps are made up in a binary fashion to denote the presence or absence of features pertaining to the subject under discussion. Different strings are, however, computed for features that must be present (category A), for those that are usually present (B) and for those only sometimes present (C). If the coding strategies developed with this thesis should be applied to Frey's approach (disregarding the absence of compression for the moment), at least three profile strings will have to be computed for each disease entity. Should it be important to denote imperative absence, a further string will be required. With the orthogonal byte method on the other hand, the extra information comprises only a few extra bytes which could be appended to the profile string.

Encoding and decoding of such a slightly longer string should prove to be more rapid and economical than the processing of three or four discrete strings.

The orthogonal byte that will contain the qualifying bits shown in the illustration, may be represented as follows:

```
01 10 11 00
 1  2  3  0
```

The orthogonal byte idea and its application to the weighting problem will be an important aspect of prototype revision at a near future date.

CHAPTER 8.4
CONCLUDING REMARKS.

A successful and useful microcomputer diagnostic aid facility has been developed with this study and the following points have emerged:

8.4.1 Microcomputer Suitability.

It can be concluded that a microcomputer of the IBM XT or AT type can accommodate an extensive knowledge base with the coding strategies developed. The following points are particularly relevant:

- i) Enlargement of the knowledge base can be accomplished very economically.

Each of the disease profiles can be expanded considerably, even up to several hundred percent, without requiring any extra disk space. This is due to the fact that the fields for the bit maps are of constant length. Expansion of the disease descriptions is accomplished "at the expense of" bits set to zero, rather than new computer space. Should new manifestations be added to a profile, more set bits are stored in newly computed bytes, but the latter will never exceed the modest number of 50. This is a serendipitous finding and probably the most significant of the entire project, as it denotes the feasibility of extending the compressed bitmapping approach to larger fields.

Even adding new manifestations to the features file is possible without further demands on computer storage resources. At the time of writing, the data base contained 715 manifestations, hence there is room for 309 more. Should more than 1024 manifestations be required, a relatively insignificant adaptation in design, including the addition of one or two tier0 bytes to the basic structure, will allow for an additional 512 or 1024 features respectively.

The only demand on more storage space will be from the addition of new disease profiles. However, this will occur at the modest rate of 417 bytes per profile. Of these, 260 bytes are at present taken up by "luxury" features such as notes and relatively extensive genetic descriptions which could, if necessary, be curtailed. The disease profile can comfortably be stored in 157 bytes. Over 2 000 profiles will therefore fit on a flexible disk of the present format. A 20 Mb hard disk, a size modest by today's standards, will accommodate over 120 000 disease profiles. These can be expanded significantly, without further demands on disk space, by switching bits on where needed. There will also be room for all the application programs and the table of manifestations.

- ii) The software is suitable.

The application which forms the subject of this thesis has been developed with three readily available and affordable software packages, namely Wordstar, Turbo-Pascal and the Macro-assembler for the 8088/8086 microprocessor. Execution times of all the programs are remarkably short. It is felt that considerably more extensive applications could be designed by means of these software "tools".

- iii) Improvement of the knowledge base.

The knowledge base improves with time as the profile descriptions are refined by the addition or deletion of relevant manifestations. The present knowledge base can be regarded as a kernel which could be adapted and expanded by ongoing perusal of the resources and interaction with the domain experts. This could be seen as a "learning process" without concomitant physical expansion into computer memory resources. This fact may well compensate for the absence of an inference engine as defined for an expert system and its accompanying severe physical demands on the computer.

The quality of the features file is cardinal to the success of an application such as the present one. In chapter 4.2 it was shown that a threat to the integrity of the knowledge base exists. Altering the meaning of a manifestation significantly can have unpleasant effects, hence the password control. This should be regarded as a weakness in the system and a subject for future improvement. An aim should be to be able to shift manifestations around the file at will, should this result in greater efficiency. This is not possible at present and for the time being one can do no better than exercise care. Optimizing the manipulation of the features file will also be given high priority lest the selection of manifestations from a relatively long list should become cumbersome.

SECTION NINE.

BIBLIOGRAPHY

Anderson IF. (1965)
Maffucci's syndrome: report of a case with a review of the literature.
S Afr Med J ; 39 : 1066-1070 .

Anghern V, Zachmann M, Prader A. (1979)
Silver-Russel syndrome. Observations in 20 patients.
Helv Paediatr Acta ; 34 : 297-308 .

Bathfield CA, Beighton PH. (1978)
Blount disease: a review of aetiological factors in 110 patients.
Clin Orthop ; 135 : 29-33 .

Beighton P, Hamersma H, Horan F. (1979)
Craniometaphyseal dysplasia - variability of expression within a large family.
Clin Genet ; 15 : 252-258 .

Beighton P, Hamersma H, Raad M. (1979)
Oculodento-osseous dysplasia: heterogeneity or variable expression?
Clin Genet ; 16 : 169-177 .

Beighton P, Hamersma H. (1980)
Frontometaphyseal dysplasia: autosomal dominant or X-linked?
Med Genet ; 17 : 53-56 .

Beighton P, Horan FT. (1981)
Spondylocostal dysostosis in South African sisters.
Clin Genet ; 19 : 23-25 .

Beighton P. (1974)
Autosomal recessive inheritance in the mesomelic dwarfism of Campailla and Martinelli.
Clin Genet ; 5 : 363-367 .

Beighton P. (1981)
Familial dentinogenesis imperfecta, blue sclerae, and wormian bones without fractures: Another type of osteogenesis imperfecta?
J Med Genet ; 18 : 124-128 .

Beligere N, Harris V, Pruzansky S. (1981)
Progressive bony dysplasia in Apert syndrome.
Radiology ; 139 : 593-597 .

Bennett WM, Musgrave JE, Campbell RA, Elliot D, Cox R, Brooks RE et al. (1973)
The nephropathy of the nail-patella syndrome: clinico-pathologic analysis of 11 kindred.
Am J Med ; 54 : 304-310 .

Berry C, Cree J, Mann T. (1980)
Aarskog's syndrome
Arch Dis Child ; 55 : 706-710 .

Brachman RJ, Amarel S, Engelman C, Engelmore RS, Feigenbaum EA, Wilkins DE. (1983)
What Are Expert Systems?
In: Hayes-Roth F, Waterman DA, Lenat DB eds. Buiding Expert Systems. London: Addison-Wesley: 31-57.

Brill PW, Beratis NG, Kouseff G, Hirschhorn K. (1975)
Roentgenographic findings in fucosidosis type 2.
Am J Roentgenol ; 124 : 75-82 .

Brown DM, Bradford DS, Gorlin RJ et al. (1976)
The acro-osteolysis syndrome: Morphologic and biochemical studies.
J Pediatr ; 88 : 573-580 .

Buchanan BG, Barstow D, Bechtal R et al. (1983)
Constructing an Expert System.
In: Hayes-Roth F, Waterman DA, Lenat DB eds Building Expert Systems. London: Addison-Wesley,: 127-167.

Campbell CJ, Papademetriou T, Bonfiglio M. (1968)
Melorheostosis. A report of the clinical, roentgenographic and pathological findings in fourteen cases.
J Bone Joint Surg [Am] ; 50 : 1281-1304 .

Cardenas AF. (1984)
Data Base Management Systems.
2nd Ed Boston: Allyn and Bacon Inc.: 593-601 .

Charniak E, McDermott D. (1985)
Introduction to Artificial Intelligence.
1st ed: Reading Ma: Addison-Wesley Publishing Company.: 319-320 .

Cheney WD. (1965)
Acro-osteolysis.
Am J Roentgenol ; 94 : 595-607 .

Connor JM, Horan FT, Beighton P. (1983)
Dysplasia epiphysealis hemimelica.
J Bone Jt Surg [Br] ; 65 : 350-354 .

Cremin B, Connor JM, Beighton P. (1982)
The radiological spectrum of fibrodysplasia ossificans progressiva.
Clin Radiol ; 33 : 499-508 .

Da Silva EO. (1982)
Autosomal recessive Klippel-Feil syndrome.
J Med Genet ; 19 : 130-134 .

Danks DM, Stevens BJ, Campbell PE et al. (1972)
Menkes kinky hair syndrome.
Lancet ; 1 : 1100-1102 .

De Dombal FT, Leaper DJ, Horrocks JC et al. (1974)
Human and computer-aided diagnosis of abdominal pain: Further report with emphasis on performance of clinicians.
Brit Med J ; 1 : 376-380 .

De Dombal FT, Leaper DJ, Staniland JR, McCann AP, Horrocks Jane C. (1972)
Computer-aided Diagnosis of Acute Abdominal Pain
Brit Med J ; 2 : 9-13 .

Duncan PA, Shapiro LR, Stangel JJ, Klein RM, Addonizio JC. (1979)
The MURCS association: Muellerian duct aplasia, renal aplasia, and cervicothoracic somite dysplasia.
J Pediatr ; 95 : 399-402 .

Eastman JR, Escobar V. (1978)
Femoral hypoplasia-unusual facies syndrome, a genetic syndrome?
Clin Genet ; 13 : 72-76 .

Edwards CN. (1985)
The New Micro BDIS Subscriber's Guide.
1st ed: Dover Ma: Center for Birth Defects Information Services, Inc.: 4-29.

Epstein DA, Levin EJ. (1978)
Bone scintigraphy in hereditary multiple exostoses.
Am J Roentgenol ; 130 : 331-333 .

Escobar V, Bixler D, Gleiser S, Weaver DD, Gibbs T. (1978)
Multiple pterygium syndrome.
Am J Dis Child ; 132 : 609-611 .

Escobar V, Gleiser S, Weaver DD. (1978)
Phenotypic and genetic analysis of the Silver-Russel syndrome
Clin Genet ; 13 : 278-288 .

Espiritu, Chen H, Wooley PV. (1975)
Mesomelic Dwarfism as the Homozygous Expression of Dyschondrosteosis.
Am J Dis Child ; 129 : 375-380 .

Everberg G, Ratjen E, Sorensen H. (1966)
Klippel-Feil's syndrome associated with deafness and retraction of the eyeball.
Br J Radiol ; 36 : 562-567 .

Fanconi S, Issler Chr., Giedion A, Prader A. (1983)
The SPONASTRIME dysplasia: familial short-limb dwarfism with saddle nose, spinal alterations and metaphyseal striation.
Helv. paediat. Acta ; 38 : 267-280 .

Felman AH, Frias JL. (1973)
Trichorhinophalangeal syndrome: study of 16 patients in one family.
Am J Roentgenol ; 129 : 631-637 .

Filippi G. (1972)
The Rubinstein-Taybi syndrome. Report of 7 cases.
Clin Genet ; 3 : 303-318 .

Frey PW. (1986)
A Bit-mapped Classifier.
Byte ; 11(12) : 161-172 .

Furukawa CT, Hall BD, Smith DW. (1972)
The Aarskog syndrome.
J Pediatr ; 81 : 1117-1122 .

Gay BB, Kuhn JP. (1976)
A syndrome of widened medullary cavities of bone, aortic calcification, abnormal dentition and muscular weakness (the Singleton-Merten syndrome).
Radiology ; 118 : 389-395 .

Gehweiler JA, Bland WR, Carden TS, Daffner RH. (1973)
Osteopathia striata - Voorhoeve's disease: review of the roentgen manifestations.
Am J Roentgenol ; 118 : 450-455 .

Gilbert EF, Opitz JM, Spranger JW, Langer LO, Wolfson JJ, Visekul C. (1976)
Chondrodysplasia punctata, rhizomelic form.
Eur J Pediatr ; 123(2) : 89-96 .

Glanz A, Fraser FC. (1982)
Spectrum of anomalies in Fanconi anaemia.
J Med Gen ; 19 : 412-416 .

Goltz RW, Peterson WC, Gorlin RJ, Ravits HG. (1962)
Focal dermal hypoplasia.
Arch Dermatol ; 86 : 708-717 .

Gorham LW. (1955)
Disappearing bones.
J Bone Joint Surg [Am] ; 37 : 985-1004 .

Gorlin RJ, Goltz RW. (1960)
Multiple nevoid basal-cell epithelioma, jaw cysts and bifid ribs.
N Engl J Med ; 262 : 908-912 .

Gruhn JG, Gorlin RJ, Langer LO. (1978)
Dyssegmental Dwarfism. A Lethal Anisospondylic Camptomicromelic Dwarfism.
Am J Dis Child ; 132 : 382-386 .

Hall BD, Greenberg MH. (1972)
Hypomelia-hypotrichosis-facial hemangioma syndrome (pseudothalidomide, SC syndrome, SC phocomelia syndrome).
Am J Dis Child ; 123 : 602-608 .

Hall JG, Levin J, Kuhn JP, Ottenheimer EJ, Van Berkum KAP, McKusick VA. (1969)
Thrombocytopenia with absent radius (TAR).
Medicine ; 48 : 411-439 .

Haumont D, Pelc S. (1983)
The Mohr syndrome: are there two variants?
Clin Genet ; 24 : 41-46 .

Hayes-Roth F, Waterman A, Lenat DB. (1983)
An Overview of Expert Systems.
In:Hayes-Roth F, Waterman A, Lenat DB eds. Building Expert Systems. London: Addison-Wesley: 3-29.

Hedayati H, Barmada R, Skosey JL. (1980)
Acrolysis in pachydermoperiostosis: primary or idiopathic hypertrophic osteoarthropathy.
Arch Intern Med ; 140 : 1087-1088 .

Heselson NG, Cremin B, Beighton P. (1977)
Pseudoachondroplasia: a report of 13 cases.
Br J Radiol ; 50 : 473-482 .

Heselson NG, Raad MS, Hamersma H, Cremin BJ, Beighton P. (1979)
The radiological manifestations of metaphyseal dysplasia (Pyle disease).
Br J Radiol ; 52 : 431-440 .

Horan F, Beighton P. (1975)
Orthopedic aspects of the Schwartz syndrome.
J Bone Joint Surg [Am] ; 57 : 542-546 .

Horan FT, Beighton PH. (1978)
Osteopathia striata with cranial sclerosis. An autosomal dominant entity.
Clin Genet ; 13 : 201-206 .

Horrocks JC, de Dombal FT. (1975)
Computer-aided diagnosis of "dyspepsia".
Am J Diges Dis ; 20 : 397-406 .

Houston CS, Gerrard JW, Ives EJ. (1978)
Dysosteosclerosis.
Am J Roentgenol ; 130 : 988-991 .

Huggins E. (1979)
Microprocessors and microcomputers, their use and programming.
1st ed. London: The MacMillan Press: 1-10.

Hunter AGW, Partington MW, Evans JA. (1982)
The Coffin-Lowry syndrome. Experience from four centres.
Clin Genet ; 21(5) : 321-335 .

Johnson CF. (1976)
Diagnosing obscure syndromes from professor to television to computer scan.
Clin Pediatr ; 15(2) : 188-196 .

Kaufman RL, Rimoin DL, McAlister WH, Hartmann AF. (1974)
Variable Expression of the Holt-Oram Syndrome.
Am J Dis Child ; 127 : 21-25 .

Kelly TE, Wells HH, Tuck KB. (1982)
The Weissenbacher-Zweymuller syndrome: possible neonatal expression of the Stickler syndrome.
Am J Med Genet ; 11 : 113-119 .

Kozlowski K, Beighton P. (1984)
Gamut Index of Skeletal Dysplasias - An Aid to Radiodiagnosis.
1st ed. Berlin: Springer-Verlag: 81-178 .

Kozlowski K, Cremin B, Beighton P. (1980)
Variability of spondylo-metaphyseal dysplasia, common type.
Radiol Diagn ; 21 : 682-686 .

Kozlowski K, Sikorska B. (1970)
Dysplasia metaphysaria, Typ Vaandrager-Pena.
Z. Kinderheilk. ; 108 : 165-170. .

Kozlowski KS, Celermajer JM, Tink AR. (1974)
Humero-spinal dysostosis with congenital heart disease.
Am J Dis Child ; 127 : 407-410 .

Lachman R, Sillence D, Rimoin D et al. (1981)
Diastrophic dysplasia: the death of a variant.
Radiology ; 140 : 79-86 .

Lachman R, Zonana J, Khajavi A, Rimoin D. (1979)
The spondylometaphyseal dysplasias, clinical, radiologic and pathologic correlation.
Ann Radiol ; 22 : 125-135 .

Langer LO Jr, Carey LS. (1966)
The roentgenographic features of the KS mucopolysaccharidosis of Morquio (Morquio-Brailsford disease).
Am J Roentgenol ; 97 : 1-20 .

Langer LO Jr. (1965)
Dyschondrosteosis, A hereditable bone dysplasia with characteristic roentgenographic features.
Am J Roentgenol ; 95 : 178-188 .

Lee FA. (1972)
Radiology of the Beckwith-Wiedemann syndrome.
Radiol Clin North Am ; 10 : 261-276 .

Lemaitre L, Remy J, Farriaux JP, Dhont JL, Walbaum R. (1978)
Radiological signs of Mucolipidosis II or I-cell disease.
A study of nine cases.
Pediatr Radiol ; 7 : 97-105 .

Leroy JG, Devos EA, van den Bulcke LJ, Robb NS. (1981)
Cerebro-costo-mandibular syndrome with autosomal dominant inheritance.
J Pediatr ; 99 : 441-443 .

Leroy JG. (1975)
Dominant mesomelic dwarfism of the hypoplastic tibia, radius type.
Clin Genet ; 7 : 280-285 .

Levin SE, Zarvos P, Milner S, Schmaman A. (1980)
Arteriohepatic dysplasia. Association of liver disease with pulmonary arterial stenosis as well as facial and skeletal abnormalities.
Pediatrics ; 66 : 876-883 .

Lord J, Beighton P. (1981)
The femoral hypoplasia-unusual facies syndrome: a genetic entity?
Clin Genet ; 20 : 267-275 .

Luthy DA, Hall JG, Graham CB. (1979)
Prenatal diagnosis of thrombocytopenia with absent radii.
Clin Genet ; 15 : 495-499 .

Lux SE, Johnson RB, August CS et al. (1970)
Chronic neutropenia and abnormal cellular immunity in cartilage-hair hypoplasia.
N Engl J Med ; 282 : 231-236 .

MacDessi JJ et al. (1978)
Spondylo-Epiphyseal Dysplasia with Ocular changes. Report of two "new" variants in two different families.
Pediat Radiol ; 7 : 220-228 .

Majewski F, Ranke M, Kemperdick H, Schmidt E. (1981)
The Weaver syndrome: a rare type of primordial overgrowth.
Eur J Pediatr ; 137 : 277-282 .

Maroteaux P, Spranger JW, Wiedemann H-R. (1966)
Der metatropische Zwergwuchs.
Arch fu"r Kinderheilk ; 173 : 211-226 .

Marshall RE, Graham CB, Scott CR, Smith DW. (1971)
Syndrome of accelerated skeletal maturation and relative failure to thrive: a newly recognized clinical growth disorder.
J Pediatr ; 78 : 95-101 .

McLennan TW, Steinbach HL. (1974)
Shwachman's syndrome: A broad spectrum of bony abnormalities
Radiology ; 112 : 167-173 .

McLeod R Jr. (1983)
Management Information Systems.
2nd ed. Chicago: Science Research Associates Inc.,: 341-373 .

Melhem R, Dorst JP, Scott CI Jr, McKusick VA. (1973)
Roentgen findings in mucolipidosis III (pseudo-Hurler polydystrophy).
Radiology ; 106 : 153-160 .

Melnick JC. (1959)
Osteopathia condensans disseminata (osteopoikilosis). Study of a family of four generations.
Am J Roentgenol ; 82 : 229-238 .

Morgan JD. (1971)
Incontinentia Pigmenti (Bloch-Sulzberger Syndrome). A Report of Four Additional Cases.
Am J Dis Child ; 122 : 294-299 .

Naumoff P, Young LW, Mazer J, Amortegula AJ. (1977)
Short rib-polydactyly syndrome type 3.
Radiology ; 122 : 443-447 .

Neuhauser G, Kaveggia EG, Opitz JM. (1976)
Autosomal recessive syndrome of pseudogliomatous blindness, osteoporosis and mild mental retardation.
Clin Genet ; 9 : 324-332 .

O'Connell DJ, Hall CM. (1977)
Craniocarpotarsal dysplasia. A report of 7 cases.
Radiology ; 123 : 719-723 .

Opitz JM, France T, Herrmann J, Spranger JW. (1972)
The Stickler syndrome.
N Engl J Med ; 286 : 546-547 .

Oshawa T. (1971)
Roentgenographic manifestations of Klinefelter Syndrome.
Am J Roentgenol ; 112 : 179-184 .

Pearn J, Bloch CE, Nelson MM. (1986)
Macrodactyly simplex congenita
A case series and considerations of differential diagnosis and aetiology.
S Afr Med J ; 70 : 755-758 .

Pena SDJ, Shokeir MHK. (1974)
Syndrome of camptodactyly, multiple ankyloses, facial anomalies and pulmonary hypoplasia: a lethal condition.
J Pediatr ; 85 : 373-379 .

Phillips GN, Gordon DH, Martin EC, Jack OH, Casarella W. (1978)
The Klippel-Trenaunay syndrome: clinical and radiological aspects.
Radiology ; 128 : 429-434 .

Phillips HE, Carter AP, Kennedy JL Jr, Rosman NP, O'Connor JF. (1978)
Aicardi's syndrome; radiologic manifestations.
Radiology ; 127 : 453-455 .

Poznanski AK, Stephenson JM. (1967)
Radiographic findings in hypothalamic acceleration of growth associated with cerebral atrophy and mental retardation (cerebral gigantism).
Radiology ; 88 : 446-456 .

Reinhardt K, Pfeiffer RA. (1967)
Ulno-fibulare Dysplasie. Eine autosomal-dominant vererbte Mikromesomelie a"hnlich dem Nievergeltsyndrom.
Fortschr Roentgenstr ; 107 : 379-391 .

Reiter S. (1978)
Acrodysostosis. A case of peripheral dysostosis, nasal hypoplasia, mental retardation and impaired hearing.
Ped Radiol ; 7 : 53-55 .

Riggs W Jr, Seibert J. (1972)
Cockayne's syndrome. Roentgen findings.
Am J Roentgenol ; 116 : 623-633 .

Robinow M, Johnson GF, Verhagen AD. (1970)
Distichiasis-lymphedema. A hereditary syndrome of multiple congenital defects.
Am J Dis Child ; 119 : 343-347 .

Robinow M, Pfeiffer RA, Gorlin RJ et al. (1971)
Acrodysostosis. A syndrome of peripheral dysostosis, nasal hypoplasia and mental retardation.
Am J Dis Child ; 121 : 195-203 .

Robinow M, Silverman FN, Smith HD. (1969)
A newly recognized dwarfing syndrome.
Am J Dis Child ; 117 : 645-651 .

Saul RA, Lee WH, Stevenson RE. (1982)
Caffey's disease revisited. Further evidence for autosomal dominant inheritance with incomplete penetrance.
Am J Dis Child ; 136 : 56-60 .

Schauerte EW, St-Aubin PM. (1966)
Progressive synosteosis in Apert's syndrome (Acrocephalosyndactyly) with a description of radiographic changes in the feet.
Am J Roentgenol ; 97 : 67-73 .

Schildt H. (1985)
C Made Easy.
1st ed. Berkeley California: Osborne McGraw-Hill: 2 .

Setzer ES, Ruiz-Castaneda N, Severn C, Ryden S, Frias JL. (1981)
Etiologic heterogeneity in the oculoauriculovertebral syndrome.
J Pediatr ; 98 : 88-90 .

Shaul WL, Emery H, Hall JG. (1975)
Chondrodysplasia punctata and maternal warfarin use during pregnancy.
Am J Dis Child ; 129 : 360-362 .

Shortliffe EH. (1976)
Computer-Based Medical Consultations: MYCIN.
1st ed. New York: American Elsevier.

Sly WS, Quinton BA, McAlister WH, Rimoin DL. (1973)
Beta-glucuronidase deficiency: Report of clinical, radiologic and biochemical features of a new mucopolysaccharidosis.
J Pediatr ; 82 : 249-255 .

Smith R. (1980)
Idiopathic osteoporosis in the young.
J Bone Joint Surg [Br] ; 62 : 417-427 .

Sohr E. (1983)
Bitmaps Speed Data-handling Tasks.
Byte ; 8(10) : 480-494 .

Sparkes RS, Graham CB. (1972)
Camurati-Engelmann disease: genetics and clinical manifestations with a review of the literature.
J Med Genet ; 9 : 73-85 .

Spranger J, Kemperdieck H, Bakowski H, Opitz JM. (1978)
Two peculiar types of enchondromatosis.
Pediatr Radiol ; 7 : 215-219 .

Spranger JW, Gehler J, Cantz M. (1977)
Mucolipidosis I, a sialidosis.
Am J Med Genet ; 1 : 21-29 .

Spranger JW, Gehler J, Cantz M. (1976)
The radiographic features of mannosidosis.
Radiology ; 119 : 401-407 .

Spranger JW, Langer LO, Wiedemann H-R. (1974)
Bone Dysplasias - An Atlas of Constitutional Disorders of Skeletal Development.
1st ed: Stuttgart: Gustav Fischer Verlag.

Steel RW, Bass JW. (1970)
Hallermann-Streiff syndrome. Clinical and prognostic considerations.
Am J Dis Child ; 120 : 462-465 .

Steinbach HL, Young DA. (1966)
The roentgen appearance of pseudohypoparathyroidism (PH) and pseudopseudohypoparathyroidism (PPH). Differentiation from other syndromes with short metacarpals, metatarsals etc.
Am J Roentgenol ; 97 : 49-55 .

Stevenson RE. (1976)
The Iduronidase-Deficient Mucopolysaccharidoses: Clinical and Roentgenographic Features.
Pediatrics ; 57 : 111-116 .

Strumpf DA, Austin JH, Crocker AC, Lafrance M. (1973)
Mucopolysaccharidoses Type VI (Maroteaux-Lamy Syndrome)
1. Sulfatase B deficiency in tissues.
Am J Dis Child ; 126 : 747-756 .

Taybi H, Linder D. (1967)
Congenital familial dwarfism with cephaloskeletal dysplasia.
Radiology ; 89 : 275-281 .

Torg JS, Digeorge AM, Kirkpatrick JA Jr, Trujillo MM. (1969)
Hereditary multicentric osteolysis with recessive transmission: a new syndrome.
J Pediatr ; 75 : 243-252 .

Tremblay JP, Sorenson PG. (1976)
An Introduction to Data Structures with Applications.
1st ed. Tokyo: McGraw-Hill Kogakusha Ltd: 536-539 .

Tremblay JP, Sorenson PG. (1976)
An Introduction to Data Structures with Applications.
1st ed. Tokyo: McGraw-Hill Kogakusha, Ltd: 468-469 .

Van Greunen F, MacGregor KJ, Beighton P. (1986)
A microcomputer-based relational database for an academic department of human genetics.
S Afr Med J ; 70 : 479-482 .

Van de Kamp JJP, Van Pelt JF, Liem KO, Giesberts MAH, Niepoth LTM, Staalman CR. (1976)
Clinical variability in Sanfilippo B disease: a report on six patients in two related sibships.
Clin Genet ; 10(5) : 279-285 .

Wackenheim A. (1981)
Cheirolumbar dysostosis.
Eur J Radiol ; 1 : 189-194 .

Weisswichert PH, Knapp G, Willich E. (1981)
Accelerated Bone Maturation Syndrome of the Weaver Type.
Eur J Pediatr ; 137 : 329-333 .

Weleber RG, Beals RK. (1976)
The Hajdu-Cheney syndrome: Report of two cases and review of the literature.
J Pediatr ; 88 : 243-249 .

Williams HJ, Sane SM. (1976)
Cerebro-costo-mandibular syndrome: long term follow-up of a patient and review of the literature.
Am J Roentgenol ; 126 : 1223-1228 .

Williams P, Secrist L, Fowler GW, Gwinn JL,Dumars KC. (1972)
Roentgenographic features of the cerebrohepatorenal syndrome of Zellweger.
Am J Roentgenol ; 115 : 607-610 .

Winter RM, Donnai D, Crawfurd Md'A. (1981)
Syndromes of microcephaly, microphthalmia, cataracts and joint
J Med Genet ; 18 : 129-133 .

Young ID, Harper PS, Newcombe RG, Archer IM. (1982)
A clinical and genetic study of Hunter's syndrome. 2 differences between the mild and severe forms.
J Med Genet ; 19 : 408-411 .

Zorkoczy PI, Gaines BR, Loxton R, Maddison RN, Monk J, Pengelly RM. (1975)
Computers and their Applications.
1st ed. Walton Hall, Milton Keynes : Open University Press: 9-28

SECTION TEN

APPENDICES

APPENDIX A

Pascal Programs.

This appendix contains listings of the Pascal programs developed for this thesis. The programs are presented in the same order as in the main script. Some explanatory notes are included as comments in the program texts.

```
Program Menu;

{This small program is the main controlling procedure from which the
 application procedures are called.  It is being compiled to menu.com
 and the code and data segments are adjusted so as to accommodate the
 larger chained programs.  The chaining of the various subordinate prc
 grams is made adequately clear in the source code.

              F van Greunen       Mar 1987.          }

{$I Typevar.Pas}       {Include commonly used types and variables}

Chn_File : File;       {Variable for this program}

{$I Asslib.Pas}        {Include Assembler procedures}
{$I Library.Pas}       {Include Commonly used utilities}
{$I Proflib.Pas}       {Include Procedures designed for this project}

Begin

     Logo2;
     Msg(22,4,'DIAGNOSTIC ASSISTANT FOR SKELETAL CONDITIONS');
     Msg(8,7,'1     Enter a Disease Profile');
     Msg(8,9,'2     Change a Disease Profile');
     Msg(8,11,'3     Retrieve a Disease Profile');
     Msg(8,13,'4     Diagnostic Aid Utility');
     Msg(8,15,'5     Return to DOS');
     Textcolor(White+Blink);
     Msg(10,18,'Select a Number ');
     Textcolor(White);

     Repeat
           Read(Kbd,Ch);
           Ch := Upcase(Ch);
     Until Ch in ['1'..'5'];

     Clrscr;
     Logo2;
     Msg(10,10,'Loading appropriate program');

     Case Ch of

        '1' : Begin
                Assign(Chn_File,'Disenter.chn'); {Program for profile
                Chain(Chn_File);                 {      entry
              End;

        '2' : Begin
                Assign(Chn_File,'Profedit.Chn'); {Program for profile
                Chain(Chn_File);                 {      editing
              End;

        '3' : Begin
                Assign(Chn_File,'Dsearch.Chn');  {Program for profile
                Chain(Chn_File);                 {      retrieval
```

```
            End;

        '4' : Begin
                Assign(Chn_File,'Prospect.Chn'); {Program for diffe- }
                Chain(Chn_File);                 { rential diagnosis }
              End;

        '5' : Clrscr;
      End;


End.
```

```
Program Mkdescrp;

{ This short program is used to initially create the descriptor file
  in which the commencing addresses of the various sections of fea-
  tures are kept.  Record 0 is unused for the time being. Re-
  cords 1 to 10 point to the various sections.  This is an 'ad hoc'
  program that will not be included in the user package.

  F van Greunen  Nov 1985 modified Oct 1986                          }


Type

    String70 = String[70];
    Descrip_Rec = Record
                        Text : String70;
                        Pntr : Integer;  {This record to be entered
                                          into
                                          Data dictionary          }
                  End;

Var
   Descrip_Buff : Descrip_Rec;
   Descriptor   : File of Descrip_Rec;


Procedure Insert (Line:String70; Address : Integer; Rec:Integer);
{Assumes open Descriptor}

Begin
     Seek (Descriptor,Rec);
     With Descrip_Buff do
          Begin
               Text := Line;
               Pntr := Address;
          End;

     Write (Descriptor,Descrip_Buff);

End;
```

```
Begin    {Main Procedure  }
         Assign (Descriptor,'Descript.Ran');
         Rewrite(Descriptor);

{ The first number represents the commencing address of the
  section in the file Features.Ran, the second is the record
  of Descript.Ran. }

         Insert ('Unused',0,0);
         Insert ('Non-skeletal Features',1,1);
         Insert ('Skull X-ray Features',128,2);
         Insert ('Thorax X-Ray Features',192,3);
         Insert ('Spine X-ray Features',256,4);
         Insert ('Pelvis X-ray Features',384,5);
         Insert ('Tubular Bone X-ray Features',448,6);
         Insert ('Joints X-ray Features',576,7);
         Insert ('Hands X-ray Features',768,8);
         Insert ('Feet X-ray Features',896,9);
         Insert ('Global X-ray Features',960,10);

         Close (Descriptor);

End.
```

```
Program Features;

{This program copies the features listed in the ten text files
 Skull.txt, Thorax.txt, Spine.txt etc to the random access file
 called  Features.ran.  There is provision for the following
 number of features under each heading :

   Non-Skeletal       :128
   Skull              : 64
   Thorax             : 64
   Spine              :128
   Pelvis             : 64
   Tubular            :128
   Joints             :192
   Hands              :128
   Feet               : 64
   Global Skeletal    : 64

   This is an 'ad hoc' program for development purposes and it
   will not be included in the final user package.

   F van Greunen  Sep 1985   Revised Oct 1986  Feb 1987       }

Type
   Mystring = String[70];
   Shortstring = String[12];

Var
   Ranfile : File of Mystring;
   Textfile : Text;

Procedure Init;
{Assumes open created Feature file, "Features.ran"}

Var
   I : Integer;
   Buffer : Mystring;
Begin
     Buffer := 'XX';     {XX indicates an unused record}
     For I := 0 to 1023 do
     Write(Ranfile,Buffer);Write(Buffer);
     Writeln;
End; {Init}


Procedure Copy_over(Filename:shortstring; Itemstart:integer);

Var
   Item : Mystring;

Begin

     Assign(Textfile,Filename);
     Reset(Textfile);
```

```
      Seek(Ranfile,Itemstart);

      While not EOF(Textfile) do
      Begin
        Readln(Textfile,Item);
        Write(Ranfile,Item);
        Writeln(Item); {Also write to screen for monitoring purposes}
      End;

      Close(Textfile);

End;


Begin  {Main Procedure}

   Assign(Ranfile,'Features.ran');
   Rewrite(Ranfile); {ie. create it from scratch }

   Init;
   Reset(Ranfile);
   Copy_over('Soft.txt',0);
   Copy_over('Skull.txt',128);
   Copy_over('Thorax.txt',192);
   Copy_over('Spine.txt',256);
   Copy_over('Pelvis.txt',384);
   Copy_over('Tubular.txt',448);
   Copy_over('Joints.txt',576);
   Copy_over('Hands.txt',768);
   Copy_over('Feet.txt',896);
   Copy_over('Global.txt',960);

   Close(Ranfile);

End.
```

```
Program Disenter;

{$C-}     {Setting off check for Ctrl-C and Ctrl-S speeds up program}

                          {Program description

A program to enter disease profiles into the random file
profiles.ran .


The following modules exist.  The modules are described in
the procedure definitions :

Init;
Process_Name_Genetics;
Process_Genetics;
Process_Features;
Process_Notes;

               F van Greunen May 1986
                                        }


{$I Typevar.PAS}{Include list of commonly used ■ types and variables}

   Menu : File;      {Variable for this program}


{ GLOBAL UTILITY PROCEDURES                                            }

{$I Library.pas}

{The following procedures are added by this library:
     Msg to print messages
     Frame to draw boxes
     Stupcase changes strings to all uppercase
     Flush initializes a three dimensional intarray to zero
     Bubble_sort
     Cancel forms a vector of numbers which are not common in
     two other vectors
     Pause waits until any key is pressed
     Respond waits for  y,n,Y,N and returns either Y or N
}

{External assembler routines}
{$I Asslib.pas}

{Procedures specific for this series of programs}
{$I Proflib.pas}
```

```
Procedure Initialise;

Var
   I : Integer;

Begin
     Aborted := False;
End{Init};


Procedure Process_Name_Genetics;

{New procedure Process_Name, May 1986.  This procedure accepts the nam
 of a disease entity, one alias, one abbreviation and puts these in
 the disease_buffer.}

Var
   Response : char;
   Acceptable : Boolean;

Begin

     Acceptable := False;
     While not acceptable do
     Begin
       Logo1;
       With Disease_Buff do
          Begin
               Gotoxy (36,8); Readln(Name);
               Gotoxy(31,10); Readln(Alias);
               Gotoxy(44,12); Readln (Abbreviation);
               Gotoxy(32,14); Readln (Genetics);
          End;

       Msg (8,16,'Do you want to change?  ');{Crude editing facility}
       Respond (Response);
       If Response = 'N' then Acceptable := True;
    End;
End; {Process_Name_Genetics}
```

```
Procedure Check_Previous;

{This procedure uses the first half of the disease name and possibly
 of the alias and performs a search to see if the entity exists in
 the knowledge base }

Var
   Number : Integer;
   Searchstring : String70;

Begin
     With Disease_Buff do
     Begin
          Clrscr;
          Msg(1,10,'Checking for possible existence of '+Name);
          Searchstring := Copy(Name,1,Length(Name) div 2);
          Dsearch(Searchstring,Number);

          If (Length(Alias)>0) and (Number=-1) then

          {Not found in Name fields, look in synonym fields}

          Begin
             Searchstring := Copy(Alias,1,Length(Alias) div 2);
             Dsearch(Searchstring,Number);
          End;
     End;

     If Number > -1 then
     Begin
           Aborted := True;
           Msg(WhereX,Wherey+4,'Process aborted');
     End;
End;


Procedure Insert_Profile;

Var
   Record_Counter : Integer;
   Temp_Buff : Disease_Record;
   Disease_File : File of Disease_Record;
   Existent : Boolean;

Begin

    Logo2;
    With Disease_Buff do
         Begin
           Msg(8,8,Name);
           Msg(8,10,Alias);
           Msg(8,12,Abbreviation);
           Msg(8,14,Genetics);
              Available := False;
```

```
                Note1 := 'Test';Note2 := 'Test2';
                Note3 := 'Test3';
            End;

    Assign (Disease_File,'Profile.Ran');
    {$I-} Reset (Disease_File) {$I+};  {Test if file exists }
    Existent := (IOResult = 0);
    If not existent then rewrite(Disease_File);

    Temp_Buff.available := False;
    Record_Counter := 0; {NB!!! USE THIS COUNTER FOR PROFILE ADDRESS
                          WHEN COMBINING WITH GAMUT INDEX          }

    If Filesize(Disease_File)>0 then
    Begin
      Repeat
          Read (Disease_File,Temp_Buff);
          Record_Counter := Record_counter + 1;
      Until( Record_Counter = Filesize(Disease_File))
          or Temp_Buff.available;

      If Temp_Buff.available then
          Record_Counter := Record_Counter - 1
      Else
          Record_Counter := Record_Counter;  {Redundant statement
                                              for clarity}
    End;

    Seek (Disease_File,Record_Counter);
    Write (Disease_File,Disease_Buff);
    Msg(8,17,'Disease entered into record ');
    Writeln (Record_Counter);
    Pause;
    Close(Disease_File);

End; {Insert_Profile }


{ MAIN PROGRAM BODY     }

Begin

     Initialise;
     Process_Name_Genetics;
     Check_Previous;
     If not aborted then Process_Features;
  {  Process_Notes; Procedure not available yet }

     If not aborted then Insert_Profile;

     Assign(Menu,'Menu.com');
     Execute (Menu);   {Return to main program}

End.
```

```
 Program Prospect;
{$R+,C-}

              {Program Description


A program to compare a bit-mapped disease profile with known ones
in the file Profile.ran.  First features are presented and a selec-
tion made exactly as in Disenter.pas.  An Assembler program 'Search'
scans the Tier 0 bytes of Profile.ran for a hit.  Assembler program
'Compare' produces a vector of features common to the entered pro-
file and that of the condition found in the file.

The following procedures are employed and are described in the proce-
dure definitions:

       Init;
       Process_Features;
       Scan_Profiles;
       List_Matched_Profiles

          F van Greunen Jun 1986       }


{$I Typevar.Pas}  {Include commonly used types and variables}
Menu : File;   {variable for this program}

{$I Library.pas}

{The following procedures are added by this library:

     Msg to print messages
     Frame to draw boxes
     Stupcase changes strings to all uppercase
     Flush initializes a two dimensional intarray to zero
     Bubble_sort
     Cancel forms a vector of numbers which are not common in
     two other vectors
     Pause waits until any key is pressed
     Respond waits for  y,n,Y,N and returns either Y or N }



Procedure Init;

Var
    I : Integer;
    {Tier0_Array :Bigarray;} {Provisionally 500 }

{Tier0_Array is an array of string2 containing the first two (tier
 0) bytes of the disease profiles.              }
```

```
Begin

     Assign (Disease_File, 'Profile.ran');
     Reset (Disease_File); {ie it should exist! }
     For I := 0 to Filesize(Disease_File)-1 do
         Begin
              Read (Disease_File, Disease_Buff);
              Tier0_Array[i] := Copy(Disease_Buff.Profile,1,2);
         End;

     Number := Filesize(Disease_File);
     Aborted := False;
     Close (Disease_File);

End;


{Assembler Procedures for this program:}
{$I Asslib.pas}

{Some Procedures specific to this package}
{$I Proflib.pas}


Procedure Scan_Profiles;

Var
   Already : Intarray;
   I,M,N,P,Indicator, Position, Level : Integer;
   Response : Char;
   Found, Seen_Already : Boolean;
   Comparator : String50;
   Feature_Rec : String70;


Begin
  Found := False;
  Assign(Disease_File,'Profile.ran');
  Reset(Disease_File);
  Assign(Feature_File, 'Features.ran');
  Reset(Feature_File);
  Level := Vec_Length;
  Indicator := 0;

  While not found and(Level > 0) do
  Begin
     Position := 0;
     Clrscr;
     While not found and(Position<=Filesize(Disease_File)-1) do
     Begin
          Search(Position,Number,Tier0_array,Codestring);
          If Position > Filesize(Disease_File)-1 then
             Position :=Filesize(Disease_File)-1;
          Seen_Already := False;
         {Check whether not already inspected before }
```

```
For I := 1 to Indicator do
    If Already[i] = Position then Seen_Already := True;

If not seen_already then
Begin
     Seek(Disease_File,Position);
     Read(Disease_File,Disease_buff);
     Comparator := Disease_Buff.Profile;
     Decode(n,Comparator,B_Vec);
     Compare(Feature_Vec,Vec_Length,B_Vec,n,C_Vec,p);
     If p >= Level then
     Begin
       Writeln('     Found at Level ',Level,' out of ',
               Vec_Length);
       Writeln('     ',Disease_Buff.Name);
       Writeln('     Synonym:  ',Disease_Buff.Alias);
       Writeln; Writeln;
       Writeln('     ',Disease_Buff.Name,
       ' has the following ');
       Writeln('      features in common with the Unknown:');
       Writeln;
       For i := 1 to p do
       Begin
         Seek(Feature_File,C_Vec[i]);
         Read(Feature_File,Feature_Rec);
         Writeln('     ',Feature_Rec);
       End;
       Cancel(Feature_Vec,Vec_Length,C_vec,p,D_Vec,m);
       Pause; Clrscr;
       Writeln('     The following features in Unknown but ');
       Writeln('     not in ',Disease_Buff.name);
       Writeln;
       If m = 0 then Writeln('     None')
       Else
       Begin
        For I := 1 to m do
        Begin
             Seek(Feature_File,D_Vec[i]);
             Read(Feature_File,Feature_Rec);
             Writeln('     ',Feature_Rec);
        End;
     End;
     Cancel(B_Vec,n,C_Vec,p,D_Vec,m);
     Pause;Clrscr;
     Writeln('     ',Disease_Buff.name,' has the following ');
     Writeln('     features not in Unknown:');
     Writeln;
     If m = 0 then Writeln('     None')
     Else
     Begin
       For I := 1 to m do
       Begin
            Seek(Feature_File,D_Vec[i]);
            Read(Feature_File,Feature_Rec);
            Writeln('     ',Feature_Rec);
```

```
                End;
              End;

              Writeln; Write('    Do you want to go further?  ');
              Respond (Response);
              If response ='N' then Found := True;
              Indicator := Indicator + 1;
              Already[Indicator] := Position;
              Clrscr;
           End;
        End;
        Position := Position + 1;
     End;
     Level := Level - 1;
  End;
    Close(Disease_File);
    Close(Feature_File);
End; {Scan_Profiles}

Procedure List_Matched_Profiles;

Var
   N,P,Position : Integer;
   Comparator : String50;
   Finished : Boolean;

Begin
     Assign (Disease_File,'Profile.ran');
     Reset(Disease_File);
     Finished := False;
     Position := 0;
     Clrscr;
     Msg(10,1,'The following conditions manifest the selected stigmata
);
     Writeln;Writeln;
     While not finished do
     Begin
          Search(Position,Number,Tier0_Array,Codestring);
          If Position >= Number then
          Begin
             Finished := True;
             Writeln;Writeln('No (more) matching conditions');
          End
          Else
          Begin
               Seek(Disease_File,Position);
               Read(Disease_File,Disease_Buff);
               Comparator := Disease_Buff.Profile;
               Decode(n,Comparator,b_Vec);
               Compare(Feature_Vec,Vec_Length,B_Vec,n,C_Vec,p);
               If P=Vec_Length then Writeln('     ',Disease_Buff.Name
          End;
          Position := Position + 1;
     End;

 Close(Disease_File);
```

```
End;


Begin {Main Procedure }
      Init;
      While not aborted do
      Begin
          Process_Features;
          If not aborted then
          Begin
             Logo2;
             Msg(10,10,'D   Differential Diagnosis');
             Msg(10,12,'L   List of matching Diseases');
             Msg(10,14,'X   Exit Program');
             Textcolor(White+Blink);
             Msg(16,18,'Select a Letter');
             Textcolor(White);

             Repeat
                Read(Kbd,Ch);
                Ch := Upcase(Ch);
             Until Ch in ['D','L','X'];

             Case Ch of
               'D' : Scan_Profiles;

               'L' : Begin
                           List_matched_profiles;
                           Pause;
                     End;

               'X' : Aborted := True;

             End {Case};
          End;
          Clrscr;
      End;

Assign(Menu,'Menu.Com');
Execute(Menu);

End.
```

```
Program Dsearch;

{ This program searches the names and synonyms in the file
  Profiles.Ran for the occurrence of a searchstring obtained
  from the first half of an entered string.  If a match is
  found, it is presented on the screen.  Should the find in-
  dicate the desired disease condition, the relevant informa-
  tion is presented.

                F van Greunen  Jul 1986       }

{$I Typevar.Pas}  {Include commonly used types and variables}

I, Place, Page: Integer;    { Additional variables used     }
Feature : String70;         {             by                }
Menu : File;                {       this program            }

{$I Library.pas} {Include the libraries described before}
{$I Asslib.pas}
{$I Proflib.pas}

Begin
  Logo2;
  Gotoxy(10,10);
  Write('Enter Disease Name or Part:    ');
  Readln(Searchstring);
  Searchstring := Copy(Searchstring,1,Length(Searchstring) div 2);
  Dsearch(Searchstring,Place);
  If Place > -1 then
  Begin

    Assign (Disease_File,'Profile.ran');Reset(Disease_File);
    Seek(Disease_File,Place);
    Read(Disease_File,Disease_Buff);
    Page :=0;
    Finished := False;
    Logo2;
    With Disease_Buff do
    Begin
      Msg(9,8,Name);
      Msg(9,10,Alias);
      Msg(9,12,Abbreviation);
      Msg(9,14,'Genetics: '+Genetics);
      Msg(9,16,Note1);
      Msg(9,18,Note2);
      Msg(9,20,Note3);
      Pause;
      Clrscr;
      Writeln('   The following manifestations have been described:')
      Writeln ('  (Not necessarily in order of importance!)');
      Writeln;
      Pause;
      While not finished do
      Begin
```

```
        Decode (Number,Profile,Feature_Vec);
        Assign (Feature_File,'Features.ran'); Reset (Feature_File);
        Clrscr;
        For I := Page*20+1 to Page*20+20 do
        Begin
          If I <= Number then
          Begin
            Seek (Feature_File,Feature_vec[i]);
            Read(Feature_File,Feature);
            Writeln ('      ',Feature);
          End;
        End;
        If I < Number then
        Begin
          Textcolor(White+Blink);
          Msg(6,24,'MORE ON NEXT PAGE...');
          Textcolor(White);
        End;
        Close (Feature_File);

        Msg(6,25,'N<ext>  P<rev page> <e>X<it>');
        Repeat Read(kbd,ch);
             Ch := Upcase(ch);
        Until Ch in ['N','P','X'];

        Case Ch of
                  'N' : Begin
                          If I < Number then
                            Page := Page + 1;
                        End;

                  'P'  : Begin
                           Page := Page-1;
                            If Page < 0 then Page := 0;
                         End;

                  'X'  :  Begin
                            Finished := True;
                            Clrscr;
                          End;
                End; {Case}
          End;
        End;
        Close(Disease_File);
      End
        Else
        Begin
          Gotoxy(10,12);
          Writeln ('Unsuccessful Search'+chr(7));
          Pause;
        End;
Clrscr;
Assign(Menu,'Menu.com');
Execute (Menu);
End.
```

```
Program Testlook;

{ This program fetches any disease record of which the number
  is entered after the prompt.  It was developed for test pur
  poses and will not be included in the final user package.
                    F van Greunen May 1986.                    }


{$I Typevar.pas}  {Include commonly used types and variables }

I, Page : Integer;  { Variables used  }
Testvec : Intarray; {       by        }
Feature : String70; {  this program   }


{ INCLUDE GLOBAL UTILITY PROCEDURES                            }

{$I Asslib.pas}
{$I Library.pas}


Begin
  Assign(Disease_File,'Profile.ran'); Reset(Disease_File);
  Write ('Enter a number  '); Readln (Chapter);
  If Chapter < Filesize(Disease_File) then
  Begin
    Seek (Disease_File,chapter);
    Read (disease_File,Disease_Buff);
    Clrscr;
    Finished := False;
    Page := 0;    {Pages in case to many features for one screen}
    With Disease_Buff do
    Begin
      Writeln (Name);Writeln ('Synonym :',alias);
      Writeln;Writeln;
      Writeln ('Genetics:  ',genetics);
      Writeln;
      Writeln('The following manifestations have been described:');
      Writeln ('(Not necessarily in order of importance!)');
      Writeln;
      Pause;
      While not finished do
      Begin
        Decode (Number,Profile,Testvec);
        Assign (Feature_File,'Features.ran'); Reset (Feature_File);
        Clrscr;
        For I := Page*20+1 to Page*20+20 do
        Begin
          If I <= Number then
          Begin
            Seek (Feature_File,Testvec[i]);
            Read(Feature_File,Feature);
            Writeln (Feature);
          End;
        End;
```

```
        If I < Number then
        Begin
          Textcolor(White+Blink);
          Msg(1,24,'MORE ON NEXT PAGE...');
          Textcolor(White);
        End;
        Close (Feature_File);

        Msg(1,25,'N<ext>  P<rev page> <e>X<it>');
        Repeat Read(kbd,ch);
          Ch := Upcase(ch);
        Until Ch in ['N','P','X'];
        Case Ch of
          'N' : Begin
                  If I < Number then
                  Page := Page + 1;
                End;

          'P' : Begin
                  Page := Page-1;
                  If Page < 0 then Page := 0;
                End;

          'X' : Begin
                  Finished := True;
                  Clrscr;
                End;
        End; {Case}
      End;
    End;
  End

  Else Writeln('Number too large!');

  Close (Disease_File);

End.
```

```
Program Profedit;

{     This program is used to make changes to a disease description.
      Since such changes can jeopardise the integrity of the knowledge
      base, it is password controlled.  Changes may be made to name,
      synonym, abbreviation, notes and genetics by simple editing.
      Diagnostic features may be added or deleted.

                        F van Greunen Mar 1987                          }


   {$I Typevar  } {Include commonly used types and variables }
   Position : Integer;  {   variables used      }
   Acceptable : Boolean;{        by             }
   Response : Char;     {       this            }
   Menu:File;           {      program          }

{ GLOBAL UTILITY PROCEDURES                                             }

{$I Library.pas}

{The following procedures are added by this library:
      Msg to print messages;
      Frame to draw boxes;
      Stupcase changes strings to all uppercase;
      Flush initializes a three dimensional intarray to zero;
      Bubble_sort;
      Cancel forms a vector of numbers which are not
      common to two other vectors;
      Pause waits until any key is pressed;
      Respond waits for  y,n,Y,N and returns either Y or N;
      String_Edit allows for changes to character strings.
}


{External assembler routines}
{$I Asslib.pas}

{Procedures specific for this series of programs}
{$I Proflib.pas}

Procedure Initialise;
Var
   Searchstring : String70;

Begin
  Aborted := False;
  Finished := False;
  Msg(8,12,'Enter Disease Name: ');
  Readln (Searchstring);
  If Length(Searchstring) > 1 then  {Length must be at least 2}
  Begin

    { Lessen selectivity of search by dividing the searchstring }
```

```
    Searchstring := Copy(Searchstring,1,Length(Searchstring) div 2);
    Dsearch(Searchstring,Position);
    If Position = -1 then
    Begin
      Clrscr;
      Writeln('      Search unsuccessful'+chr(7));
      Pause;
      Assign (Menu,'Menu.Com'); Execute (Menu);
    End;
  End

  Else

  Begin
    Msg(10,20,Chr(7)+'String too short, search aborted');
    Pause;
    Assign (Menu,'Menu.Com'); {Return to main menu }
    Execute (Menu);
  End;
End;

Procedure Edit_Name_Genetics;

{ This procedure provides for elementary text editing of the name,
  synonym, abbreviation and genetic description of disease profile.}

Begin

  Assign(Disease_File,'Profile.ran'); Reset(Disease_File);
  Seek(Disease_file,Position);
  Read(Disease_File,Disease_Buff);
  Logo1;

  With Disease_Buff do
  Begin
    Gotoxy (36,8); Writeln(Name);
    Gotoxy(31,10); Writeln(Alias);
    Gotoxy(44,12); Writeln (Abbreviation);
    Gotoxy(32,14); Writeln (Genetics);
  End;

  Msg (8,18,'Do you want to change?  '); { Crude editing facility }
  Respond (Response);
  If Response = 'N' then Acceptable := True
  Else Acceptable := False;

  {$V-}   {Disable checking of length of string parameters   }

  While not acceptable do
  Begin
    With Disease_Buff do
    Begin
      String_Edit(Name,36,8);
      String_Edit(Alias,31,10);
      String_Edit(Abbreviation,44,12);
```

```
      String_Edit(Genetics,32,14);
    End;

    Msg (8,18,'Do you want to change?  ');{ Crude editing facility }
    Respond (Response);
    If Response = 'N' then Acceptable := True;
  End;

  {$V+}  {Restore length checking }

  If not aborted then
  Begin
    Seek(Disease_File,Position);
    Write(Disease_File,Disease_Buff);
    Close(Disease_File);
    Writeln('     Edited Profile entered into Knowledge Base');
  End
  Else
    Close(Disease_File);

End; {Edit_Name_Genetics}


Procedure Edit_Features;

{This procedure deletes features from or adds features to a known
 disease profile}

Var
  I,Length,Number,Page : Integer;
  Ch : Char;
  Feature_Rec : String70;
  Temp_Array : Intarray;
  Finished : Boolean;

Begin{0}
  Assign (Disease_File, 'Profile.Ran'); Reset (Disease_File);
  Seek (Disease_File,Position);          {Get disease record }
  Read (Disease_File,Disease_Buff);      {into buffer again}
  Close (Disease_File);
  Finished := False;
  Page := 0;     {Pages in case more features than can fit on screen}

  While not finished do
  Begin {1}
    Decode(Length,Disease_Buff.Profile,Temp_Array);
    Assign (Feature_File,'Features.Ran');
    Reset(Feature_File);
    Clrscr;
    For I := Page*20+1 to Page*20+20 do {Shows 20 features at a time
    Begin{2}
      If I <= Length then
      Begin{3}
        Seek (Feature_File,Temp_Array[i]);
        Read (Feature_File,Feature_Rec);
```

```
    Writeln('      ',I:2,' ',Feature_Rec);
  End{3};
End{2};
Close (Feature_File);

Msg(6,25,'<A>dd <D>elete <N>ext page <P>rev page  e<X>it <Q>uit');

Repeat Read(Kbd,ch);
  Ch := Upcase(ch);
Until Ch in ['A','D','N','P','Q','X'];

Case Ch of

  'N' : Begin{2}
          If Length > (Page+1)*20 then
          Page := Page + 1;
        End{2};


  'P'  : Begin{2}
           Page := Page -1;
           If Page < 0 then Page := 0;
         End{2};


  'D'  : Begin{2}
           Msg(6,23,'Enter number of feature to be deleted ');
           Readln(Number);
           I := 1;
           While (I <> Number) and (I <= Length) do
           I := I+1;
           For I := I to Length -1 do    {Remove from vector}
           Temp_Array[i] := Temp_Array[i+1];
           Length := Length - 1;
           Encode (Length,Temp_Array,Codestring);
           Disease_Buff.Profile := Codestring;
         End{2};


  'A'  : Begin{2}
           Process_Features;

           {NB! Process_Features will replace the profile
           string of disease_buff with the codestring re-
           presenting the newly selected features.  The
           profile must therefore be decoded and the numbers
           added to the vector and then re-encoded    }


           Decode(Number,Disease_Buff.Profile,Feature_Vec);

           For I := 1 to Number do
             Temp_Array[Length + I] := Feature_Vec[i];

           Length := Length + Number;
```

```
                Bubble_Sort(Temp_Array,Length);
                Encode(Length,Temp_Array,Codestring);
                Disease_Buff.Profile := Codestring;
              End{2};


      'X'   : Begin{2}
                Assign (Disease_File,'Profile.Ran');
                Reset (Disease_File);
                Seek (Disease_File,Position);
                Write (Disease_File,Disease_Buff);
                Close (Disease_File);
                Finished := True;
              End{2};


      'Q'   : Finished := True;

    End;{Case}
  End{1};
End{0};

Procedure Edit_Notes;

{ Elementary text editing of the strings that constitute the notes. }

Begin
  Assign(Disease_File,'Profile.ran'); Reset(Disease_File);
  Seek(Disease_file,Position);
  Read(Disease_File,Disease_Buff);
  Logo2;


  With Disease_Buff do
  Begin
    Msg(20,5,Name);
    Msg(9,10,Note1);
    Msg(9,12,Note2);
    Msg(9,14,Note3);
  End;

  Msg (8,18,'Do you want to change?  ');      { Crude editing facility
  Respond (Response);
  If Response = 'N' then Acceptable := True
  Else Acceptable := False;

  {$V-}
  While not acceptable do
  Begin
    With Disease_Buff do
    Begin
      String_Edit(Note1,9,10);
      String_Edit(Note2,9,12);
      String_Edit(Note3,9,14);
    End;
```

```
    Msg (8,18,'Do you want to change?  ');    { Crude editing facility
    Respond (Response);
    If Response = 'N' then Acceptable := True;
  End;
  {$V+}
  Seek(Disease_File,Position);
  Write(Disease_File,Disease_Buff);
  Close(Disease_File);
End;


{Main Program}

Begin

  Logo2;
  Msg(8,10,'Enter Password'+chr(7));
  Gotoxy(8,14);
  Password(Acceptable);
  If acceptable then
  Begin
    Initialise;
    While (not finished and not aborted) do
    Begin
      Logo2;
      Msg(9,10,'D  Edit Disease Name, Synonym, Abbreviation, Genetics'
      Msg(9,12,'F  Edit (Add or delete) Features');
      Msg(9,14,'N  Edit Notes ');
      Msg(9,16,'X  Exit Program');
      Textcolor(White+Blink);
      Msg(15,18,'Select a Letter');
      Textcolor(White);

      Repeat
        Read(Kbd,Ch);
        Ch := Upcase(Ch);
      Until Ch in ['D','F','N','X'];

      Case Ch of

        'D': Edit_name_genetics ;
        'F': Edit_Features;
        'N': Edit_Notes;
        'X': Finished := True;
      End;{Case}
    End;

    If not aborted then
    Begin
      Logo2;
      Msg(9,10,'Edit complete, returning to menu');
      Pause;
    End;
```

```
    Clrscr;
  End
  Else
    Begin
      Writeln('Password rejected'+chr(7));
      Pause;
    End;


  Assign (menu,'Menu.com');
  Execute(Menu);

End.
```

```
{LIBRARY.PAS }

{This library contains frequently used procedures for inclusion
 in Pascal programs.  Brief descriptions are given with each pro-
 cedure
                F van Greunen  1985 - 1987          }


{ GLOBAL UTILITY PROCEDURES                                        }


Procedure Msg(x,y:integer;s:string70);

{  prints a message s at coordinates x,y                           }

Begin
     gotoxy(x,y);
     write(s);
End; {Msg}



Procedure Frame(upperleftx,upperlefty,
                lowerrightx,lowerrighty:integer);

{draws a double-rimmed box using IBM ASCII character set           }

Var
   I : Integer;

Begin
     clrscr;
     gotoxy(upperleftx,upperlefty);write(chr(201));
     for i:= upperleftx+1 to lowerrightx-1 do write(chr(205));
     write (chr(187));
     for i:= upperlefty+1 to lowerrighty-1 do
     begin
          gotoxy(upperleftx,i);write(chr(186));
          gotoxy(lowerrightx,i);write (chr(186));
     end;
     gotoxy(upperleftx,lowerrighty);
     write(chr(200));
     for i:=upperleftx+1 to lowerrightx-1 do write(chr(205));
     write(chr(188));
End; {Frame}



Function Stupcase(St:String70):String70;

{Converts all the characters of a string to uppercase. }

Var
   J : Integer;
```

```
Begin
     For j := 1 to length(St) do
         st[j] := upcase(st[j]);
     Stupcase := st
End; {Stupcase}


Procedure Flush (Var Ary : Bytearray; l,m,n:integer);

{Initializes a two dimensional intarray to zero }

Var
   I,J,K : Integer;

Begin
     For I := 1 to l do
     Begin
          For J := 1 to m do
          Begin
            For K := 1 to n do
              Ary[i,j,k] := 0;
          End;
     End;
End; {Flush}


Procedure Bubble_sort (Var Ary : Intarray; n:integer);

{According to standard algorithm taken from An Introduction to Data
 Structures by Tremblay and Sorenson  p468.  It sorts an integer
 vector into ascending order.                                        }

Var
   K,J,Temp : Integer;
   Sorted : Boolean;

Begin
     Sorted := False; K := 1;
     While (not sorted) and ( K < n) do
     Begin
          Sorted := True;
          For J := 1 to n-k do
          Begin
               If Ary[J+1] < Ary[J] then
               Begin
                    Temp := Ary[j];
                    Ary[j] := Ary[j+1];
                    Ary[j+1] := Temp;
                    Sorted := False;
               End;
          End;
     K := K + 1;
```

```
     End;
End; {Bubble_sort}


Procedure Cancel(Vec1:intarray; Count1:integer; Vec2:intarray;
                 Count2:integer;
                 var Vec3:intarray; var Count3:integer);

{ This procedure forms a vector3 containing those numbers of vector1
  which are not present in vector2 }

Var

  I,J,K  : Integer;

Begin
     J := 1; K := 1;

     For I := 1 to Count2 do
     Begin
          While Vec1[j] < Vec2[i] do
          Begin
               Vec3[k] := Vec1[j];
               J := J + 1;
               K := K + 1;
          End;

          J := J + 1;
     End;

     For I := J to Count1 do
     Begin
          Vec3[k] := Vec1[i];
          K := K + 1;
     End;

     Count3 := K - 1;

End; {Cancel }


Procedure Fill(Var Ary:Intarray; p: Integer);
{Fills the one dimensional intarray with p }

Var
   I : Integer;
Begin
     For I := 1 to 100 do Ary[i] := p;

End;
```

```
Procedure Pause;
{Waits until a key is pressed }

Var
   ch : char;

Begin
     Writeln;
     Msg (9,Wherey,'Press any key to continue...');
     Ch := chr(255);
     Repeat
           Read(Kbd,ch)
     Until ch <> chr(255);
End;{Pause}


Procedure Respond(Var ch:char);
{Accepts only  y,Y,n,N  to a question}
Begin
     Repeat
           Read(Kbd,ch);
           Ch := Upcase(Ch);
     Until ch in ['Y','N'];
End; {Respond}


Procedure String_Edit (var Editstr:string70;x,y:integer);
{Affects simple editing to a single string }
Var
   Finished, Accepted : Boolean;
   Reference, Position, L, I : Integer;
   Ch : Char;

Begin
     Reference := X;Position := 1;
     Textcolor(White+Blink);
     Msg(1,23,'IN EDIT MODE     ');
     Textcolor(White);
     Msg(25,23,'<Return> ends editing of particular string');
     Msg(1,25,'>  moves cursor right, <  moves left, #  deletes char')
     Msg(54,25,'                          ');
     Gotoxy(x,y);
     Textcolor(Black);TextBackground(White);
     Write(Editstr);Gotoxy(x,y);
     L := Length(Editstr);
     Finished := False;
     While not finished do
     Begin
       Repeat
         Read (Kbd, Ch)
       Until Ch in ['A'..'Z','a'..'z','0'..'9','<','>','#','*',
                    ':',';','"','?','.',',','(',')','-','/',
                     chr(13),chr(32),chr(39)];
       Case ch of
              '>'  : Begin  {Moves cursor to the right }
```

```
                        Write(Copy(Editstr,Position,1));
                        Position := Position + 1;
                      End;

              '<'   : If Position > 1 then   {Moves cursor to the left
                      Begin
                        Write(chr(8));
                        Position := Position-1;
                      End;

              '#'   : Begin    {Deletes a character }
                        Delete(Editstr,Position,1);
                        Gotoxy(Reference+Length(Editstr), Wherey);
                        Write (chr(32));
                        Gotoxy (Reference,Wherey);
                        Write (Editstr);
                        Gotoxy(Reference+Position-1,Wherey);
                      End;

        Else
                      Begin
                        If ch <> chr(13) then

                       Begin     {Insert a character }

                          Insert(Ch,Editstr,Position);
                          Position := Position + 1;
                          Gotoxy (Reference, Wherey);
                          Write (Editstr);
                          Gotoxy (Reference+Position-1, Wherey);
                        End

                        Else

                        Begin   {Carriage return ends editing }
                          Finished := True;
                          Gotoxy (Reference, Wherey);
                          Textcolor(White);TextBackground(Black);
                          For I := 0 to L do write(' ');
                          Gotoxy(Reference,Wherey);
                          Write(Editstr);
                          Gotoxy(Reference,Wherey);
                        End;
                      End;
          End;
      End;
{     End;}

End;
```

```
{LIBRARY PROFLIB.PAS  }
{This library contains frequently used procedures specific to the
application programs described in the thesis:

    Microcomputer-assisted diagnosis of inherited diseases of the
                          Skeleton.

                  F van Greunen  1986 - 1987       }


Procedure Logo2; Forward;  {See further down  }


Procedure Present_Features (number,page : integer);

{This subprocedure lists  20 features on the screen at any one time,
 starting at the record of the features file indicated by parameter
 'number'.  It assumes an open feature_file.  }

Var
   Index : Integer;
   Feature_Rec : String70;

Begin

  Clrscr;
  Seek (Feature_File,number);       {Assumes open feature file}
  Index := 1; End_of_list := false;
  Read (Feature_File,feature_Rec);
  While (Index <= 20) and (feature_rec <> 'XX') do
  Begin
    Writeln ('       ',Feature_Rec);{5 spaces added. Temp measure}
    Index := index + 1;
    Read (Feature_File,Feature_Rec);
  End;

  For Index := 1 to 20 do
  Begin
    If Markarray[chapter,page,index] <> 0 then
    Begin
      Gotoxy(1,markarray[chapter,page,index]); { Mark a pre-  }
      Writeln (chr(175));                      { viously selec- }
    End;                                          { ted feature. }

    If feature_rec = 'XX' then end_of_list := true;
  End;

Msg(1,25,'A<ccept> C<hange> D<own> E<xit> N<ext page> P<rev page>  ');
Msg (62,25,' R<emove>  U<p>');
Gotoxy(1,1);

End; {Present_Features}
```

```
Procedure Process_Features;

{ This module presents selected extracts from the knowledge-base of
  known Radiological and clinical features, from which some may be
  selected for inclusion in the disease profile.  Any number may be
  selected and provision is made to reject features that may have been
  selected erroneously.    The features are kept under the following
  headings which also serve as   items on a selection menu:

                  NON-SKELETAL (INCLUDING SOME CLINICAL FEATURES)
                  SKULL
                  THORAX
                  SPINE
                  PELVIS
                  TUBULAR BONES
                  JOINTS
                  HANDS
                  FEET
                  GLOBAL


  A small descriptor file contains the addresses where the various
  headings commence in the random file.  The clinical features are now
  incorporated   in the non-skeletal file (Soft.txt).          }


Type
    Descrip_Rec = Record
                    Text : String70; {No use for this string as yet  }
                    Pntr  : Integer;
                  End;
Var
   J,K,Last_Val,Marker : Integer;
   Finished : Boolean;
   Descrip_Buff  : Descrip_Rec;
   Feature_Rec : String70;
   Descriptor    : File of Descrip_Rec;
   Ch, Response : Char;


Procedure Select_Features (Marker : Integer);

{Subprocedure of Process_Features whereby features are selected
 from lists presented by procedure Present_Features.  }

Var
   J,Page,Pntr,Index, Temp : Integer;
   Warning : String[20];
   Section_Finished, Accepted, Acceptable : Boolean;
   Ch : char;

Begin

 Assign (Feature_file,'Features.ran');
 Reset (Feature_file);
```

```
Seek (Feature_file,Marker); {Marker is a parameter from calling proc
Index := Marker;
Page := 1;
Present_Features (Index,page);
Section_Finished := False;
While not Section_finished do
Begin
  Repeat
    Read(Kbd,ch);
    Ch := Upcase(ch);
  Until ch in ['A','C','D','E','N','P','R','U'];

  Case ch of
    'A' : Begin                                {Accept feature routine}
            Pntr := Wherey + Index-1;
            J := 0;
            Repeat
              J := J + 1
            Until (Markarray[chapter,page,j] = 0) or { Record where
            (Markarray[chapter,page,j] = Wherey);     { mark is being
                                                      { made
            Markarray[chapter,page,j] := Wherey;
            Write ('     ',Chr(175)+chr(13)+chr(10)); {Make mark}
            J := 0;
            Repeat
              J := J+1
            Until Feature_Vec[j] <1;     { Could be -1 indicating a  }
            If Feature_Vec[j] = 0 then   { deleted feature           }
              Vec_Length := Vec_Length + 1
            Else {Vec_Length unchanged}; {feature simply replaced}
            Feature_Vec[j] := Pntr;
          End;

    'C' : Begin

            Pntr := Wherey + Index -1;
            Temp := Wherey;
            Msg(1,23,'Enter Password'+chr(7));
            Gotoxy(16,23);
            Password(Accepted);

            If accepted then
            Begin

              Seek (Feature_File,Pntr);
              Read (Feature_File, Feature_Rec);
              String_Edit(Feature_Rec,8,Temp);

Msg(1,23,'

Msg (1,25,'A<ccept> C<hange> D<own> E<xit> N<ext page> P<rev page>');
     Msg (62,25,' R<emove>  U<p>');

              Gotoxy(1,Temp);
              Seek (Feature_File,Pntr);
```

```
          Write(Feature_File,Feature_Rec);
        End;
      End;

'D' : Write (Chr(10));     {Down Cursor Routine, simple line feed

'E' :  {Exit from Section Routine  }
            Section_Finished := True;

'N' : Begin                    {Next Page Routine  }

        If end_of_list then
        Begin
          Msg(20,22,chr(7)+'END OF LIST');
          Gotoxy(1,1);
        End

        Else
        Begin
          Page := Page + 1;
          If page > 13 then Page := 13; { No section has more  }
          Index := Index + 20;          { than 20*13 features  }
          Present_Features(Index,page);
        End;
      End;

'P' : Begin                          {Previous Page Routine  }
        Index := Index - 20;
        If index < Marker then  { ie do not go into previous sec-
                  Index := Marker;      { tion }

        Page := Page -1;
        If page < 1 then
        Begin
          Warning := chr(7)+'BEGINNING OF LIST';
          Page := 1;
        End
        Else Warning := ' ';

        Present_Features (Index,page);
        Msg (20,22,Warning);
        Gotoxy(1,1);
      End;



'R' : Begin                           {Remove Feature Routine  }
        Pntr := Wherey + Index-1;
        J    := 1;
        While (Pntr <> Feature_Vec[j]) and (J<100) do
          J := J + 1;
          If Feature_Vec[j] = Pntr then
          Begin
            Feature_Vec[J] := -1;  {To denote deletedfrom vector
            J := 0;
```

```
                Repeat
                  J := J+1
                Until (Markarray[chapter,page,j] = Wherey)
                              or (J = 100);
                If markarray[chapter,page,j] = Wherey then   { Delete }
                Markarray[Chapter,Page,J] := 0;     {from list of marks}

                Write ('      ',Chr(32)+chr(13)+chr(10)); {Remove mark }
              End

              Else {Nothing};
            End;

     'U' : Gotoxy (Wherex,Wherey-1);       {Up Cursor Routine }

    End;
  End;
End  {Select_Features};

  {Main Body of Procedure Process_Features }

Begin
  Flush (Markarray,10,13,20);
  Assign (Descriptor,'Descript.ran');
  Reset (Descriptor);
  Finished := False;
  For J := 1 to 100 do          { Initialize features vector  }
    Feature_Vec[j] := 0;
  Vec_Length := 0;
  While not finished do
  Begin
    Clrscr;
    Frame (20,5,60,22);
    Msg(22,7,'A   NON-SKELETAL');
    Msg(22,8,'B   SKULL');
    Msg(22,9,'C   THORAX');
    Msg(22,10,'D   SPINE');
    Msg(22,11,'E   PELVIS');
    Msg(22,12,'F   TUBULAR BONES');
    Msg(22,13,'G   JOINTS');
    Msg(22,14,'H   HANDS');
    Msg(22,15,'I   FEET');
    Msg(22,16,'J   GLOBAL SKELETAL');
    Msg(22,17,'X   EXIT           Q TO ABORT ENTIRELY');

    Textcolor(White+Blink);
    Msg(22,19,'Select Letter ');
    Textcolor(White);

    Repeat
      Read(Kbd,ch);
      Ch := Upcase(ch);
    Until Ch in ['A'..'J','Q','X'];

    If ch in ['X','Q'] then
```

```
  Begin
    Finished := True;
    If ch = 'Q' then Aborted := True;
  End

  Else
  Begin
    Chapter := Ord(ch) - 64;
    Seek(Descriptor,Chapter);  {Note for the time being that
                                chapter happens to
                                match the first few records
                                of the descriptor file.    }

    Read(Descriptor,Descrip_buff);{ ie where particular }
    Marker := Descrip_buff.pntr;  { section starts }
    Select_Features(marker);
  End;
End;

Clrscr; Gotoxy(1,1);
Bubble_Sort(Feature_Vec,Vec_length);
J := 1;  {Remove any leading -1's indicating deleted features}
While Feature_Vec[j] = -1 do
Begin
  J := J + 1;
  Vec_Length := Vec_Length - 1;
End;

If Vec_Length > 0 then
Begin
  J := J - 1;
  {Adjust vector and remove any duplicate entries }
  Last_Val :=-1;       { Initial impossible last value }

  For K := 1 to Vec_Length do
  Begin
    If Feature_Vec[k+j] <> Last_Val then
    Begin
      Feature_Vec[k] := Feature_Vec[k+j]; { Shift to beginning  }
      Last_Val := Feature_Vec[k];         { if necessary }
    End

    Else
    Begin
      J := J + 1;
      Vec_Length := Vec_Length -1;      { Eliminate the duplicate }
      K := K - 1;
    End;
  End;


  Encode (Vec_length,Feature_Vec,Codestring);
  Disease_Buff.Profile := Codestring; {Enter Profile byte into
                                       disease record}
End
```

```
  Else
    Aborted := True;
    Close(Feature_file);
    Close (Descriptor);

End; {Process_Features }


Procedure Dsearch (Searchstr : string70;var Position: Integer);

Var
   I,Number,Counter : Integer;
   Disease_Buff : Disease_Record;
   Disease_File : File of Disease_Record;
   Feature_File : File of String70;
   End_of_List,Found,Existent : Boolean;
   Feature : String70;
   Testvec : Intarray;
   Response : Char;
   Teststr : String70;

Begin
     {$V-}
  End_of_List := False;
  Assign (Disease_File,'Profile.Ran');
  {$I-} Reset (Disease_File) {$I+};  {Test if file exists}
  Existent := (IOResult = 0);
  If existent then
  Begin
    Counter := 0;
    While (not End_of_List) and (Counter < Filesize(Disease_File)) do
    Begin
      Read(Disease_File,Disease_Buff);
      Scan(Disease_Buff.name,Searchstr,Found);
      If (not found) and( length(Disease_Buff.alias) >0) then
      Scan(Disease_Buff.alias,Searchstr,Found);
      If found then
      Begin
        Clrscr;
        With Disease_Buff do
        Begin
          Logo2;
          Msg(8,8,'      '+Name);
          Msg(8,10,'      Synonym:   '+Alias);
          Msg(8,12,'      Genetics: '+Genetics);

          Writeln;
          Msg(8,14,'      This profile exists at record ');
          Write (Counter);
          Msg(8,16,'      Does it relate? ');
          Respond(Response);
          If Response = 'Y' then End_Of_List := True;
        End;
      End;
```

```
      Counter := Counter + 1;
     End;

     If (Counter = Filesize(Disease_File)) and (not end_of_list) then
        Position := -1
     Else Position := Counter-1;
     Close (Disease_File);
   End

   Else Position := -1;

{$V+}

End;

Procedure Logo1;

Begin
  Clrscr;
       Frame (6,1,78,22);

       Msg (20,3,'DEPT OF HUMAN GENETICS  UNIVERSITY OF CAPE TOWN');
       Msg (8,8,'Disease Name, max 50 chars: ');
       Msg (8,10,'Synonym, max 50 chars: ');
       Msg (8,12,'Standard Abbreviation, max 6 chars: ');
       Msg (8,14,'Genetics, max 50 chars: ');
End;

Procedure Logo2;

Begin
  Clrscr;
       Frame (6,1,78,22);
       Msg (20,3,'DEPT OF HUMAN GENETICS  UNIVERSITY OF CAPE TOWN');
End;
```

```
{Library ASSLIB.PAS

 This library contains the references to the assembler procedures
 used by certain of the application programs.  When  included in
 a source (.pas) file, the code from the corresponding .com files
 is incorporated with that of the main calling program during com-
 pilation.

                                F van Greunen  1986  }


Procedure Scan(var Str1:string70;var str2:string70;var Boole:boolean);
          external 'Scan.com';

{Str2 is compared with sections of str1 from left to right.  If a
 match is found, Match is returned as true }

Procedure Password (var Boole: Boolean); external 'Passwd.com';

Procedure Encode (n:integer;var vec:intarray;var codestr:string50);
          External 'Convert.com';

Procedure Decode (var n:integer;var codestr:string50; var vec:intarray);
          External 'Decode.com';

Procedure Search (var Position:integer;var Number:integer;var
          vec:Bigarray;var Codestr:string50); External 'Search.com';

Procedure Compare(var Avec:intarray;var m:integer;var Bvec:intarray;
                      var n:integer;var Cvec:intarray;var p:integer);
                      External 'Compare.com';
```

```
{LIBRARY TYPEVAR.PAS
 This library contains frequently used type definitions and va-
 riables used by several of the application programs.

                    F van Greunen Mar 1987     }

Type
    String70 = string[70];
    String50 = string[50];
    String30 = string[30];
    String6  = string[6];
    String4  = string[4] ;
    String2  = string[2];

    Disease_Record =
                    Record
                          Available    : Boolean;
                          Profile      : String50;
                          Name         : String50;
                          Alias        : String50;
                          Abbreviation : String6;
                          Note1        : String70;
                          Note2        : String70;
                          Note3        : String70;
                          Genetics     : String50;
                    End;


   Intarray = array[1..100] of integer;
   Bytearray = array[1..10,1..13,1..20] of byte;
   Bigarray = array[0..500] of string2;

{ GLOBAL VARIABLES AND FILES                                          }

var
   Disease_Buff : Disease_Record;
   Disease_File : File of Disease_Record;

Chapter,Vec_Length,Number : Integer;
Searchstring : String70;
Codestring :String50;
Feature_Vec : Intarray;
Markarray : Bytearray;
Tier0_Array : Bigarray;
B_Vec, C_Vec, D_Vec : Intarray;
End_of_List, Finished, Aborted : Boolean;
Feature_File :File of String70;
Ch : Char;
```

```
Program Analyse;

{This program determines how often each of the features in the
 file features.ran is used by the disease profiles.  It also pro-
 duces a list of diseases together with the length of the profile
 bytes, as well as a list of manifestations together with the number
 of times each is used.  These are written to files Temp1.txt and
 Temp2.txt which may then be inspected and later removed.      }

{$I Typevar.Pas}
   I,J,Index : Integer;
   Profarray:Array[0..1023] of Integer;
   Feature_Rec : String70;
   FilVar:  Text;
{$I Library.pas}
{External assembler routines}
{$I Asslib.pas}

Begin

    For I := 0 to 1023 do
        Profarray[i] := 0;              {Zeroise whole array}

Assign (Disease_File,'Profile.ran');
Reset (Disease_File);
Assign(FilVar,'Temp1.txt');
Rewrite(FilVar);
For I := 0 to Filesize(Disease_File)-1 do
    Begin

         Read(Disease_File,Disease_buff);
         With Disease_Buff do
         Begin
              Writeln(I:3,' ',Name,'  ',Length(Profile));
              Writeln(FilVar,I:3,'  ',Name,'  ',Length(Profile));
              Decode(Number,Profile,Feature_Vec);
              For J := 1 to Number do
              Begin
                   Index := Feature_Vec[j];
                   Profarray[Index]:= Profarray[Index]+1;
              End;
         End;

    End;

Close (Disease_File);
Close(FilVar);
Assign(Feature_File,'Features.ran');
Reset(Feature_File);
Assign(Filvar,'Temp2.txt');
Rewrite(Filvar);
For J := 0 to 1023 do
Begin
     Read(Feature_File,Feature_rec);
     If Feature_Rec <>'XX' then
```

```
      Begin
         Writeln(J:3,' ',Feature_Rec,'   ',Profarray[j]:2);
         Writeln(Filvar,Feature_Rec,'   ',Profarray[j]);
      End;
End;

Close (Feature_File);
Close (FilVar);
End.
```

APPENDIX B

Assembler Programs

This appendix contains the source code of the assembler programs used in the thesis. The programs were written by means of the text editing option (N-option) of the word-processing application Wordstar.

The source programs were assembled with the Microsoft Macro-assembler version 1-27 and linked with the Microsoft linker. The resulting ".exe" programs were converted to ".com" programs as required by the Turbo Pascal interface. The method of this conversion is the author's own, but will not be detailed here.

```
Title           Convert         ;To change an integer vector into a
                                ;compressed byte string.

; The routine is called by a Pascal program as a near routine defined:

; Convert (n:integer; var vector:array[0..127] of integer;var mystring
;:string[k]); external 'convert.com';


code            segment
                assume cs:code
                org 100h

convert         proc    near
                push bp             ;save registers for safety sake
                mov bp,sp           ;and set base pointer to stack
                push ax             ;pointer
                push bx
                push ds
                push es
                assume ds:code
                jmp init

;A .com program cannot have a separate data segment, hence the
;presence of variables and structures in the code segment over
;which the IP must "jump"

masks           label           byte
                db              8 dup(?)

divisor         db              ?
byte2           db              ?       ;tier2 byte
index2          db              ?       ;index for tier 2 bytes
vec2count       db              ?       ;number of bytes in vec2
vec2            db              128 dup(?)    ;longest possible string

byte1           db              ?       ;tier1 byte etc
index1          db              ?
vec1count       db              ?
vec1            db              16 dup(?)

byte0           db              ?       ;tier0 byte etc
index0          db              ?
vec0count       db              ?
vec0            db              2 dup(?)


init:           mov al,8            ;initialise data bytes indexes
                mov divisor,al      ;counters to 0 and divisor to 8
                sub al,al           ;and masks as follows:
                mov byte2,al        ;     00000001
                mov byte1,al        ;     00000010
                mov byte0,al        ;     00000100
                mov index2,al       ;     00001000
                mov index1,al       ;     00010000
```

```
                mov index0,al       ;     00100000
                mov vec2count,al    ;     01000000
                mov vec1count,al    ;     10000000
                mov vec0count,al

                mov al,10000000b
                mov cx,8
                mov bx,0
setmasks:       rol al,1
                mov masks[bx],al
                inc bx
                loop setmasks

                mov cx,[bp]+12      ;no of integers in vector
                sub bh,bh           ;only bl used in entire prog

                mov si,[bp+8]       ;point source at vector start
                mov es,[bp+10]
loopstart:      mov ax,es:[si]      ;get an integer
                add si,2            ;set pointer to next integer
                mov di,offset byte2     ;process tier 2 byte
                call bytehandler
                jc loopend          ;no other processing required
                mov di,offset byte1     ;process tier 1 byte
                call bytehandler
                jc loopend
                mov di,offset byte0     ;process tier 0 byte
                                        ;with own routine
                div divisor             ;Q=Q/8+R
                cmp al,byte0[1]         ;need for new byte?
                je nonew
                mov index0,al
                mov dl,[di]             ;if so save old one
                mov bl,vec0count
                add bl,3
                mov [bx][di],dl
                inc vec0count
                sub dl,dl               ;and prepare fresh one
                mov byte0,dl
nonew:          mov dl,byte0            ;no new byte required so
                mov bl,ah
                or dl,masks[bx]         ;just set bit
                mov byte0,dl
loopend:        clc                     ;may have been used as flag
                loop loopstart          ;by bytehandler
finalize:       mov bl,vec2count        ;move last three bytes to
                mov dl,byte2            ;to respective vectors
                mov vec2[bx],dl
                inc vec2count

                mov bl,vec1count
                mov dl,byte1
                mov vec1[bx],dl
                inc vec1count
```

```
                mov bl,vec0count
                mov dl,byte0
                mov vec0[bx],dl         ;no need to increment counter
                                        ;as both tier 0 bytes stored.
                cmp bl,0                ;save all-zero byte0 prn
                jne move
                sub dl,dl
                mov vec0[1],dl

move:           mov si,offset vec0      ;mov bytes to string starting
                mov di,[bp+4]           ;with byte0.
                mov es,[bp+6]
                cld
                mov cl,2                ;the 2 obligatory tier 0 bytes,
                add cl,vec1count        ;then add others to obtain
                add cl,vec2count        ;string length which is saved
                mov es:[di],cl          ;at first byte of string addr
                inc di                  ;string starts after length
                movsb
                movsb
                mov si,offset vec1
                mov cl,vec1count
                sub ch,ch
                rep movsb
                mov si,offset vec2
                mov cl,vec2count
                rep movsb

                pop es                  ;restore stack and registers
                pop ds
                pop bx                  ;for return from subroutine
                pop ax
                pop bp

                ret 10                  ;discard parameters
convert         endp

bytehandler     proc    near

                div divisor
                mov dl,[di]
                cmp dl,0                ;deal with initial all-zero
                je isq                  ;byte
                cmp al,[di]+1           ;compare quotient with index
                jne newbyte             ;if changed [redacted] make fresh byte
                mov bl,ah
                or dl,masks[bx]         ;set appropriate bit from re-
                mov [di],dl             ;remainder
                stc                     ;flag to bypass other bytes
                jmp exit
newbyte:        mov bl,[di]+2           ;compute place for processed
                mov [di+3][bx],dl       ;byte to go and put it there
                inc byte ptr[di+2]      ;and increment counter
                sub dl,dl               ;and replace with zero byte
                mov [di],dl
```

```
isq:            mov dl,[di]
                mov bl,ah
                or dl,masks[bx]
                mov [di],dl
                mov [di+1],al           ;replace index with quotient
                                        ;get Q into al and clear ah
                sub ah,ah
exit:           ret
bytehandler     endp
code            ends
                end
```

```
Title           Decode

;This program does the opposite to program convert in that it acts on
;a byte string which had been bit-mapped and compressed and expands it
;into a vector of integers that can serve as indices to a lookup table
;of, for example, X-ray or clinical features of disease entities.
;The subprogram is called from a Pascal program where it it is dec-
;lared as an external procedure decode(var n:integer;var string:astring;
;vec:anarray);
;
; F van Greunen  Sept 1985
;

code            segment
                assume cs:code
                org 100h
decode          proc near
                push bp                 ;usual start stacking regs
                mov bp,sp
                push ax
                push bx
                push ds
                push es
                assume ds:code
                jmp init                ;jump over variables

eight           db      ?
index0          db      ?
index1          db      ?
index2          dw      ?
numcount        dw      ?
pointer1        dd      ?               ;pointer to tier 1 bytes
pointer2        dd      ?               ;pointer to tier 2 bytes
vecpointer      dd      ?

init:           sub ax,ax               ;zeroize variables
                mov index0,al
                mov index1,al
                mov index2,ax
                mov numcount,ax
                mov al,8
                mov eight,al
                mov si,[bp+8]           ;string address from stack
                mov es,[bp+10]
                inc si                  ;point to tier 0 bytes
                mov ax,es:[si]          ;move both to ax
                sub bx,bx               ;count set bits in tier o bytes
                mov cx,16               ;by right rotation and counting
countbegin:     ror ax,1                ;set carries into bh used as
                jnc countend            ;temporary counter
                inc bx                  ;set bit counted
countend:       loop countbegin


                inc si                  ;load pointers with values just
```

```
                mov word ptr pointer1,si ;before tier 1 and 2 bytes
                mov word ptr pointer1[2],es  ;respectively so that first
                add si,bx                    ;increments will have
                mov word ptr pointer2,si     ;pointers at their starts
                mov word ptr pointer2[2],es
                mov di,[bp+4]                ;initialize vector pointer
                mov es,[bp+6]
                mov wor ptr vecpointer,di
                mov word ptr vecpointer[2],es
                mov cx,bx                    ;get set-bit count into cx
                                             ;for loop

mov0:           shr ax,1                    ;start shifting tier 0 bytes
                jc tier1
                inc index0
                jmp mov0

tier1:          push ax                     ;stack t-0 bytes and process
                mov al,index0               ;t-1 bytes
                mul eight                   ;index1 =index0*8
                mov index1,al
                inc index0
                inc word ptr pointer1
                push cx                     ;t-0 counter
                mov cx,8                    ;t-1 counter
                les si,pointer1             ;fetch a t-1 byte
                mov al,es:[si]


mov1:           shr al,1                    ;start processing it
                jc tier2                    ;bit set
                inc index1                  ;bit not set
                jmp mov1end

tier2:          push ax                     ;calculate index2
                mov al,index1               ;= index1 * 8
                sub ah,ah
                mul eight
                mov index2,ax
                inc index1
                inc word ptr pointer2
                push cx                     ;t-1 counter
                mov cx,8                    ;t-2 counter
                les si,pointer2             ;get t-2 byte
                mov al,es:[si]
                les di,vecpointer           ;point to integer vector

mov2:           shr al,1                    ;process t-2 byte
                jnc movon                   ;bit not set
                mov bx,index2               ;bit set and number corres-
                mov es:[di],bx              ;ponding to index2 must be put
                inc numcount                ;into vector using only availa-
                inc di                      ;ble reg bx. point di to next
                inc di                      ;location in vector
                mov word ptr vecpointer,di
```

```
movon:          inc index2
                loop mov2

                pop cx
                pop ax
movlend:        loop mov1

                pop cx
                pop ax
                loop mov0


finalize:       mov di,[bp+12]          ;put number count in var n
                mov es,[bp+14]
                mov ax,numcount
                mov es:[di],ax

exit:           pop es                  ;restore registers for return
                pop ds
                pop bx
                pop ax
                pop bp
                ret 12

decode          endp
code            ends
                end
```

```
Title Compare.asm

; This program compares a vector avec containing m integers with a
; vector bvec containing n integers.  It produces a vector cvec contai-
; ning p integers which are common to avec and bvec. Thus cvec will
; contain the record numbers of those features which the search profile
; has in common with those of a known one in the knowledge base.
;
;               F van Greunen  Sep 1986


cseg            segment
                assume cs:cseg
                org 100h                ;com type prog for Turbo Pascal

compare         proc
                jmp     start           ;jump over the local variables

acount          dw      ?
bcount          dw      ?
ccount          dw      ?
apoint          dd      ?
bpoint          dd      ?
start:          push    bp
                mov     bp,sp
                push    ax
                push    bx

                lds     si,[bp+20]      ;number of features (m) in avec
                lodsw
                mov     cs:acount,ax

                lds     si,[bp+12]      ;number of features (n) in bvec
                lodsw
                mov     cs:bcount,ax

                les     di,[bp+8]       ;addr of cvec
                lds     si,[bp+24]      ;addr of avec to pointer
                mov     word ptr cs:apoint,si
                mov     ax,ds
                mov     word ptr cs:apoint+2,ax
                lds     si,[bp+16]      ;same for bvec
                mov     word ptr cs:bpoint,si
                mov     ax,ds
                mov     word ptr cs:bpoint+2,ax

begin:          lds     si,cs:bpoint
                lodsw
                mov     bx,ax           ;number from bvec into bx

entry1:         lds     si,cs:apoint
                lodsw                   ;number from avec into ax

comp:           cmp     ax,bx
                je      equal
```

```
                jg      agreater

bgreater:       dec     cs:acount
                jz      exit                ;no more in avec
                add     word ptr cs:apoint,2    ;adjust apointer
                jmp     entry1              ;next round

agreater:       dec     cs:bcount
                jz      exit                ;no more in bvec
                add     word ptr cs:bpoint,2    ;adjust bpointer
                push    ax                  ;conserve ax
                lds     si,cs:bpoint        ;next from bvec
                lodsw
                mov     bx,ax
                pop     ax
                jmp     comp

equal:          stosw                       ;store the common feature
                inc     cs:ccount           ;in cvec
                dec     cs:acount           ;decrement both avec and bvec
                jz      exit                ;and exit if one of them is 0
                dec     cs:bcount
                jz      exit
                add     word ptr cs:apoint,2
                add     word ptr cs:bpoint,2
                jmp     begin

exit:           mov     ax,cs:ccount        ;store p (number of features
                les     di,[bp+4]           ;in cvec in proper place )
                stosw

                pop     bx      ;restore environment
                pop     ax
                pop     bp

                ret     24

compare         endp
cseg            ends
end
```

```
Title  Scan.asm

; This program scans a longer string for the occurrence of a shorter
; string.  If found a boolean parameter is set to true, otherwise
; false.  The procedure is called by a Turbo pascal program as:
; Procedure Scan (var longstring:string70;var shortstring:string70;
;                 var Boole: Boolean); external 'Scan.com';

;       F van Greunen   Oct 1986

include mymacs.bib

cseg            segment
                assume  cs:cseg
                org     100h            ;.com program

scan            proc    near
                jmp     start           ;jump over local variables


index           db      ?               ;no.of possible substrings
shortstr_length db      ?
flag            db      ?
keep_di         dw      ?

start:          push    bp              ;set up base for parameters
                mov     bp,sp           ;and save environment
                push    ax
                push    bx
                push    cx
                push    dx
                push    ds
                push    es
                lds     si,[bp+12]      ;start of long string
                mov     al,[si]         ;str len forms basis for index
                mov     cs:index,al
                les     di,[bp+8]       ;pointer to short string
                mov     keep_di,di      ;store pointer safely so as to
                inc     keep_di         ;pnt to first char of short str
                mov     al,es:[di]
                mov     cs:shortstr_length,al
                mov     cl,cs:shortstr_length   ;set up cx for loop
                xor     ch,ch

                mov     dl,cs:shortstr_length   ;yardstick for comparin

loop1:          dec     cs:index        ;adjust index to L-l+1
                loop    loop1           ;L=length of long string
                inc     cs:index        ;l=length of short string
                cmp     cs:index,0      ;error if short string
                jle     not_found       ;longer than long string!
                mov     cs:flag,0       ;initialize flag, when
                                        ;set it indicates a match

short_next:     inc     di              ;next char of short str
```

```
long_next:      inc     si                  ;ditto for long string
compare:        mov     al, [si]
                cmp     al,'a'              ;change long str chars to
                jl      cont1               ;upper case if necessary
                cmp     al,'z'
                ja      cont1
                sub     al,32
cont1:          mov     bl,es:[di]          ;ditto for short str chars
                cmp     bl,'a'
                jl      cont2
                cmp     bl,'z'
                jg      cont2
                sub     bl,32
cont2:          cmp     al,bl
                je      match
                dec     cs:index            ;no match, one char less
                jz      not_found           ;unsuccessful end of scan
                cmp     cs:flag,0           ;look for matched prede-
                jz      long_next           ;cessor
                mov     di,keep_di          ;return to beginning of
                mov     dl,cs:shortstr_length   ;short str and reset fl

                mov     cs:flag,0
                jmp     compare             ;continue scan

match:          mov     cs:flag,1
                dec     dl
                jnz     short_next          ;compare next char of
                                            ;short str
found:          mov     al,1                ;set boolean parameter
                jmp     save_boole

not_found:      mov     al,0                ;reset boolean parameter
save_boole:     les     di,[bp+4]
                stosb

return:         pop     es                  ;restore environment
                pop     ds
                pop     dx
                pop     cx
                pop     bx
                pop     ax
                pop     bp
                ret     12

scan            endp
cseg            ends
end
```

```
Title Search.asm

; This program has a search routine which is called by the main Pascal
; program as follows:

; Search (Position,Number,Array,Codestring)

; The program starts at Position which is 0 originally and compares the
; tier 0 bytes of the unknown condition with those from the profiles
;file and contained in Tier0array.

; As soon as a word (2 bytes) is found with one or more set bits in
; common with the search string, a new Position is returned to the main
; program.  Should none be found, -1 is returned.

;               F van Greunen Oct 1986

cseg    segment
        assume  cs:cseg
        org     100h                    ;com type prog for Turbo Pascal

search  proc    near

        jmp     start
index   dw      ?
start:  push    bp                      ;save the environment
        mov     bp,sp
        push    ax
        push    bx
        push    cx
        push    ds
        push    es

        les     di,[bp+16]              ;address of Position
        mov     ax,es:[di]              ;get it to index
        mov     cs:index,ax

        lds     si,[bp+4]               ;address of codestring
        inc     si                      ;jump over length byte
        lodsw
        mov     bx,ax                   ;get Tier 0 bytes into bx

        lds     si,[bp+12]              ;address of Number
        mov     cx,[si]
        dec     cx                      ;Max position one less than no.
        mov     ax,cs:index             ;Position into ax
        push    bx                      ;Multiply by three as
        mov     bx,3                    ;there are 3 bytes per string
        imul    bx                      ;temporarily using bx as multi-
        pop     bx                      ;plier
        lds     si,[bp+8]               ;beginning of array
        add     si,ax                   ;offset into array

loop1:  inc     si                      ;past length byte
        lodsw                           ;t0 bytes from array into ax
```

```
        test    ax,bx                   ;compare
        jne     end_off                 ;some bits in common
        inc     cs:index                ;no bits in common thus next
        cmp     cs:index,cx             ;end of array?
        jl      loop1                   ;no

end_off:        mov     ax,cs:index
                stosw                   ;es,di already initialized, put
                                        ;new value into position

                pop     es
                pop     ds
                pop     cx
                pop     bx
                pop     ax
                pop     bp

                ret     16

search          endp
cseg            ends
end
```

```
Title     Passwd.asm

;Simple password program.  No comments for obvious reasons

;         F van Greunen              Nov 1986

include mymacs.bib                   ;library of useful macros

cseg              segment
                  assume  cs:cseg
                  org     100h

password          proc
                  push    bp

                  mov     bp,sp
                  push    di
                  xor     bx,bx
                  mov     cx,7
                  xor     ah,ah
loop1:            read_kbd_no_echo
                  and     ax,00ffh
                  add     bx,ax
                  loop    loop1
                  cmp     bx,272h
                  jne     reject
                  mov     al,1
                  jmp     store_boole
reject:           mov     al,0
store_boole:      les     di,[bp+4]
                  stosb
                  pop     di
                  pop     bp
                  ret     4
password          endp
cseg              ends
                  end
```

The following is the listing of a batch file "autoexec.bat" which will effect automatic execution of the menu program, provided that it is present on the device from which the computer is normally booted up. This could be a flexible disk in drive A: or the root directory of the fixed disk drive C:

```
echo off
rem   assume programs reside in directory 'skeletal' on
rem   the hard disk which is drive c:
c:
chdir skeletal
menu
```

APPENDIX C

This appendix contains the list of skeletal disorders presently supported by the knowledge base. This is an alphabetical list of **NAMES** and **SYNONYMS**, hence there are two entries for several of the disorders.

Aarskog Syndrome
Achondrogenesis Type 1
Achondrogenesis Type 2
Achondroplasia
Acro-osteolysis Carpotarsal Type
Acro-osteolysis Francois type
Acro-osteolysis Hajdu-Cheney Type
Acro-osteolysis Phalangeal type
Acro-osteolysis Torg type
Acrocehalopolysyndactyly
Acrocephalosyndactyly
Acrodysostosis
Acromesomelic Dysplasia
Aicardi Syndrome
Albers-Schonberg Disease
Alkaptonuria
Amyoplasia
Angio-osteohypertrophy
Apert Syndrome
Arteriohepatic Dysplasia
Arthro-ophthalmopathy
Arthrogryposis multiplex congenita
Asphyxiating Thoracic Dysplasia
Atelosteogenesis
Basal Cell Nevus Carcinoma Syndrome
Beckwith-Wiedemann Syndrome
Black-Sulzberger Syndrome
Blount Disease
Brachydactyly Syndromes
Brachyolmia Fontaine type
Brachyolmia Hobaek type
Brachyolmia dominant type
Brachyolmia recessive type
Caffey Disease
Campomelic Dysplasia
Camurati-Engelmann Disease
Carpenter Syndrome
Cartilage-Hair Hypoplasia
Catel-Schwartz-Jampel Syndrome
Cerebral Gigantism
Cerebro-costo-mandibular Syndrome
Cerebro-hepato-renal Syndrome
Cerebro-oculo-facio-skeletal syndrome
Cervico-oculo-acoustic Syndrome
Cheirolumbar Dysostosis
Cherubism

Chondrodysplasia Punctata Dominant Form
Chondrodysplasia Punctata Rhizomelic Form
Chondrodysplasia punctata Recessive type
Chondroectodermal Dysplasia
Cleidocranial Dysplasia
Cockayne Syndrome
Coffin-Lowry Syndrome
Coffin-Siris Syndrome
Conradi-Hunermann Syndrome
Cornelia de Lange Syndrome
Cranio-carpo-tarsal (whistling face) syndrome
Craniodiaphyseal Dysplasia
Craniofacial Dysostosis
Craniometaphyseal Dysplasia
Cretinism
Crouzon Syndrome
Cutis laxa
De Barsey Syndrome
De La Chapelle Dysplasia
Dentinogenesis Imperfecta
Diaphyseal Aclasis
Diaphyseal Dysplasia
Diastrophic Dysplasia
Disappearing bones
Distal Osteosclerosis
Distichiasis-Lymphedema Syndrome
Down Syndrome
Dyggve-Melchior-Clausen Dysplasia
Dyschondrosteosis
Dysosteosclerosis
Dysplasia Epipysealis Hemimelica
Dyssegmental Dysplasia
Ehlers-Danlos Syndrome
Ellis-van Creveld Syndrome
Enchondromatosis
Enchondromatosis with Hemangiomata
Enchondromatosis with irregular vertebral lesions
Endosteal Hyperostosis Van Buchem form
Endosteal Hyperostosis Worth form
Epiloia Syndrome
FG Syndrome
Faciogenital Dysplasia
Familial Undifferentiated Hypermobility
Fanconi Pancytopenia Syndrome
Femoral Dysgenesis Syndrome with Coxa Vara
Femoral Facial Syndrome

Fetal Alcohol Syndrome
Fibrochondrogenesis
Fibrodysplasia Ossificans Progressiva
Fibrous Dysplasia
Fibrous Dysplasia Dermal pigmentation Precocious P
Fibrous Dysplasia of the Jaws
Focal Dermal Hypoplasia
Freeman-Sheldon Syndrome
Frontometaphyseal Dysplasia
Fucosidosis
Fuhrmann Dysplasia
GMI Gangliosidosis
Galt Lipomucopolysaccharidosis
Gardner Syndrome
Gaucher Disease
Generalized enchondromatosis
Geroderma Osteodysplastica
Giedion-Langer Syndrome
Goldenhar Syndrome
Goltz Syndrome
Gorham Osteolysis
Hallermann-Streiff Syndrome
Holt-Oram Syndrome
Humerospinal Dysostosis
Hunter Disease
Hurler Syndrome
Hyperphosphatasia
Hypochondroplasia
Hypophosphatasia
Hypophosphatemia: Vitamin D-Resistant Rickets
Hypopituitarism
Hypothyroidism
I-cell Disease
Incontinentia Pigmenti
Infantile Cortical Hyperostosis
Jaffe-Lichtenstein Syndrome
Jeune Syndrome
Juvenile Idiopathic Osteoporosis
Kenny-Caffey Syndrome
Klinefelter Syndrome
Klippel-Feil Syndrome
Klippel-Trenaunay-Weber Syndrome
Kniest Syndrome
Larsen Syndrome
Leri-Weill Disease
MCD with Swiss type agammaglobulinemia

MURCS Association
Macrodactyly simplex congenita
Maffucci Syndrome
Mandibulofacial Dysostosis
Mannosidosis
Marfan Syndrome
Maroteaux-Lamy Disease
Marshall Syndrome
McCune-Albright Syndrome
Melnick-Needles Syndrome
Melorheostosis
Menkes Kinky Hair Syndrome
Mesomelic Dysplasia Hypoplastic Tibia and Radius
Mesomelic Dysplasia Langer Type
Mesomelic Dysplasia Nievergelt type
Mesomelic Dysplasia Reinhardt-Pfeiffer Type
Mesomelic dwarfism of Campailla and Martinelli
Metachondromatosis
Metaphyseal Chondrodysplasia (Pancreatic and Marro
Metaphyseal Chondrodysplasia Jansen Type
Metaphyseal Chondrodysplasia McKusick type
Metaphyseal Chondrodysplasia Schmid type
Metaphyseal Chondrodysplasia Vaandrager-Pena type
Metaphyseal Chondrodysplasia with Thymolymphopenia
Metaphyseal Dysplasia
Metatropic Dysplasia
Meyer Dysplasia
Micrognathic Dwarfism
Mohr Syndrome
Mongolism
Morquio Disease
Mucolipidosis I
Mucolipidosis II
Mucolipidosis III
Mucopolysaccharidosis I-H
Mucopolysaccharidosis I-S
Mucopolysaccharidosis II
Mucopolysaccharidosis III
Mucopolysaccharidosis IV
Mucopolysaccharidosis VI
Mucopolysaccharidosis VII
Multiple Cartilagenous Exostoses
Multiple Epiphyseal Dysplasia
Multiple Pterygium Syndrome
Multiple Synostoses Syndrome
Myotonic Chondrodysplasia

Nail-Patella Syndrome
Naumoff (Lethal Thoracic Dysplasia) Syndrome
Neu-Laxova Syndrome
Neurofibromatosis
Nievergelt Syndrome
Ochronosis
Oculoauriculo-vertebral Dysplasia
Oculodento-digital Dysplasia
Oculodento-osseous Dysplasia
Oculomandibulo-facial Syndrome
Ollier Disease
Opitz-Kaveggia syndrome
Oro-facial-digital Syndrome Type 1
Oro-facial-digital Syndrome type 2
Osteo-onychodysostosis
Osteodysplasty
Osteoectasia with Hyperphosphatasia
Osteofibrous Dysplasia of the Tibia and Fibula
Osteogenesis Imperfecta Congenita
Osteogenesis Imperfecta Tarda
Osteoglophonic Dysplasia
Osteomesopycnosis
Osteopathia Striata
Osteopathia Striata with Cranial Sclerosis
Osteopetrosis with Delayed Manifestations
Osteopetrosis with Precocious Manifestations
Osteopoikilosis
Osteoporosis-Pseudoglioma Syndrome
Oto-palato-digital Syndrome
Pachydermoperiostosis
Papillon-Leage Syndrome
Parastremmatic Dysplasia
Parkes-Weber Syndrome
Pena-Shokeir Syndrome
Peripheral Osteosclerosis
Perthes Disease
Phocomelia
Pituitary Dwarfism
Popliteal Pterygium Syndrome
Progeria
Pseudo-Hurler disease; Landing disease
Pseudo-Hurler polydystrophy
Pseudoachondroplasia
Pseudohypoparathyroidism
Pycnodysostosis
Pyle Disease

Radial Aplasia-Thrombocytopenia Syndrome
Rickets
Roberts Pseudothalidomide Syndrome
Robinow Syndrome
Rothmund-Thomson Syndrome
Rubinstein-Taybi Syndrome
Russell-Silver Syndrome
SPONASTRIME syndrome
Saldino-Noonan Syndrome
Sanfilippo Disease
Scheie Disease
Sclerosteosis
Seckel Syndrome
Short Rib Syndrome Type 2
Short Rib Syndrome type 1
Short Rib Syndrome type 3
Short Rib-Polydactyly Syndrome Majewski type
Shwachman Syndrome
Silver-Russell Syndrome
Singleton-Merten Syndrome
Smith-Lemli-Opitz Syndrome
Smith-McCort Syndrome
Smith-McCort Syndrome
Sotos Syndrome
Spondylo-epi-metaphyseal Dysplasia
Spondylo-epi-metaphyseal Dysplasia Joint Laxity
Spondylo-epiphyseal Dysplasia Congenita
Spondylo-epiphyseal Dysplasia Tarda
Spondylo-epiphyseal Dysplasia with arthropathy
Spondylochondrodysplasia
Spondylocostal Dysostosis
Spondyloenchondrodysplasia
Spondylometaphyseal Dysplasia - Heterogeneous
Spondylometaphyseal Dysplasia Brazilian type
Spondylometaphyseal Dysplasia Kozlowski type
Spondylometaphyseal Dysplasia Schmidt type
Spondyloperipheral Dysplasia
Stanescu Syndrome
Stickler Syndrome
Sturge-Weber Syndrome
Taybi-Linder Syndrome
Thanatophoric Dysplasia
Treacher Collins Syndrome
Trichorhinophalangeal Acrodysplasia with Exostoses
Tuberous Sclerosis
Tubular Stenosis

Tumoral Calcinosis
Turner Syndrome
Upper limb-CVS syndrome
VATER Association
Von Recklinghausen Disease
Voorhoeve Disease
Weaver Syndrome
Weismann-Netter-Stuhl Syndrome
Weissenbacher-Zweymuller Syndrome
Wildervanck Syndrome
Zellweger Syndrome

APPENDIX D

This appendix contains the list of manifestations currently pre sent in the file "Features.Ran". It represents the present state of affairs, but may be altered by the addition of new stigmata where vacancies exist as well as at the ends of the various sec tions where ample space has been provided.

Non-skeletal Features

Small stature/Dwarfism
Face - Unusual/Characteristic
Face - Coarse features
Mental retardation
Lethal chondrodysplasia
Eyes - Coloboma
Eyes - Microphthalmia
Eyes - Myopia/Hypermyopia
Eyes - Corneal clouding/opacity/cataract
Eyes - Non-specific defects
Eyes - Pseudoglioma
Eyes - Proptosis/Bulging
Eyes - Cherry-red macular spots
Ears - Abnormal external ears
Ears - Deafness/impaired hearing
Vacant
Vacant
Mouth - Cleft lip
Mouth - Cleft palate
Mouth - Macrostomia
Mouth - Microstomia
Mouth - Macroglossia
Mouth - Gingival hypertrophy
Vacant
Nose - Abnormal shape
Nose - Depressed nasal bridge
Nose - Broad nasal bridge
Philtrum - abnormal
Vacant
Lymphedema
Cardiac defect
Arterial abnormalities
Venous abnormalities
Hemangiomata - widespread
Arterio-venous fistula
Pulmonary defect
Abnormal genitalia/Urogenital defect/renal disease
Gastrointestinal defect
Hernia
Abnormal skin/pigmentation
Nail disorder
Abnormal hair

Hirsutism
Alopecia
Neoplasm (Benign or malignant)
Metabolic abnormality - heteroglycan
Metabolic abnormality - lipid
Metabolic abnormality - amino acid
Metabolic abnormality - calcium
Metabolic abnormality - phosphate
Hematopoetic disorder
Immunologic disorder
Hepatic disease/enlargement
Splenomegaly
Lymphadenopathy
Sexual development - Retarded
Sexual development - Precocious
Gigantism/Excessive growth
Asymmetry
CNS defect (excluding cranial nerves)
Cranial nerve defect
Webbed neck/axillae/elbows/knees (pterygia)
Subcutaneous calcification/calcified spheroids
Soft tissue - Ossification
Soft tissue - Hyperlaxity
Soft tissue - Ulceration
Soft tissue - Hypertrophy
Muscle - Wasting/reduced mass
Muscle - Myotonia
Muscle - Hypotonia

Skull X-ray Features

Macrocephaly
Microcephaly
Dyscephaly/Trigonocephaly
Brachycephaly
Dolicocephaly
Hypertelorism
Turricephaly
Calvaria - Osteomata
Calvaria - Poor ossification
Calvaria - Wide sutures
Calvaria - Sclerosis
Calvaria - Hyperostosis
Calvaria - Thickened
Calvaria - Coarse trabecular appearance
Calvaria - Late open fontanelles
Calvaria - Large/wide fontanelles
Calvaria - Small fontanelles
Frontal sinuses - Large
Frontal bones - Prominent/bossed
Supraorbital ridges - Prominent
Vacant
Vacant Large or wide fontanelles
Vacant Small fontanelles
Cranial sutures - Wormian bones
Cranial sutures - Premature fusion
Sella turcica - J-shaped
Craniostenosis
Maxilla/Midface - hypoplasia
Maxilla - prognathism
Mandible - cysts
Mandible - hypoplasia/Micrognathia
Mandible - prognathism
Vacant
Vacant
Vacant
Mandible - Osteomata
Mandible - Hyperostosis
Mandible - Radiolucent defects
Mandible - Fibrous dysplasia
Mandible - Wide angle
Teeth - Irregular
Teeth - Delayed eruption

Teeth - Precocious loss
Teeth - Dysplasia of enamel
Teeth - Impaction
Teeth - Absence of pulp chambers
Skull base - Short
Skull base - Small foramen magnum
Skull base - Basilar impression
Skull base - Sclerosis
Skull base - Hyperostosis
Vacant
Vacant
Intracranial calcification - Basal ganglia
Intracranial calcification - 'Double contour'of the meninges
Intracranial calcification - Periventricular
Intracranial calcification - Falx cerebri
Intracranial calcification - Generalized or non-specific
Vacant
Vacant
Vacant

Thorax X-Ray Features

Thorax - Narrow
Thorax - Short
Thorax - Broad
Pectus carinatum
Pectus excavatum
Vacant
Vacant
Vacant
Clavicles - Straight
Clavicles - Hypoplastic
Clavicles - Absent
Clavicles - High
Clavicles - Short
Clavicles - Bowed
Clavicles - Wide
Clavicles - Hyperostosis
Clavicles - Gross expansion
Vacant
Vacant
Vacant
Scapula - Abnormal position
Scapula - Absent or hypoplastic
Scapula - Irregular contours
Scapula - Plump
Scapula - Hyperostosis
Scapula - Exostoses
Vacant
Ribs - Thin/slender
Ribs - Short
Ribs - Wide/oar-shaped
Ribs - Malsegmentation
Ribs - Irregular contours
Ribs - Fractures
Ribs - Gaps
Ribs - Hypoplastic/absent
Ribs - Non-specific dysplasia
Ribs - Sclerosis
Ribs - Splayed anterior ends
Ribs - Cupped anterior ends
Ribs - Hyperostosis
Ribs - Exostoses
Vacant

Spine X-ray Features

Spine - Degenerative changes (non-specific)
Spine - Generalized structural abnormalities
Spine - Generalized Scoliosis/ Kyphoscoliosis
Vacant
Vacant
Vacant
Cervical spine - Kyphosis
Cervical spine - Lordosis
Cervical spine - Hypoplastic vertebrae
Cervical spine - Fused vertebrae
Odontoid Process - Hypoplastic
Vacant
Vacant
Vacant
Thoracic spine - Hypoplastic vertebrae
Thoracic spine - Accentuated kyphosis/gibbus
Thoracic spine - Fused vertebrae
Vacant
Vacant
Vacant
Lumbar spine - Accentuated lordosis
Lumbar spine - Incomplete ossification vertebrae
Lumbar spine - Fused vertebrae
Lumbar spine - Canal narrowed (stenosed)
Lumbar spine - Interpedicular narrowing
Lumbar spine - Spina bifida
Vacant
Vacant
Vertebral bodies - Poor or absent ossification
Vertebral bodies - Calcific stippling
Vertebral bodies - Short
Vertebral bodies - Tall
Vertebral bodies - Ovoid
Vertebral bodies - Hookshaped
Vertebral bodies - Anterior wedging
Vertebral bodies - Posterior wedging
Vertebral bodies - Concave posterior surfaces
Vertebral bodies - Concave anterior surfaces
Vertebral bodies - Irregular endplates
Vertebral bodies - Central indentations
Vertebral bodies - Anterior ossification defects
Vertebral bodies - Antero-superior ossification defects

Vertebral bodies - Antero-inferior ossification defects
Vertebral bodies - Butterfly appearance
Vertebral bodies - H-shaped in A-P projection
Vertebral bodies - Osteopenia
Vertebral bodies - Pear-shaped
Vertebral bodies - Rounded lower and upper margins
Vertebral bodies - Coronal fissure/clefts
Vertebral bodies - Clefts other than coronal
Vertebral bodies - Distortion/dysplasia (generalised)
Vertebral bodies - Dorsal hump
Vertebral bodies - Increased A-P diameter
Vertebral bodies - Decreased A-P diameter
Vacant
Vertebrae - Malsegmentation (localized)
Vertebrae - Thinned or short pedicles
Vertebrae - Widened spinal canal
Vertebrae - Hemivertebrae
Vertebrae - Supernumerary
Vacant
Vertebrae - Enlarged interpedicular spaces
Vertebrae - Increased disc spaces
Vertebrae - Platyspondyly
Vertebrae - Anisospondyly
Vertebrae - Sclerosis of end-plates
Vertebrae - Sclerosis ('rugger jersey' spine)
Vertebrae - Striations
Vacant
Vacant

Pelvis X-ray Features

Pelvis - Small
Pelvis - Irregular calcification
Pelvis - Retarded bone age
Pelvis - Poor ossification
Vacant - Synchondrosis pubis
Pelvis - Failure of bone fusion
Pelvis - Multiple small sclerotic foci
Vacant
Vacant
Acetabula - Low angles
Acetabula - Steep
Acetabula - Horizontal
Acetabula - Irregular
Acetabula - Lateral enthesopathy
Acetabula - Triradiate
Supra-acetabular constriction
Acetabula - Deficient ossification of roof
Acetabula - Wide roof
Acetabula - Shallow roof
Ilia - Absent/hypoplastic
Ilia - Sclerosis
Ilia - Linear striations
Ilia - Flared
Ilia - Squaring
Ilia - Horns
Ilia - Reduced indices
Ilia - Crenated crests
Ilia - Small/shortened sacro-iliac notches
Ilia - Narrow sacro-iliac notches
Ilia - Exostoses
Vacant
Pubis - Synchondrosis
Pubis - Small
Pubis - Retarded ossification
Pubis - Narrow bones
Pubis - Irregular
Ischia - Irregular
Vacant
Vacant
Vacant

Tubular Bone X-ray Features

Long bones - Non-specific dysplasia
Long bones - Exostoses
Long bones - Enchondromata
Long bones - Generalised bowing
Long bones - Calcific stippling
Long bones - Lytic lesions
Long bones - Osteolysis
Long bones - Excessive new bone formation
Long bones - Short shafts
Long bones - Thin shafts (Hypertubulation)
Long bones - Wide shafts (Hypotubulation)
Long bones - Submetaphyseal undermodelling
Long bones - Overconstriction
Long bones - Fractures
Long bones - Osteopenia of whole shaft
Long bones - Submetaphyseal osteopenia
Long bones - Diffuse sclerosis of all
Long bones - Diffuse sclerosis of some
Long bones - Diffuse sclerosis of parts
Long bones - Striation
Long bones - Thin cortices
Long bones - Irregular widening of cortices
Long bones - Coarse trabeculation
Long bones - Radiolucent cysts
Long bones - Irregular contours
Long bones - Excessive length compared to width
Long bones - Unilateral irregular linear areas of increased de
Long bones - Asymmetric shortening
Long bones - Rhomboid shaped
Long bones - Diaphyseal cortical endosteal hyperostosis
Long bones - Markedly expanded metaphyses with 'endobones'
Long bones - Multiple small round sclerotic foci at ends
Long bones - Periosteal thickening
Long bones - Triradiate ends
Long bones - Cortical thickening
Long bones - Narrow medullary cavities
Vacant
Vacant
Vacant
Vacant
Long bones - Irregular ossification of epiphyses
Long bones - Premature epiphyseal fusion

Long bones - Generalized epiphyseal dysplasia
Long bones - Punctate epiphyses during infancy
Long bones - Delayed epiphyseal fusion
Vacant
Long bones - Expanded metaphyses
Long bones - Metaphyseal spurs
Long bones - Metaphyseal flaring
Long bones - Rounded metaphyseal borders
Long bones - Flared osteoporotic metaphyses
Long bones - Generalized metaphyseal dysplasia
Long bones - Metaphyseal cysts
Long bones - Irregular ossification of metaphyses
Long bones - Metaphyseal striations
Vacant
Vacant
Humerus - Aplasia/Severe hypoplasia
Humerus - Proximal hypoplasia
Humerus - Distal hypoplasia
Humerus - Bowing
Humerus - Severe distortion
Vacant
Vacant
Radius - Aplasia/Severe hypoplasia
Radius - Proximal hypoplasia
Radius - Distal hypoplasia
Radius - Radial ray hypoplasia
Radius - Bowing
Radius - Severe distortion
Radius - Hyperostosis
Radius - Short
Ulna - Aplasia/Severe hypoplasia
Ulna - Proximal hypoplasia
Ulna - Distal hypoplasia
Ulna - Bowing
Ulna - Severe distortion
Ulna - Triangular configuration
Ulna - Hyperostosis
Ulna - Short
Vacant
Femur - Aplasia/Severe hypoplasia
Femur - Proximal hypoplasia
Femur - Distal hypoplasia
Femur - Bowing
Femur - Severe distortion
Femur - Cyst-like lesions proximal
Femur - 'Telephone receiver' Configuration

Femur - Marginal spur
Vacant
Vacant
Vacant
Tibia - Aplasia/Severe hypoplasia
Tibia - Proximal hypoplasia
Tibia - Distal hypoplasia
Tibia - Bowing
Tibia - Severe distortion
Tibia - Pseudoarthrosis
Tibia - Wavy configuration
Tibia - Irregular intracortical osteolytic and sclerotic changes
Tibia - Hyperostosis
Tibia - Short
Vacant
Fibula - Aplasia/Severe hypoplasia
Fibula - Proximal hypoplasia
Fibula - Distal hypoplasia
Fibula - Bowing
Fibula - Severe distortion
Fibula - Triangular configuration
Fibula - Disproportionately long
Fibula - Hyperostosis
Fibula - Short

Joints X-ray Features

Joints - Loose bodies
Joints - Degenerative osteo-arthropathy
Joints - Multiple bony ankyloses
Joints - Large localized calcified lesions in overlying soft tissue
Joints - Generalized hypermobility
Joints - Generalized hypomobility
Joints - Generalized dislocation/subluxation
Joints - Generalized contractures
Vacant
Vacant
Vacant
Shoulder - Delayed ossification of epiphyses
Shoulder - Large epiphyses
Shoulder - Small epiphyses
Shoulder - Flat epiphyses
Shoulder - Irregular epiphyses
Shoulder - Fragmented epiphyses
Shoulder - Sclerotic epiphyses
Shoulder - Cone-shaped epiphyses
Shoulder - Splayed metaphyses
Shoulder - Cupped metaphyses
Shoulder - Irregular margins of metaphyses
Shoulder - Sclerotic margins of metaphyses
Shoulder - Circumscribed ossification defects of metaphyses
Shoulder - Metaphyseal spurs
Shoulder - Premature fusion of metaphyses
Shoulder - Dislocation/subluxation
Shoulder - Contracture
Vacant
Vacant
Vacant
Elbow - Cubitus valgus
Elbow - Cubitus varus
Elbow - Bifid distal humeral metaphyses
Elbow - Radio-ulnar synostosis
Elbow - Radio-humeral fusion
Elbow - Radial head dysplasia
Elbow - Radial head hypoplasia/aplasia
Vacant
Elbow - Dislocation/subluxation
Elbow - Contracture
Vacant

Vacant
Vacant
Wrist - Delayed ossification of epiphyses
Wrist - Large epiphyses
Wrist - Small epihyses
Wrist - Flat epiphyses
Wrist - Irregular epiphyses
Wrist - Fragmented epiphyses
Wrist - Sclerotic epiphyses
Wrist - Cone-shaped epiphyses
Wrist - Splayed metaphyses
Wrist - Cupped metaphyses
Wrist - Irregular margins of metaphyses
Wrist - Sclerotic margins of metaphyses
Wrist - Circumscribed ossification defects of metaphyses
Wrist - Spurs of metaphyses
Wrist - Premature fusion of metaphyses
Wrist - Dislocation/subluxation
Wrist - Contracture
Wrist - Delta deformity
Wrist - Madelung deformity
Vacant
Vacant
Vacant
Hip - Delayed ossification of epiphyses
Hip - Premature ossification of epiphyses
Hip - Large epiphyses
Hip - Small epiphyses
Hip - Flat epiphyses
Hip - Irregular epiphyses
Hip - Fragmented epiphyses
Hip - Sclerotic epiphyses
Hip - Cone-shaped epiphyses
Hip - Splayed metaphyses
Hip - Cupped metaphyses
Hip - Irregular margins of metaphyses
Hip - Sclerotic margins of metaphyses
Hip - Circumscribed ossification defects in metaphyses
Hip - Spurs of metaphyses
Hip - Premature fusion of metaphyses
Hip - Thin femoral neck
Vacant
Hip - Short femoral neck
Hip - Broad/thick femoral neck
Hip - Medial elongation of femoral neck
Hip - Absent femoral neck

Hip - Aseptic necrosis of femoral neck
Hip - Collapsed femoral head
Hip - Dislocation
Hip - Contracture
Hip - Coxa valga
Hip - Coxa vara
Vacant
Vacant
Vacant
Knee - Delayed ossification of epiphyses
Knee - Large epiphyses
Knee - Small epiphyses
Knee - Flat epiphyses
Knee - Irregular epiphyses
Knee - Fragmented epiphyses
Knee - Sclerotic epiphyses
Knee - Cone-shaped epiphyses
Knee - Hat-shaped tibial epiphyses
Knee - Splayed metaphyses
Knee - Cupped metaphyses
Knee - Irregular margins of metaphyses
Knee - Sclerotic margins of metaphyses
Knee - Circumscribed ossification defects of metaphyses
Knee - Spurs of metaphyses
Knee - Premature fusion knee metaphyses
Knee - Absent/ hypoplastic patella
Knee - Dislocated patella
Knee - Granular calcifications of patella
Knee - Prominent medial plateau of tibia
Knee - Dislocation/subluxation
Knee - Contracture
Knee - Genu valgum
Knee - Genu varum
Vacant
Ankle - Delayed ossification of epiphyses
Ankle - Large epiphyses
Ankle - Small epiphyses
Ankle - Flat epiphyses
Ankle - Flat medial tibial condyle
Ankle - Irregular epiphyses
Ankle - Fragmented epipyses
Ankle - Sclerotic epiphyses
Ankle - Cone-shaped epiphyses
Ankle - Splayed metaphyses
Ankle - Cupped metaphyses
Ankle - Irregular margins of metaphyses

Ankle - Sclerotic margins of metaphyses
Ankle - Circumscribed ossification defects of metaphyses
Ankle - Metaphyseal spurs
Ankle - Premature fusion of metaphyses
Ankle - Dislocation/subluxation
Ankle - Contracture
Vacant
Vacant
Vacant
Vacant

Hands X-ray Features

Hands - Absent
Hands - Retarded ossification
Hands - Accelerated ossification
Hands - Oligodactyly
Hands - Polydactyly/hexadactyly
Hands - Syndactyly
Hands - Brachydactyly
Hands - Arachnodactyly
Hands - Overgrowth/macrodactyly
Hands - Brachycheiry/short tubular bones
Vacant
Vacant
Vacant
Carpal bones - Calcific stippling
Carpal bones - Irregular
Carpal bones - Supernumerary
Carpal bones - Fused
Carpal bones - Osteolysis
Carpal bones - Delayed ossification
Carpal bones - Accessory ossification centres
Carpal bones - Accentuated maturation
Vacant
Vacant
Vacant
Vacant
Metacarpals - Absent
Metacarpals - Short
Metacarpals - Wide
Metacarpals - Rectangular
Metacarpals - Overtubulated
Metacarpals - Irregular epiphyses
Metacarpals - Cone-shaped epiphyses
Metacarpals - Irregular metaphyses
Metacarpals - Proximal pointing
Metacarpals - Osteolysis
Metacarpals - Hyperostosis
Metacarpals - Sclerosis
Metacarpals - Enchondromata
Metacarpals - Exostoses
Metacarpals - Cysts
Metacarpals - Pseudoepiphyses
Metacarpals - Bizarre configuration

Metacarpals - Increased index
Vacant
Vacant
Vacant
Vacant
Hand phalanges - Partially absent
Hand phalanges - Partially fused (Symphalangism)
Hand phalanges - Hypoplasia of proximal phalanges
Hand phalanges - Hypoplasia of middle phalanges
Hand phalanges - Hypoplasia of distal phalanges
Hand phalanges - Short proximal
Hand phalanges - Short middle
Hand phalanges - Short distal
Hand phalanges - Overtubulated
Hand phalanges - Epiphyseal irregularities
Hand phalanges - Cone-shaped epiphyses proximal
Hand phalanges - Cone-shaped epiphyses middle
Hand phalanges - Cone-shaped epiphyses distal
Hand phalanges - Metaphyseal irregularities
Hand phalanges - Osteolysis of proximal
Hand phalanges - Osteolysis of middle
Hand phalanges - Osteolysis of distal
Hand phalanges - Wide
Hand phalanges - Hyperostosis
Hand phalanges - Sclerosis
Hand phalanges - Enchondromata
Hand phalanges - Exostoses
Hand phalanges - Cysts
Hand phalanges - Bifid distal
Hand phalanges - Hyperplastic distal tufts
Hand phalanges - Bizarre configuration
Vacant
Vacant
Thumbs - Broad
Thumbs - Enlarged terminal phalanges
Lack of ossification in single phalanges and metacarpals (switch)
Thumbs - Hypoplasia
Thumbs - Reduplicated proximal phalanx
Thumbs - Short proximally implanted
Thumbs - Short distal phalanx
Thumbs - Triphalangeal
First Metacarpal - Hypoplastic
First Metacarpal - Ovoid
Hypoplasia of first fingers (switch)
Fourth Metacarpals - Short
Fifth finger - Hypoplasia of middle phalanx

Fifth finger - Clinomicrodactyly
Simian crease
Ulnar deviation of fingers
Vacant
Vacant

Feet X-ray Features

Feet - Absent
Feet - Equinovarus
Feet - Polydactyly
Feet - Syndactyly
Feet - Oligodactyly
Feet - Brachydactyly
Feet - Arachnodactyly
Feet - Macrodactyly
Vacant
Vacant
Vacant
Tarsal bones - Calcific stippling
Tarsal bones - Fusions
Tarsal bones - Osteolysis
Tarsal bones - Supernumerary
Tarsal bones - Delayed ossification
Tarsal bones - Malformed
Tarsal bones - Irregular overgrowth and calcification
Tarsal bones - Vertical talus
Tarsal bones - Double ossification centre in calcaneum
Vacant
Vacant
Metatarsal bones - Wide
Metatarsal bones -Short
Metatarsal bones - Metatarsus adductus
Metatarsal bones - Osteolysis
Metatarsal bones - Pseudoepiphyses
Metatarsal bones - Cone-shaped epiphyses
Metatarsal bones - Enchondromata
Vacant
Vacant
Vacant
Feet phalanges - Hypoplasia of proximal
Feet phalanges - Hypoplasia of middle
Feet phalanges - Hypoplasia of distal
Feet phalanges - Short proximal
Feet phalanges - Short middle
Feet phalanges - Short distal
Feet phalanges - Fusion
Feet phalanges - Osteolysis
Feet phalanges - Hyperostosis
Feet phalanges - Sclerosis

Feet phalanges - Enchondromata
Feet phalanges - Exostoses
Feet phalanges - Cysts
Feet phalanges - Hyperplastic distal tufts
Feet phalanges - Cone-shaped epiphyses
Feet phalanges - Bifid distal
Feet phalanges - Bizarre configuration
Vacant
Vacant
Vacant
Vacant
First toes - Hypoplasia
First toes - Large first metatarsal and phalanges
First toes - Enlarged terminal phalanges
First toes - Duplication
Vacant
Vacant
Vacant

Global X-ray Features

Dysplastic changes of bone - Widespread
Macromelia
Micromelia
Phocomelia
Osteolysis - Generalized or Multicentric
Osteoporosis - Generalized
Radiolucent bones - Generalized
Osteosclerosis - Generalized
Sclerotic foci in skeleton - Generalized
Rickets-like changes with spur formation - Generalized
Coarse trabeculations of bone - Generalized
Altered trabecular pattern of bone - Generalized
Irregular periosteal sclerosis - Generalized
Accelerated bone age - Generalized
Retarded bone age - Generalized
Dysostosis multiplex - Severe
Dysostosis multiplex - Moderate
Dysostosis multiplex - Mild
Exostosis-like opacities near tendon insertions
Bone Hypertrophy - Generalized or localized
Hyperostosis/New bone formation - Generalized
Fractures - Widespread (tubular and flat bones)
Enchondromata - Widespread involvement including flat bones

APPENDIX E

This appendix contains five examples of disease profiles, randomly chosen from the 202 present in the knowledge base.

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN

Chondrodysplasia Punctata Dominant Form

Conradi-Hunermann Syndrome

Genetics: AD with variable expression; XL form(?)

Shaul et al (1975) Am J Dis Child 129:360-362

Also known as "stippled epiphyses, but several other conditions

manifest punctate epiphyseal changes.
Press any key to continue...

Face - Unusual/Characteristic
Nose - Depressed nasal bridge
Abnormal skin/pigmentation
Alopecia
Asymmetry
Spine - Generalized structural abnormalities
Spine - Generalized Scoliosis/ Kyphoscoliosis
Vertebral bodies - Calcific stippling
Long bones - Calcific stippling
Long bones - Short shafts
Long bones - Asymmetric shortening
Long bones - Punctate epiphyses during infancy
Carpal bones - Calcific stippling
Tarsal bones - Calcific stippling

N<ext> P<rev page> <e>X<it>

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN

FG Syndrome

Opitz-Kaveggia syndrome

FGS

Genetics: XL recessive

Opitz et al (1982) Am J Med Gen 12:147-154

Riccardi et al (1977) Am J med Gen 1:47-58

Press any key to continue...

Small stature/Dwarfism
Face - Unusual/Characteristic
Mental retardation
Eyes - Non-specific defects
Ears - Abnormal external ears
Ears - Deafness/impaired hearing
Mouth - Macrostomia
Philtrum - abnormal
CNS defect (excluding cranial nerves)
Macrocephaly
Microcephaly
Hypertelorism
Frontal bones - Prominent/bossed
Mandible - hypoplasia/Micrognathia

N<ext> P<rev page> <e>X<it>

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN

SPONASTRIME syndrome

Genetics: AR

SPOndylar lesions, NAsal changes, STRIated MEtaphyses in four sibs

of unaffected parents.

Fanconi et al Helv Paed Acta 38 267-280 1983
Press any key to continue...

Small stature/Dwarfism
Nose - Depressed nasal bridge
Macrocephaly
Calvaria - Thickened
Frontal bones - Prominent/bossed
Lumbar spine - Accentuated lordosis
Lumbar spine - Interpedicular narrowing
Lumbar spine - Spina bifida
Vertebral bodies - Irregular endplates
Vertebral bodies - Pear-shaped
Long bones - Generalized metaphyseal dysplasia
Long bones - Metaphyseal striations

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN

Arthro-ophthalmopathy

Stickler Syndrome

Genetics: AD with variable expression

Opitz et al (1972) N Engl J Med 286:546-547

Press any key to continue...

Eyes - Myopia/Hypermyopia
Ears - Deafness/impaired hearing
Mouth - Cleft palate
Mandible - hypoplasia/Micrognathia
Vertebral bodies - Irregular endplates
Vertebrae - Platyspondyly
Long bones - Excessive length compared to width
Long bones - Generalized epiphyseal dysplasia
Joints - Degenerative osteo-arthropathy

N<ext> P<rev page> <e>X<it>

DEPT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN

Pycnodysostosis

Genetics: AR

Elmore SM (1967) J Bone Joint Surg (Am) 49:153-162

Press any key to continue...

Small stature/Dwarfism
Calvaria - Late open fontanelles
Calvaria - Large/wide fontanelles
Mandible - Wide angle
Long bones - Osteolysis
Long bones - Fractures
Osteosclerosis - Generalized
Fractures - Widespread (tubular and flat bones)

N<ext> P<rev page> <e>X<it>

APPENDIX F

Gamut Index Program

This appendix contains the listing of program Gamshow.pas which performs a differential diagnostic facility similar to that of the book "Gamut Index of Skeletal Dysplasias" by Kozlowski and Beighton. The program assumes the existence of certain random access files which had been created by means of the database program dBASEII and ad hoc pascal programs. As this is a minor component of this thesis, the mechanism by which these random access files were created, will not be detailed.

```
Program Gamshow;

{                              Program description

This program presents a differential diagnostic facility similar
to the Gamut Index by Kozlowski and Beighton.  It presents suc-
cessive options lists from which selections may be made and which
best fit a problem condition under investigation.

The program uses random access files sections.ran, gamut1.ran and
lookup1.ran.  These had been preconstructed using dBASEII and the
method of preparation will not be detailed here.

Only section 1 of the Gamut Index (Generalized Skeletal Abnormalities)
can be emulated for the time being.  Section 2, dealing with Localized
conditions will be dealt with as a future project.

                        }
type
     anystring =string[80];
     shortstring = string[3];
     newstring =string[70];

     lookuprecord =
                    record
                          address:integer;
                          maximum:integer;
                          contents:integer;
                          itemstring:string[16];
                   end;

     gamutrecord =
                    record
                           text:newstring;
                           propointer:integer;
                    end;

    sectionrecord =
                    record
                           number:integer;
                           heading:anystring;
                    end;


{ GLOBAL VARIABLES AND FILES                                              }

var
    i,j,k:integer;
    labelindex,section,errpos:integer;
    searchstring:anystring;
    sectionstring:shortstring;
    lookupfile,gamutfile:anystring;

    lookupbuff:lookuprecord;
```

```
   gamutbuff:gamutrecord;
   sectionbuff:sectionrecord;
   labels:array[1..10]of anystring;


   lookup:file of lookuprecord;
   gamut :file of gamutrecord;
   sections:file of sectionrecord;

{ GLOBAL UTILITY PROCEDURES                                              }

procedure msg(x,y:integer;s:anystring);

{  prints a message s at coordinates x,y                                 }

begin
     gotoxy(x,y);
     write(s);
end;


procedure frame(upperleftx,upperlefty,lowerrightx,lowerrighty:integer)

{draws a double-rimmed box using IBM ASCII character set                 }

begin
     clrscr;
     gotoxy(upperleftx,upperlefty);write(chr(201));
     for i:= upperleftx+1 to lowerrightx-1 do write(chr(205));
     write (chr(187));
     for i:= upperlefty+1 to lowerrighty-1 do
     begin
          gotoxy(upperleftx,i);write(chr(186));
          gotoxy(lowerrightx,i);write (chr(186));
     end;
     gotoxy(upperleftx,lowerrighty);
     write(chr(200));
     for i:=upperleftx+1 to lowerrightx-1 do write(chr(205));
     write(chr(188));
end;

Procedure Capture (var s:shortstring);

Var Ch: Char;

Begin

S := '';

Repeat
      Read(Kbd,ch);
      If  Ch in ['0'..'9','b','B',^M]  then
          Begin
            s := s+ ch;
            write(ch);
```

```
          End;
Until ch = ^M;
s:=copy(s,1,length(s)-1);
End;


{ PROCEDURES MAKING UP STATEMENTS OF MAIN PROGRAM BODY                    }

procedure initialize;

begin
     labelindex:=1;
     frame(5,1,70,24);
     msg(30,3,'GAMUT INDEX');
     msg(10,5,'DEPARTMENT OF HUMAN GENETICS UNIVERSITY OF CAPE TOWN');

end;

procedure select_section;
var
   line:integer;

begin
     assign (sections,'sections.ran');
     reset(sections);
     seek(sections,1); { this is to skip over record 0 which is null }
     line :=8;
     while not eof(sections) do
           begin
                read(sections,sectionbuff);
                line := line+1;
                gotoxy(8,line);
                writeln(sectionbuff.number,'  ',sectionbuff.heading);
           end;
     msg(8,23,'select a section, enter corresponding number    ');

     Capture(sectionstring);
     searchstring := sectionstring;
     Val(sectionstring,section,errpos);
end; {select_section  }

procedure select_files;

begin
     lookupfile := 'lookup'+sectionstring+'.ran';
     gamutfile := 'gamut'+sectionstring+'.ran';

     assign(lookup,lookupfile);reset(lookup);
     assign(gamut,gamutfile);reset(gamut);
end; {select_files}


procedure exit;

begin
```

```
      close(sections);
      close(lookup);
      close(gamut);
      frame(1,1,78,24);
      msg(30,3,'GAMUT INDEX');
      msg(30,12,'END OF SESSION');
      halt;
end {exit};


procedure display;
var required : boolean;
    pointers : array[1..20] of integer;

{SUB-PROCEDURES LOCAL TO PROCEDURE DISPLAY}

procedure show_subsections;

   begin
        seek(sections,section);
        read(sections,sectionbuff);
        clrscr;
        msg(8,2,'Section '+sectionstring+' '+sectionbuff.heading);
        labels[labelindex] := sectionbuff.heading;
  end; {show_subsections }

procedure locate(s:anystring);

var found:boolean;

  begin
     found := false;
{ note : pointer moves forward all the time except with go_backward
  which resets file lookup

     while (not found) and not eof(lookup) do
       begin
         read(lookup,lookupbuff);
         if s=copy(lookupbuff.itemstring,1,length(s))then found:=true;
       end;
       if  not found then required := true;


  end;{locate}



procedure list_items;

var intstring:string[2];
    line:integer;

begin
  seek(gamut,lookupbuff.address);
  line:=5;
```

```
  for i:=1 to lookupbuff.contents do
    begin
      read(gamut,gamutbuff);
      str(i,intstring);
      msg(8,line,searchstring+'.'+intstring+' '+gamutbuff.text);
      pointers[i] := gamutbuff.propointer;
      line:=line+1;
    end;
end; (* list_items *)

procedure show_disease;
var ch:char;

begin
     clrscr;
     gotoxy(20,10);
     if pointers[section] = 0 then
        writeln ('No disease profile available')
     else
         begin
              writeln('A disease profile may exist for ',searchstring);
              writeln('The searchstring is ',searchstring);
         end;
     writeln('Enter 0 to terminate');
     repeat read(kbd,ch);until ch = '0';
     exit;
end;


procedure go_forward;

begin
     labelindex := labelindex+1;
     searchstring := searchstring+'.'+sectionstring;
     clrscr;
     msg(10,2,'Subsection '+searchstring);
     val(sectionstring,section,errpos);
     seek(gamut,lookupbuff.address+section-1);
     read(gamut,gamutbuff);
     labels[labelindex] := gamutbuff.text;
     msg(30,2,labels[labelindex]);
end;  (* go_forward  *)


procedure go_backward;

procedure remove_section; (* local to go_backward  *)

begin
     (* find furthest period  *)
     while copy(searchstring,length(searchstring),1)<>'.' do
       searchstring := copy(searchstring,1,length(searchstring)-1);
       { remove the period                                        }
       searchstring := copy(searchstring,1,length(searchstring)-1);
end;  (* remove_section  *)
```

```
begin  (* go_backward  *)
     if length(searchstring)>2 then
     begin
          remove_section;
          if labelindex>1 then
            labelindex := labelindex-1
          else
            labelindex := 1; (* dummy *)
          clrscr;
          reset(lookup);
          if labelindex>1 then
            msg(2,2,'Subsection '+searchstring)
          else
            msg(2,2,'Section '+searchstring);
          msg(20,2,labels[labelindex]);
     end
     else
        begin
          reset(lookup);
          msg(10,1,'At beginning of Index');
        end;
end;

(*  body of procedure display  *)
begin
     required := false;
     show_subsections;
     while (sectionstring<>'0') do
     begin
          locate(searchstring);
          if required then
             show_disease
          else
             list_items;
          msg(10,23,'Select number, 0 to exit, B to go backward  ');
          Capture(sectionstring);
          if sectionstring='0' then exit
          else
            if (sectionstring = 'b')or (sectionstring = 'B') then go_bac
kward
             else go_forward;
     end;
end;


(* MAIN PROGRAM BODY  *)

begin
     initialize;
     select_section;
     select_files;
     display;
end.
```