SOME APPLICATIONS of the COLLECTION and RETRIEVAL of INFORMATION in UNIVERSITY ADMINISTRATION

James Hartley-Taylor

This thesis is submitted in fulfilment of the course requirements for the Degree of Master of Science in the University of St. Andrews, Scotland. June 1970

ACKNOWLEDGEMENTS

I should like to express my sincere thanks to Professor A. J. Cole, whose patient and understanding guidance has been invaluable at all stages of this thesis.

The assistance of Dr. R. Erskine and his staff is also acknowledged with gratitude.

J. H-T.

TABLE of CONTENTS

APPENDIX 2

APPENDIX 3

DIAGRAMS

1. DATA PROCESSING

a) Introduction

Data is processed in order to yield information. The basic reasons for processing data are:

1) to keep detailed facts about individual transactions for records.

2) to produce operating documents, paysheets, class lists; to enable day-to-day administration of the organization to continue.

3) to analyse facts into informative reports, annual reports, statistical surveys; to enable policy decisions to be made on the basis of knowledge.

Management is not primarily interested in source reports, documents, data; its primary requirement is information in easily assimilable form - hence data processing occupies an important role in the running of an organization.

The word 'data' may be used to cover all the facts obtained, and the term 'information' to denote the particular combination of facts of relevance to the project under consideration.

The value of information is dependent upon its accuracy, otherwise it has negative value; its novelty, it tells the reader something he did not know before; focus, it gives a clear picture of the relevant facts and does not confuse the reader with extraneous detail.

b) Operations in Data Processing

The basic operations in the processing of data are:

1) ORIGINATION
2) PROCESSING
3) OUTPUT

1) The ORIGINATION of data in a form suitable for processing involves three stages:

i) Data Collection - capturing the facts when they are available; for instance the time at which an employee starts or stops work may be recorded in writing by a timekeeper, stamped or punched in numerals by a time clock, recorded on disc, magnetic or paper tape by a computer. Data collection often begins with the manual operation of keyboards, although recent developments permit the automatic collection of data in machine-processable form. The collection of data in machine-processable form may yield information which requires mechanical interpretation before it can be made available in a form comprehensible to man. Alternatively, characters may be produced in a form legible to both machines and people; in which case character reading machines may be required to convert the characters to a form suitable for automatic processing.

ii) Data Verification includes checking data to determine whether they are in the approved format, convey the correct meaning to the reader, and will lead to the appropriate action.

The desired degree of accuracy does not necessarily imply perfection; a slight misspelling of a name, for example, may be trivial; but the crediting of the wrong customer with a purchase, or the wrong student with a qualification can cause much trouble.

The simplest type of verification is to ensure that each data field contains the correct kinds of characters, numeric and alphabetic - an error in this sphere defies further processing and must be corrected before processing can proceed. More difficult problems of verification arise from incompleteness of data input.

The meaning of some data in terms of reasonableness serves to verify it; it is possible for a person to embark upon a university career at the age of 91 but it would be unreasonable to assume unquestioningly that this is his age - it may be that he is aged 19, and has made an error. The detection of possible errors need not stop the checking, the questionable data need only be flagged as requiring further verification.

Some data which pass the format and meaning tests may require some verification action before processing, this is evidenced by the processing, or clearing, of a cheque - the signature and the available funds must be verified before the bank will honour the cheque.

2) The PROCESSING of data involves the rearranging of input and the processing of files.

i) Rearrangement involves classifying data by type and ordering them into sequence without changing their content by computation, this is required when two or more kinds of transactions originate together but require separate handling; the processing can, however, be eliminated by arranging for the separate origination of each class of data. A second way of rearranging data is to use each type of transaction to produce several outputs. Depending upon the kind of output desired and the file arrangement, the transactions are arranged in different sequences for efficient processing.

In addition to the problem of arranging transactions in the same sequence as the file before processing them, there is the related problem of keeping the file itself in a specified sequence and eliminating inactive records. Maintaining a file of customers' accounts, items of stock, or student records, for example, may require the insertion of records for new customers or students according to some sequence, and the deletion of inactive records as custom is transferred away, and students leave.

Files kept on magnetic tape are often arranged in alphabetic or numeric sequence and are most efficiently processed in the same sequence. Random - access equipment, on the other hand, is designed to handle transactions without regard for file or transaction sequence.

A third kind of rearrangement occurs when the elements of data in an item are in one sequence but are wanted in a different sequence.

Data may be rearranged within an item during input conversion, processing, or output editing.

ii) File Processing involves the manipulation of data to yield the desired type of output - estimate, prediction, decisions based on quantitative criteria. The type of output required also determines the manner of manipulation - mathematical calculation, alphabetization, transliteration, translation.

3) The OUTPUT is required in an acceptable and meaningful form; since processed results are seldom in precisely the form desired, it is often necessary to select the information to be output.

i) Job manufacturing costs are examples of historical reports about what has happened.

ii) A manufacturing schedule is a forecast of what is supposed to happen.

iii) A bill sent to a customer is an example of an action document.

The content, frequency, and format of output are determined jointly by requirements and capabilities of the people who use the output and those who prepare it.

The method of output preparation adopted depends upon the way in which demands occur - scheduled, or random - the lenght of time available for meeting the demands, the ability to forecast what will be demanded.

Output preparation becomes difficult when the number of transactions is high and the reporting interval and permissible delay in processing are short. One may establish a scale of difficulty of output preparation from the preparation of an annual report on fixed assets through weekly reports of receipts, sales, and inventories, to the up-to-the minute reports of the sales activity or inventory of any one item, airline ticketing, for example.

Outputs not anticipated in advance may pose extremely difficult problems.

c) Processing Facilities

Data processing facilities must be able to :

1) receive input data
2) manipulate data according to rules of mathematics or logic
3) keep records
4) produce output

The Human Operator is the earliest form of data processor.
A person receives input chiefly by seeing or hearing, stores it in his brain which also serves as an operating and control unit.
His outputs are oral or written reports and various physical actions.

Whilst the human mind can cope unaided with relatively simple situations; it is slow in performing arithmetical operations and erratic in applying logical rules.
The earliest aid to the human processor was the development of written records. Records increase the capacity and reliability of data storage, which otherwise would be restricted to what people can remember. Written records are also a simple and reliable method of data transmission.
Man still remains responsible for data input, control, and output; unless aided by other processing facilities.

Man's processing ability has been further widened by the development of special mechanical aids to data processing, such as the calculating machine and the typewriter; the two functions of which can be combined to create bookkeeping and accounting machines which allow an operator to perform at the same time the multiple operations of preparing statements, ledgers, and journals.

The most recent development in data processing facilities is the electronic data processor, the unique feature of which is its ability to store modifiable operating instructions in the same way and in the same place as the data to be processed. High speed is one of the greatest boons of computers; the ratio of computer to manual time needed to solve the problem can be 1:500000 - and this will increase as computers become more powerful.

d) <u>The Present Status of Electronic Data Processing</u>

Whilst it has hitherto been claimed by many writers on the subject of man's inter-relationship with computers that whilst the computer can be left to take care of the 'number - crunching' aspects of computing; man has the monopoly in the sphere of decision making. It is true that decision taking, according to rational criteria, especially in real - world conditions is a difficult task to translate into terms acceptable by a computer because of the incompleteness of available data. For this reason and because of man's belief in his judgement capability; the transfer of decision making to a computer has been retarded and man has continued to take decisions which have, in fact, been based more on hunch and guess-work than on a complete consideration of possibilities subject to logical criteria. There is, however, no rational reason to believe that it is impossible to programme computers to perform just these tasks.

The optimum product-mix, as determined by linear programming techniques, has become accepted as a valuable sphere of computer decision making; the extension of computer methods to decisions which have more personal connotations, the selection of students for university admission, for example, is more a question of consumer acceptability than of conceptual or programming difficulty. If the programme were written in such a way as to incorporate an acceptable weighting system for non-academic attributes, if these were considered important, such a method of selection would not only have the advantage of speed, but also of impartiality, and it is to be hoped, of accuracy in selecting candidates with a study-potential which turns out to have been realised in subsequent results. The use of a method of selection which uses precisely determined criteria and procedures for selection is conducive to the further improvement of the method, guided by feed-back from results obtained. It now becomes possible to compare, objectively, the performance of those selected with the criteria used for making the selection, and from this to discover which are the important qualifications for success and the common characteristics of failure. From this basis one is in a position to modify the

characteristics sought in the selection process in order to optimise the efficacity of the procedure in the future.
If the validity of the selection process could be more fully realised than under the present human system, it could be expected to have the positive attributes of reducing the drop-out rate and its attendant human misery, as well as the important economic consequence of making investment in education a more rewarding venture. Computer selection is envisaged in the near future at Loughborough University of Technology.
This type of development indicates that, in the light of advancing computer technology, it is inaccurate to visualise data processing by computer as confined to the replacement of low grade clerical personnel by machines; it should rather be seen as one stage in the process of administration, all stages of which can be conducted by computer as an integrated process subject to the availability of computers with sufficiently sophisticated programmes and the required degree of consumer - acceptability.

2. BASES OF THIS PROJECT

Whilst data processing methods have been evolved primarily to satisfy the needs of business administration; a university can also reap real benefits by applying these methods in its administrative procedures.

The method of student record keeping hitherto employed at St. Andrews involved:

1) Completion of matriculation forms by students annually.
2) Checking of replies by administrative staff.
3) Coding.
4) Punching of computer cards.
5) Computer processing, to yield examination timetables and for record storage; although documentary records continue to play an important role in administration.

Modification of existing procedures was called for to meet UCCA and UGC requirements for more information, as well as to take advantage of the added facilities of the new computer - IBM 360/44.

This project, accordingly, seeks to lay the foundations for an integrated system of data processing which will serve as a corner stone in the whole process of university administration.

The first positive step, therefore, was a comparative study of data processing procedure employed by universities already using computers, as well as enquiries into commercial methods of data processing.

Information on methods of data processing employed by other universities was obtained by sending the attached questionnaire to sixteen universities, fourteen of whom replied.

A covering letter, which stated the aim of this project, invited completion of the questionnaire and the sending of a specimen copy of the Registration and Coding forms used; the seven questions in the questionnaire focussed upon the following stages of processing:

i) form completion

ii) coding procedure

iii) error detection and analysis

iv) validation of the record produced

The format of the questionnaire was multiple-choice orientated with supplementary open-ended questions designed to elicit amplification of the more detailed questions.

The replies indicated that a variety of matriculation form formats is in use at universities with computerized data processing systems.

A state of flux was shown by a wide-spread discrepancy between current practice and expressed intention or recognized optimum. The salient features which emerged from the replies are outlined below.

Question 1.

PREPRINTING OF REGISTRATION FORMS

There is a general trend towards the preprinting of information which is not expected to change from year to year, for example date of birth, next of kin; the source of the information was previous year's records completed by the student and stored on tape. East Anglia, however, noted that 30% of students write information which has been preprinted. At Southampton and Newcastle correct completion is ensured by filling the form and coding in the presence of a member of staff (tutor); the draw-back of this method is that it may be time-consuming for the staff concerned. There appears a general movement away from reliance upon form completion by the students and towards the taking of information from U.C.C.A., the Departments, and Residences. In this context it is worth while to note that East Anglia reports that it finds that the preprinted forms are not good punching documents, and that Cambridge does not use registration forms at all but conducts all central administrative record keeping by clerical staff working from U.C.C.A data and application forms.

Questions 2 & 3.

POST-COMPLETION ADDITIONS

In the majority of Universities in the survey, the accounting information was added after the students had returned the form. In many cases coding does not appear to have been an integral part of form completion, for example by marking a numbered box, but appears to have been done subsequently by a clerk, usually from the University's clerical department.

Both instances appear wasteful in man-power and can, apparently, be to a large extent eliminated by the combination of the completion and coding process on the form itself, as exemplified by Glasgow.

Question 4.

ERROR-CHECKING

Although checking is reported as conducted by various people; students, Assistant Registrars (Newcastle), members of staff and others; the most efficient procedure appears to have been devised by London - this involves:

i) Coding and Punching and entry of initial information to the Computer about each student proceeds simultaneously with the checking of this information.

ii) The detection of an error causes modification of the computer record just generated.

It is claimed that this system produces a definitive list of registered students both quickly and accurately.

Question 5.

SOURCES OF ERROR

This question was not completed by many Universities, most of the replies were merely rough estimates.

Authoritative information was only supplied by Glasgow, which reports very low per-centages compared with Reading's claim that 10% of records require at least one correction. It is, perhaps, significant that Glasgow employs the services of an outside computer bureau. This bureau conducted punching and verifying of cards and also wrote a programme to check the information on the cards.

Errors in coding through vagueness of specifications and mistakes in coding itself figure prominently among those other universities which replied to this question.

Question 6.

MANUAL RETURNS

Manual returns are kept by a few of the Universities, but appear to be losing favour. The trend appears to be for the computer to producelists, which are checked by the interested parties, departments and residences for example, prior to the production of definitive statistics by the computer.

Question 7.

FUTURE PLANS

These include the extension of the present system for the processing of registration forms to link with admissions and archives. Greater use of U.C.C.A files as a source of information is also envisaged. These plans involve the use of a continuous file, rather than the preparation of seperate records for each year.

Questionnaires

The formulated series of questions which form a questionnaire depend for their efficacity upon two factors; primarily upon the respondent's ability to answer the questions accurately, secondarily, and in inter-action with the first, the suitability of the form of question to the eliciting of the information required. It follows, therefore, that one's aim in the formulation of a questionnaire is the optimal choice of question type to elicit the information required.

From the methodological point of view, the most obvious differentiating feature of data is the number of categories required to take account of the vast majority of cases. On the one hand, there can be a dichotomous categorization to some questions for example male/female. On the other hand there is the unique combination of name and address. Between these two poles there stretches a continuum in the number of categories required to take account of the vast majority of cases. Whilst for storage purposes it is necessary to retain unaltered such information as 'name' or 'home address'; on the other hand the value of questionnaire data to the administrator is enhanced by having a high degree of comparability between replies. Comparability almost invariably involves the categorization of information and the consequent impairment of the individuality of the response. To this end it is often necessary to shift information along the continuum of categorization in the direction of dichotomy; for example 'date of birth' may be converted into an entry in an age group, 'father's profession' may profitably be given under the numeric Board of Trade classification, and 'home address' may profitably be converted to a 'domicile zone' - as in programme JT74.

When information is divided into a number of classes, it evidently becomes amenable to numeric processing, with its attendant economy of storage.

Corresponding to the various treatments which may be given to data there are, basically, three types of question:

1. Open-ended - especially suited to individual or personal data, for example name or home address.
   Whilst valuable to elicit information of this nature for storage, this type of question yields data with low comprability and requires much storage space.
2. Rating & Multiple Choice - suitable for data which is readily available in categorized form, for example age group, or domicile zone. This type yields information with high comprability though low individuality; also, it is economical in its storage requirements.

This type of question is well suited for computer processing as it facilitates reading of the questionnaire by electronic document readers, it is also very easily understood by respondents, hence low errors in the completion of questionnaires of this type are reported.

3. Coded Reply (specified abbreviations, numeric codes) this type too is suitable for data which is readily amenable to categorization, and as such is economical with storage; but it is liable to errors compared with the clear-cut replies elicited by the Rating/Multiple choice type.

In the sphere of matriculation schedules, one may see the relevance of open-ended questions for the collection of socio-domestic data which is more usually individual in nature and required primarily for storage; and multiple choice or coding for socio-educational data which is more usually classified in some way.

The enrolment forms sent by respondent universities show that Multiple choice is used with success by Glasgow, and Edinburgh. Coding issued by East Anglia, with reservations about the number of errors; this method is also used by Loughborough, Newcastle and Belfast. Bangor and Lancaster have open-ended questions but have many items completed by staff.

3. ON - LINE OPERATION

a) Evaluation of on-line processing of student records

It is generally accepted that on-line operation offers the greatest advantage over batch processing in the case of problems which are dependent upon a swift response available over a wide time spectrum. This may be exemplified by the airline ticketing application; where a speedy solution to the problem of space availability is a pre-requisite for an efficient sales service.

On the other hand, it is agreed that on-line operation can offer little advantage over batch processing for major relatively infrequent problems which only require to be completed by a certain dead-line, for example payroll computation. Now, student enrolment has factors common to both the above examples. On the one hand enrolment occurs once per annum, or even only once in a person's university career, so it would appear that it has much in common with the payroll example above; and yet it has similarities with the airline ticketing problem as well, for instance when information is being fed-in and a speedy solution to the problem of the acceptbility of certain replies e.g. Basque in reply to the question 'Nationality' is required.

Additionally, once the information has been stored it may frequently be desirable to consult it; for example, members of staff may wish to verify a student's eligibility for a course by consulting his file - this information can be supplied instantly with an efficient on-line system with consoles situated around the university. The processing of student enrolment data thus offers a rewarding sphere for on-line operation.

A device is said to be operating 'on-line' if it is external to the computer system, and it and the computer alternately take action; such that the external device affects the data processing operation within the computer and such that the computer affects the external device in a significant manner.

b) User-philosophy

On-line processing represents a major change in user-philosophy, compared with batch processing. Computing centres which had hitherto adhered to batch operation in which only a few professional computer operators were permitted direct access to the computer can be thrown open to simultaneous and direct use by widely dispersed users whose previous acquaintance with computing may well be slight.

In the case of student record processing, one now has the choice of having the students themselves reply to questions displayed on the consoles, or using the consoles as the in-put point for coded information in-put by clerical staff; the former represents a major innovation vis-à-vis batch processing, the latter substitutes input from the console for input from cards, though, of course, the possible advantage of instant verification or arbitration of difficult cases is opened up. On-line operation with typing staff allows the tedious verification of the input cards to be eliminated, as the input can be checked visually on the screen at the time of typing, and if necessary, easily and swiftly altered - exemplified by the treatment of Individual Data in the attached programmes. Additionally, when the computer accepts the display, interrogation routines can check the validity of the input against pre-determined criteria, as with Categorized Data in the programmes; alternatively the input can be interpreted to the user so as to highlight possible discrepancies - as in the conversion of 'date of birth' in Numeric Data into an age on a specified recent date.

These latter advantages are also applicable when the consoles are thrown open to all comers, in which case the consoles must be programmed to accept input in a form close to natural language - an innovation which is necessary to cope with the fact that students are for the most part inexperienced in computer languages. Although this type of operation would result in more corrections in input data being required, the fact that each entry requires relatively little processing would keep the waste in C.P.U. time within bounds. A considerably greater amount of time would also be required for the typing of the input than would have been the case with clerical staff; on the practical side, however, enrolment takes place at a slack time since the computer is not required for formal teaching so there is little premium on maximising the speed of the operation. Furthermore, in a world which brings an ever wider section of the community into contact with computers, such an initiation could be regarded as a valuable part of general education.

The opportunity for the rapid retrieval of past records would on the one hand pose questions of security and confidentiality, whilst on the otherhand it holds out the opportunity for eliminating the necessity of supplying full information at each annual matriculation - astudent's record need only be up-dated in the relevant respects. A limited attempt to simulate this feature is shown in the programme JT74 where returning students upon giving their name will receive a display of their 'Individual Data'. If this remains unchanged, they only need to complete the other sections, this saves considerable console-occupancy time as well as C.P.U.time, as the processing of existing data is eliminated. Timetable construction and class-list production could also be facilitated by on-line operation because difficult cases could be resolved on the spot and when a completed timetable has been produced,hypothetical changes could be introduced and the results known very quickly.

c) Organizational Advantages

When operating on-line almost all the tasks of input preparation are performed by the computer - the man at the console can effectively be the 'customer' in the figure, and the other human links in the chain can be eliminated. The communications system and the remote console take care of routing the information to the user, furthermore, a properly designed system will result in fewer input errors than a convential system, since, for one thing, fewer steps are necessary.

Sequence of events in batch processing

Fill-out request using application-orientated language	CUSTOMER	Study Results
Submit to Processing Centre	MESSENGER	Sumit to Customer
Receive Message & Log	DISPATCHER	Log Output & Dispatch
Translate to Computer-Orientated Language & Format	INFORMATION SPECIALIST	Output Control
Key Punch & Verify	KEY-PUNCH	
Run on Computer	OPERATIONS	

d) Costs

Ultimately, the pro's and con's of on-line processing must be set along the measuring-stick of cost.

1) HARDWARE COSTS

These costs for a system capable of efficient on-line operation will exceed those of batch processing. The principal forces tending to increase costs are as follows:

i) Additional working storage features

ii) Multiaccess to and independent operation of working storage

iii) Large internal high speed memory

iv) Increased storage capacity of auxiliary memories

v) Hardware speed degradation

1) i) Additional working storage features result from the need of the executive to be able to determine the instantaneous status of the computer system where attention is being turned very rapidly from one user to the next. Probably the most important feature here is the relocatability hardware - the hardware which allows part of the programme (segment/

page) to be read in from auxiliary memory into any position of the physical working storage. Since that part of the working storage may not be the same part of the programme operated from before and since, there is a requirement that the programmer (user) need not be concerned with the location of his programme each time it is to be executed, the modificattion of the addresses for relocation must be handled automatically. Therefore, in the interpretation of an address, a table look-up is accompanied by hardware to determine in which proportion of the memory the address is to be found, and then the proper modification to the address is made to determine the effective address.

1) ii) Multiaccess to and independent operation of working storage. The Protect feature must be extended from writing into unauthorized space to reading from unauthorized space.

iii) Large internal high-speed memory is required to increase the probability that the part of the programme or data which is required next is on hand in high-speed storage, and hence ready for execution. Thus it is necessary for a complex executive to reside in working-storage which cannot, because of the high frequency of usage be kept even in part in auxiliary store. The high speed memory represents the most costly part of the machine.

1) iv) Increased storage capacity of auxiliary memories is required for buffering and communicating with the user station.

1) v) Hardware speed degredation. The process of associating the address of each executed instruction with one of the various portions of the programme, and then modifying the effective address requires an additional 15/20% of computer time - this results in the basic memory cycle's being slowed by this amount. This is tantamount to the addition of a 15/20% addition to operating costs of the machine.

2) SOFTWARE COSTS

Increased programming and analysis costs are incurred as a result of the increased system complexity.

i) The programme must make allowances for the manifold of random occurences in the computer system which result from human inputs which in turn provide a multiple of interrupts to the system operation.

ii) The problem of working storage overlay. The system must enable the swapping of programming data between working storage and auxiliary storage to be considered with maximized efficiency while giving all users a sufficient number of time slices within a given period of time.

iii) The memory management problem becomes far more complex since there will be many levels of data storage depending upon the frequency of usage.

iv) A scheduler is required to take account of the various conflicting objectives of the system and the users.

v) The development of conversational mode language and debugging aids is also complicating.

e) Rate Structure

The rate structure for batch processing has evolved as a charge for computing time used plus the cost of card punching and verification. When operating on-line, which by its very nature implies time sharing, there are basically two rate structures available. IBM philosophy is reflected in charging for on-line availability of the machine, whilst Key-Data charge for the amount of computer time used. The latter would appear the more fair in that one pays for exactly what one obtains; as such this method would give a more sensitive indication of the relative efficiency of the user's programming in maximizing the return from the most expensive commodity - C.P.U. time. The IBM system, on the other hand, would place a premium on rapid keying-in because the time spent at the console keyboard serves as a basis of charging; the emphasis is shifted from programming efficiency to manual dexterity.

4. THE RATIONALE OF INFORMATION RETRIEVAL

As we have seen, on-line computing systems are well suited to the storage and subsequent retrieval and display of technical and business information.

a) Information retrieval involves:

1) COMMUNICATION
2) PATTERN MATCHING
3) ORGANIZATION

1) Communication is taken care of by using a pre-defined vocabulary (English) structured into pre-determined formats and fields according to the rules of FORTRAN.

2) Pattern Matching is said to require a succession of rules for matching which may be regarded as successive screens of ever finer detail. The method employed in the attached programmes may be outlined as consisting of:

i) the successive isolation of the first four-character unit of the item under consideration

ii) the comparison of this set with the first four-character unit of the word against which the matching is to take place

iii) in the event of the comparison's yielding a negative relationship, the first four-character unit of the next item in the sequential storage list in question is compared, and so on until a positive comparison (match) is achieved - in this case the next four-character unit of both the item for which a match is being searched and the standard from the sequential storage list are compared. If a match is achieved the process continues until all the four-character units in the 'standard' for comparison have found an identical counterpart - a message which states that a 'match' has been achieved is then displayed. If a match is not achieved, a message to this effect is displayed. In JT74 where, given the match of a person's name, his Individual Data file will be displayed; in the Archiving routine the programme will display a whole bibliography once a category or category-combination match has been achieved.

3) Organization is an attempt to provide a balance among:

i) required response time

ii) required degree of selectivity

iii) cost

The basic form of organization is the sequential list in which records are organized solely on the basis of their successive physical positions in the data set; they are read or written in the same order in which they appear and individual records cannot be deleted or inserted unless the entire data set is rewritten. Sequential organization is the only type which may be used for data sets on magnetic tape volumes, unit record equipment, and paper tape.

The stack is the most basic form of manipulation applicable to sequentially organized data sets, as only one end of the test is involved in manipulation, it invariably results, however, in the 'youngest' addition to the list being the first to be removed. The more sophisticated form of sequential data set, the queue, requires for its optimal operation two pointers F(front) and R(rear). F=R=0 when the queue is empty; this allows greater flexibility in that additions and deletions can be conducted independently. It also offers the practical advantage over the stack that the 'oldest' addition to the list is (presumably the one of the least current interest) the first to be removed. The opposite can beachieved by reversing the direction of the queue. To circumvent the problem of the queue's overrunning memory availability, Knuth points to the opportunities offered by the circular queue, the deque.

Economy in storage requirements can be achieved when there are two variable size lists, by allowing the lists to grow towards each other. The lists may independently expand and contract so that the effective maximum of each one could be significantly more than half the available space.

We have seen that the sequential list can deal with additions and deletions, but that it does not provide for easy insertion or deletion of items at points other than the beginning or end of the list; this inflexibility can be effectively overcome by the 'linked list concept' in which each node contains a link to the next node of the list. This method can be elaborated into the Indexed Sequential system.

Indexed Sequential system has records which are organized on the basis of a collating sequence determined by KEYS which precede each block of data, and which exists in space allocated on direct access volumes as prime areas, overflow areas and indexes. This type of organization gives the programmer such flexibility, and for this reason has been used in the programmes attached.

The Indexed Sequential system allows the programmer to:

i) read or write logical records whose keys are in ascending collating sequence (as in sequential organization) without requiring an index search,

ii) read or write individual logical records whose keys are in any order. For the retrieval of each record a search for pointers in indexes is required, consequently this operation is slower than reading according to a collating sequence,

iii) add logical records with new keys or delete old ones. The system locates the proper position in the data set for the new record and makes all necessary adjustments to the indexes.

This form of organization, therefore, is the one suitable for our purpose of storing and retrieving information to which additions and deletions are to be made from time to time, as new students enrol and others leave, and as course choices and addresses change during an academic career. The version of this method used most widely in the attached programmes is of a basic sequential list, each unit of which stores 4 alphanumeric characters, and an index which points to the last unit in the sequential list which belongs to the category in question. With this system the beginning of the current or previous list can easily be located- this is required for scanning, storage and display purposes.

5. THE PROGRAMMES DEVELOPED FOR THIS PROJECT

Whilst three programmes have been developed; all have the common factors of being in the sphere of data processing, and of employing as their basic technique that of the storage and subsequent selective retrieval of information. Their application is also orientated towards use in university administration.

a) Generalized Data Processing Programme

This programme is designed to handle the three types of data usually encountered by any data processing problem.

1) PERSONAL DATA - data which is passed on for storage in the form in which it is supplied by the respondent because of its being personal to him and hence unique.

2) CATEGORIZED DATA - data which, to be accepted, must be supplied in a form which conforms to the categories specified on the screen to the respondent. These categories must have been stored by the computer prior to the user display stage.

3) NUMERIC DATA - data which must conform to the required format specifications, but which is checked by the respondent himself.

The programme is written in conversational mode for use in conjunction with remote access consoles. A main programme JT74XP, handles the main I/O instructions and calls the subroutines which conduct specific operations.
In the programme only standard messages are stored as data blocks, instructions are included in the programme itself, and topics are supplied by the supervisor. The first display is a description of the programme and its capability, followed by instructions to the operator/supervisor on the inclusion of ready-supplied data attached at the end of the file, or alternatively for the opportunity to specify one's own data requirements.
If the former option is chosen , data and all inclusion flags are read-in without further human intervention through JTDATA.
The next display gives the user-instructions for use of the console.
If the latter option is chosen the next display after the choice point gives details of the categories available and the number of topics which may be considered and their permitted distribution among the types of data. This is followed by a request to enter the number of topics for individual and subsequently for categorized data. In case the summation of the topic requests for these two categories exceeds the maximum available, a message

to this effect is displayed; the user is also told of the extent by which the maximum has been exceeded and the numbers requested for each category, an opportunity is then given for a re-statement of the topic requirements. If the topic requirements fall within the prescribed limits, then the individual data topics and subsequently the categorized data topics are read in. A count of the number of topics so far included is displayed. The opportunity to include topics for the individual numeric data section, if it has been requested, is given. Finally, the number of questionnaire presentations required, is to be indicated. The operator's task is now completed and control passes to the user and the programme will run the requested number of times without further necessary intervention by the operator.

The instructions to the respondent for the use of the console now appear on the screen. If the ready-supplied data has been used, all categories will be considered, otherwise a category will only be mentioned if it has previously been requested by the supervisor. The first section to be considered is the personal data section, which includes data such as a name and address. If the screen has been read as blank, a message to this effect is displayed and the request for information is repeated. If the display request code VZTX has been inserted, control is passed to the display section at the end of the programme. If XXXX is sensed this code induces the deletion of the previous reply and an opportunity is given to replace the offending reply by the correct answer. The subroutine JTREPT acknowledges the request and handles the special repeat instructions and leads to the redisplay of the question. If all goes normally, however, the reply is passed to the subroutine JTSKOR for storage; upon exit from this routine a check is made to ensure that there has been no overflow in characters or words for storage; in the event of overflow, control is passed to JTEXIT which displays the type of overflow involved and the culpable item, the programme then aborts.

Under normal circumstances control moves to the next topic in the section. At the end of the section a message is displayed and the user must enter 'NEXT' to proceed to the next section - Categorized Data - the title and format of which are displayed. The next display is a request to reply to the question, using one of the acceptable replies previously read-in for the topic under consideration and currently displayed. The reply is then passed

to the subroutine JTIGET for checking with the master list of acceptable replies; if the item is located in the appropriate category then the message,'OK CHECK FOR <u>topic</u>, <u>reply</u>' is displayed and the programme continues. If the reply is not found then the message,'NO GO FOR <u>topic</u>, <u>reply</u>' is displayed and a request to give the correct reply is displayed. If the error occurs a second time, a message to this effect is displayed and the question is repeated. If the error persists the message, 'THE PROBLEM IS NOW PASSED TO THE ARBITER FOR CLASSIFICATION' is displayed and this activity is simulated on the user's console; as inter-console transfer is not feasible at the moment.
The name of the topic under consideration and the offending reply are displayed, as if to an arbiter. The arbiter then types in the acceptable reply and this is then stored in the normal manner. At the end of the section, a message to this effect is displayed, and the title of the next section ,individual numeric data, is displayed and control is passed to subroutine JTDATE which handles all numeric data. If numeric data has been excluded then the end of cycle message is displayed to the user.
The first question of the new section requests a date(date of birth) in the form DDMMYY which is passed to JTAGED where it is converted

into DD MONTH 19YY; to facilitate checking by the respondent this date is interpreted as an 'age' on 31 December 1969, which is displayed as well but not stored. In the event of impossible replies, a message to this effect is displayed and the question presented again. An opportunity is given for the user to correct data which, whilst not 'impossible' according to the criteria supplied to the computer, is incorrect in a matter of fact. The remaining questions deal only with years. The instructions and examples in this section particularly have been angled towards academic application, but are easily modifiable. Upon completion of this section the end of cycle message is displayed to the respondent, and once he has complied with the request to enter 'NEXT', or in fact any four character combination other than VZTX, the user instructions will be displayed for the benefit of the next respondent, and the computer is ready to process another respondent's replies. If VZTX is entered, control is passed to the display section which indicates the total number of files processed to date and the opportunity is given to continue questionnaire presentation, display of all or a specified number of processed files by calling JTSHOW, or finally terminate the run.

This programme was developed under the RAX44 system, and at the moment there is no easy way of saving files. The above represents a pilot scheme for subsequent development under 360 Main Line RAX which includes a comprehensive file storage system.

b) Matriculation Schedule Processing Programme with Display Facilities JT74

This programme is designed to illustrate a variety of data processing techniques which can readily be applied to the replies given to the questions on a matriculation schedule of the type used at the University of St. Andrews. The replies can be collected either on the traditional forms and then keyed in to the computer via the consoles, by clerical staff; or alternatively, and this programme has been designed with this possibility in view, the students themselves could supply the answers to the questions directly to the consoles. This programme was specifically constructed in order to handle the data from matriculation schedules and the topics available for consideration by the programme are pre-defined, although the inclusion of a particular category or topic can be determined by the operator, he can also supply acceptable answers for the categorized section.

Inclusion settings and acceptable replies are also available in ready-supplied form as data.

The framework of this programme is akin to the Generalized Data Processing Programme, JT74XP, which has already been described; accordingly only the divergent characteristics of this programme will be explored.

If the operator chooses not to use the ready-supplied inclusion settings and replies, he may indicate the number of topics from the 12 on the screen to be given categorized storage answers.

The first of these topics, which are stored in 4A4 format in locations 25 - 72 of the array IAD, 'Domicile Zone', incorporates a factor frequency count; this means that each time a respondent gives one of the three acceptable replies to this question, a count is kept of the number who give each of the permissible replies, and the respondent's file will be categorized according to his reply to this question, when a request to display all files from a particular category is made. The operator also has the opportunity to specify the number of pre-determined personal and numeric data topics which he wishes to have considered.

The option is also given to include 'existing files', this is a simulation of a step which was not available at the time of programme development; instead of using data which has been previously stored on disc or tape, a set of 'existing files' is merely read-in as data. If the existing files' inclusion flag has been set, the first display following the user instructions asks the respondent if this is his first matriculation. If the answer to this question is negative the respondent is asked to give his name in a specified format; as such it is passed to the subroutine JTRCRD where a search is made for a file with this name. If one is found a message to this effect is displayed, followed by the file itself and an invitation to confirm its correctness. If this is done the information on the file is incorporated into the current file on this respondent and control is then passed to the next included section (usually Categorized Data). If the file is not located, a message to this effect is displayed and the opportunity to choose between making another attempt to locate the file or to complete the full questionnaire is given. This latter option is imposed in the case of respondents who are matriculating for the first time. All 'untraceable' names are recorded and may be consulted by requesting the 'Programme Irregularities Log' in the display section. The purpose of this is to exemplify the technique of recording erroneous replies for future analysis so that the points which respondents find difficult may be studied and simplified or eliminated in subsequent programmes.

In the data collection stages, this programme follows the same general course as JT74XP. The point of divergence is in the display facilities, where the operator has the choice of seven operations. He can continue the questionnaire presentation - this simply involves a branch back to the first address of the outermost loop. The operator can display all or a specified number of files; this involves the setting up of a loop, the upper limit of which is the number of files required by the operator. The total number of files processed to-date is displayed each time the display facilities are requested. The operator also has the opportunity, as in JT74XP, to terminate the run. The facilities which are exclusive to this programme include the display of files classified according to the replies given to the question which included the factor frequency count facility; a record of the file numbers which fell within the specified categories for the count had been made in the location subroutine JTIGET.
The method used for display is simply that the numbered files are displayed by a loop which uses the successive file numbers associated with a particular category as its limits. The factor frequency count facility displays the number of students who fall within the specified categories. The display of named individual's file and the display of the programme irregularities log have already been

mentioned.

The facilities which we have illustrated show how easy it would be to obtain a wide-ranging statistical analysis of the students within moments of the final enrolment. Of course, samples could be taken at any time in the enrolment process. The sorting of the files according to the possession of a certain attribute is used to display either the whole file, or just the name of the student. This supplies the basic requirements for the production of class lists and for the construction of conflict matrices and ultimately of lecture and examination timetables. The same facility can also be used to produce personal data files on students for the information of Wardens of residences and Tutors - in the event that they do not have ready use of a remote access console. In future no doubt the electronic transmission of the relevant parts of personal data files to potential employers could be achieved.

A synthesis of these two data processing programmes illustrates the wide opportunities which are opened in the sphere of university administration by the application of the basic techniques of storage and retrieval of information. Both literal and numeric data can be handled. Any topic may be specified for consideration and any reply may be considered as acceptable, alternatively acceptable replies may be closely defined. Difficult cases may be resolved on the spot. The input can be processed to yield detailed statistical analysis of the Year's enrolment; and once recorded, all the above may be retrieved and displayed at remote locations. It is only regretted that limited facilities circumscribed the scope of these schemes.

c) <u>Archiving Programme</u>

This programme, which was developed with card input and line printer display, is designed to display bibliographies on requested topics or groups of topics. To this end, the available topics are read-in and subsequently attached to the titles of the works which the programmer designates as having relevance to the category in question. Parameter values are stored on disc, the bibliographic data is stored in a sequential list on tape.
The programme consists of a main programme which handles the main I/O operations and calls subroutines to conduct specific tasks, such as the storing of topic/category designations JTCATG, the verification of the availability of the requested categories JTFIND, and the display of bibliographies JTDISP. All the essential information for the programme is contained in four arrays: ITOP, with as many locations as possible, is used for the sequential list which stores all bibliographic data and pointers to the next entry associated with the particular topic/category; JKEY with 950 locations stores the individual characters of the category designations which are stored in 7A1 format, IKEY with 100 locations stores the end values of the category designations. The 100 x 3 array, IAH, operates in conjunction

with ITOP, IKEY and JKEY. IAH(L,1) stores the number of entries attached to the category designated 'L', which corresponds to the counter value in the IKEY array; this value is the number which was attached to the category designation when it was first read in to the JKEY array and by which the category can subsequently be designated. It is recalled that IKEY array stores the end values of the category designations stored in A1 format in the JKEY array, and the IKEY counter is augmented by 1 when each category designation is read in. IAH(L,2) stores the value of the ITOP array of the first entry for category 'L'. IAH(L,3) stores the value on the ITOP array of the last entry for category 'L' - there may be 100 'L' values to correspond with the 100 end of category designation values which may be stored in the IKEY array. The other arrays serve primarily I/O or specific subroutine functions.

The first operation is the reading of the operation code IQUIZ. A value of 11 signifies that this is the first run of the programme, and initialization is conducted accordingly; categories for the classification of records as well as initial bibliographic data are read in. For subsequent runs of the programme, a selection must be made from one of the other three other codes; 33, 55 or 77. When IQUIZ is set equal to 33, additional categories for the classification of bibliographic entries are read in and stored, the parameters which determine the positioning of these categories in the arrays are read from disc and the up-dated values returned to it prior to the end of the run. No further bibliographic data are read in under this code: in order to obtain this additional facility the code IQUIZ 55 is to be used. Code 77 calls subroutine JTDISP which will effect the display of those bibliographic entries which have been associated with specified categories; this facility may be used to display all entries associated with one specific category or only those bibliographic entries which are common to a maximum of six specified categories.

In order to illustrate the operation of the programme we shall consider a typical run for adding category designation and bibliographic entries. Operation code 11, which is used only on the first run, is the same in all respects except that the arrays and parameters are initialized in the programme rather than read from disc. In the case of code 55 being requested, the parameter values and arrays are read from disc where they have been stored since the last run. In addition to the arrays ITOP, IAR, IKEY and JKEY; the value of IENTRY, the number of bibliographic entries to date, N - the counter value for the number of categories recorded in IKEY to-date; J, the counter of the number of characters stored for the designation of category names to date, and IE - the counter of the number of locations on the ITOP array taken to date for bibliographic data. If code 77 had been specified, control would now pass straight to JTDISP which controls all bibliographic information display activities; otherwise JTCATG is called. The subroutine reads the numberof categories to be read-in from cards by the parameter ICTG, this value is then used to control a loop which will read the first seven characters on the card in A1 format, any blanks are eliminated. The individual characters, initially read into the array IAA are transferred by loop to the JKEY array and the IKEY array records the value of the end of the JKEY LIST when this word has been read in; as such it can be used as apointer to any category designation. Upon return to the main programme, a test of IQUIZ is made, if the value is 33control is transferred to the end of the programme where the principal arrays and parameters are returned to disc for storage; otherwise the first card of bibliographic data is read in 80A1 format into the store, after the 320 characters of the IAA array which are used have been set at blanks. A test is made to ascertain if the first character is an asterisk ; if so, this signals the final entry of bibliographic data for this run of the programme. A test is also made to ascertain if the 80th character on the card is an asterisk, if so another card of 80A1 characters is read in upto a limit of four cards. The 320 characters stored in IAA are then transferred in unfromatted form to tape by a loop in blocks of 80 characters (as the maximum number of characters which may be written to a tape at one time is 90). The entry count IENTRY is augmented by 1. The next step is the reading-in of the sets of seven characters used for the designation of the categories to be attached to this bibliographic entry. A test is made for an asterisk in position 1 on the card; if this is found this indicates that the last category has been read in and control passes to address 1 which is the starting point of the read-in of bibliographic data; otherwise JTFIND is called. This subroutine searches the JKEY array for a match for the category designation currently in IAV. The method is that a loop (700) with upper limit of N (the number of categories read in to-date) is set up; yet another loop (206) is set up with upper limit and lower limit determined by the values stored in IKEY array. If a match is found at the lower limit value of JKEY the next JKEY and IAV characters are compared; in the case of continued matching the loop 206 continues to its upper limit and then returns a flag (NOGO = 0) to the main programme which indicates that the category designated by IAV matches one already stored. On the other hand, if there is no match between the initial characters of the two designations, or if a full match fails before the upper limit of the loop has been reached the loop 206 is aborted, and new upper and lower limits of search are established by the outer loop (700) - in other words another category is considered as a possible match for the category designated by IAV: this procedure continues up to N times, at which stage all available categories have been considered and it must be concluded that the category in question has been wrongly designated, or has, as yet, not been incorporated into store. A message which states the failure to locate a category specified is printed upon return to the main programme of the flag (NOGO = 10). In this case, the programme is terminated and the arrays and parameters written to disc for storage. In the case that a match has been achieved, the category number (JZ) is converted to L which in turn is stored in the IAH array as outlined above, and ITOP (IE) stores the entry number and ITOP (IPX), the next entry in the array, stores a pointer which indicates the position on the ITOP array of the next entry which falls into the relevant category. Once one category has been successfully attached to the bibliographic data another one is read in until a termination request is sensed. In this case control passes to address 1 and another bibliographic entry is made; if, however, a termination request is sensed here too, the run is terminated after the arrays and parameters have been written to disc.

We have seen how data is stored; we shall now describe the method employed for its selected retrieval. Operation code 77 calls forth the display facilities of the programme incorporated in JTDISP. After the arrays and parameter values have been read from disc, control is passed directly to JTDISP. After initialization of the pointers I and KL, the seven characters of the requested designation of a category are read into IAV in format 7A1. Up to six different categories may be requested as being required for consideration together - this means that only bibliographic entries which have <u>all</u> the designated categories <u>in common</u> will be displayed, the display of bibliographic entries under a single category is, of course, possible. The end of the category - combination which has just been requested is indicated by a single asterisk, in column 1; the end-of-run request is indicated by asterisks in columns 1 & 2.

After the seven characters of the category designation have been read-in and the termination tests successfully negotiated control is passed to JTFIND which checks the correctness of the category designation and returns the number code of the category in question. In case of an error the NOGO flag is set to 10 in JTFIND and upon return to JTDISP this is detected and the run aborted, a message explaining this is displayed. If the category is located, the number code (JZ- of the category is stored in KK(I); the beginning value on the ITOP list of entries under this category, IAH(JZ,2), is stored in JEV(I); and the end value on the ITOP list of entries under this category, IAH(JZ,3) is stored in JUV(I). I is a counter which is augmented by 1 for each complete set of above data read-in. Control is then passed to address 200 where an end of category combination test is made. If this end flag is detected, control is passed to address 205 where JEV(1) - the beginning value on the ITOP list of the entries under the first category which has been mentioned in the category-combination list - is set equal to JZ. JZ is subsequently used in ITOP to find the entry number at this position; this value is in turn set equal to IFIND: the purpose of this manoeuvre is to establish a lower limit for the search through the ITOP array. If I, the counter of the number of categories requested in this category combination list, is equal to 1; in other words, if there is only one category to be located display procedure is initiated; otherwise a loop is established with index M and upper limit I (the number of categories for which to be searched). The value of M is used as the subscript for JEV array - in this way the beginning values of all the categories requested will be considered. This JEV (M) value is set equal to MM, which is then used as the subscript for ITOP; this will yield the entry number of the first entry attached to a particular category. If this entry number is found to be equal to IFIND (the entry number at the lower limit of search, as previously determined) control is passed to address 207, and if the loop goes on to consider the beginning value of the next category. If the value of IFIND (lower limit of search, based upon the category read in first in the category-combination list) is found to be less than the beginning value of the category list determined by ITOP (MM), control is passed to address 208 where JEV(1) - the beginning value on the ITOP list of the first category to be considered - is set equal to the value indicated on the ITOP list by the location following IFIND.- This is the pointer to the location on the ITOP array of the next entry which is recorded as being linked with the category in question. JUV(1), hitherto the upper limit of the category in question, so far as the search is concerned, is now set equal to IJ which is then used as the subscript for ITOP. A comparison is then made between IFIND, the lower limit, and ITOP(IJ) the upperlimit of the search for this category. If they are found to be equal, control is passed to address 213, the tape is rewound and control is passed to 199 in readiness for the processing of the next request; if not control passes to 205 (see above). If after passing through the tests the following 205, as described above, it is found that IFIND - the starting value of the first category in the list of categories - is greater than ITOP(MM), the default option of the tests which follow 205, then a further test is made which compares MM, the starting value on the ITOP list & JUV(M), the end value of the Mth. category requested in the list of categories requested. If these values are found to be equal, control is passed to address 213 where the tape is rewound in readiness for the processing of a new list of requests; otherwise JEV(M) - the starting value of the Mth category in the list of categories requested - is set equal to the value indicated by the position on the ITOP array one greater than ITOP(MM) - the starting value of the category under consideration.

This value, ITOP (MM +1), therefore points to the position on the ITOP list of the next entry associated with this category. Control is now transferred to 209 with MM set equal to the higher value of JEV(M). In thisway all the entries in the category under consideration will be examined in turn; only if IFIND - starting value of the first category in the list of categories requested - is found to be equal to the starting value of the category MM will control be passed to 207, and only upon completion of the loop 207, which provides for the consideration of entries in all categories in the list of categories to be considered, will control pass to the writing phase of the routine - this ensures that only entries which are in fact common to all categories in the list of categories requested will be displayed (the object of the exercise).

(IFIND - 1).4 - 1 represents the number of records involved in the storage of the entries up to the value indicated by IFIND. The reason For multiplying by four is that because there is a maximum of 90 characters per record on the number of characters which may be written to tape at one time, and since 4 x 80 characters have to be written to tape, four records per entry are used.

If (IFIND-1).4-1 is found to be equal to KL - pointer to the number of records passed over to date in this particular quest for entries with common categories - then there is no need to step the tape and writing can be conducted directly from the record at present pointed to by KL; otherwise a loop is set up with KL as the lower limit and KU as the upperlimit - KU has been set equal to four times the value of IFIND - 1 for the reason explained above. This loop conducts a dummy read operation between the prescribed limits, and in so doing steps the tape to the position required for display ofthe bibliographic information associated with the categories required. The actual transfer from tape (SYS003) to printing (SYS006) is conducted by loop 250 which runs from 1 (beginning of the first record) to 320 (end of fourth record) in steps of 80 (one card length). The index of loop 250 is K1 and this is used to augment the upper limit of the READ(3) loop by 79 each time so that all 80 characters will be read. The write operation is conducted by an implied 'do loop' running from 1 to 320 and which writes in format 80A1. Upon completion of the operation, KL - pointer to the number of records passed over to date - is set equal to four times IFIND 1. This concludes the first write operation cycle. The next begins automatically when JEV(1) is set equal to ITOP(JZ+1) at the address

208, as described above.

Upon completion of the operation, control is returned to the main programme for termination of the run after writing the parameter values and arrays to disc. In the event of an incorrect category designation, control is again passed to the main programme but upon arrival an error condition is sensed by the test for NOGO = 10 and a message to this effect is printed prior to the termination of the run. Programmes of this type are evidently ideal for use in conjunction with remote access equipment. Unfortunately, as has already been noted, at the time of writing storage facilities were not available on the remote access system in use; for this reason the routines in this programme were developed with card input and line printer output, and of course, the programme will serve satisfactorily in this way. It is easy to operate; the retrieval section, especially, where the user needs only select the combination of categories required from a tray of pre-punched category designation cards and end of list markers (asterisks), prior to handing them to the operator for processing. It is anticipated that these routines will be adapted for use on the remote access system once storage facilities become available. This programme could play a valuable role in the library. It also illustrates one technique for the storage and subsequent retrieval of existing files which could only be simulated in JT74.

6. TIMETABLING

a) Introduction

In step with the application of computers to filing and data processing has been their application to the 'temporal ordering of events subject to constraints' - timetabling. This is a task which can be laborious, time consuming, and really very inefficient if conducted manually.

The earliest large scale applications of computers to timetabling in the academic context were made at universities in the U.S.A. This inovation was accelerated by the larger numbers involved in U.S. universities and the wider choice of courses available to students. In 1963 Holz described his method of timetabling at the M.I.T.- the requirements for each course are given as the number of classes together with possible instructors and time patterns listed in order of preference. At each step in the compilation the best available time patterns and instructors are allocated. If there are not sufficient instructors available without causing a clash, the requirement remains unsatisfied and is marked appropriately; thus the final timetable may not contain all the required classes, prepared, as it is, on the basis of probable rather than actual student selections.

Each student selects several courses and these selections are entered into the programme. Initially, the selection for a student is checked to see if it is compatible with the timetable; if not the student cannot be scheduled and a message to this effect is printed. The next stage is to compute a workable solution; the 'value' of the schedule is computed to determine whether it is a good one or not. Output occurs if the schedule is acceptable; if not a new schedule is computed and tested. A limit is imposed on the number of schedules tried, and when the limit is reached the best schedule computed is printed. Each schedule is considered only on its own merits without reference to any other student's schedule. Thus, if a student is scheduled badly and fills the last available place in a class this step cannot be retraced. A slightly different approach to this problem has been made by Sherman (1958) at Purdue where the number of students was 20,000. With this number the time to schedule each student is of major importance. The method used aims to schedule each student as quickly as possible by reducing the number of schedules tried before a satisfactory one is found. This can be achieved by ordering the classes so that the most difficult class to fit occurs first in the student selection.

Several factors can be brought into this ordering process. The most obvious one is the number of classes available to the course. If only one class is available then the student can either be fitted into a class or not, so that rejection occurs very early. Another factor taken into account by this progamme is the distribution of students within the classes. By keeping the numbers fairly even fewer classes reach their full capacity, thus late registrants can be fitted into the timetable more easily.

In U.K. the experience of Manchester University reported by Wood (1968) is significant in this respect. Prior to computerization, three administrators and three clerks were employed from November to June in the preparation of the summer examination timetable. 85% accommodation utilization was achieved, compared with 99% after the process had been taken over by computers.

At St. Andrews timetabling had been conducted manually. The process took six weeks by a senior member of the administrative staff. In view of reserve capacity, no optimization of room scheduling was required.

Whilst there are underlying similarities between all timetabling problems in the sphere of academic applications, there is an important distinction between school and university timetabling in that in the former, classes can be treated as a unit which must be occupied throughout school time. At the university, however, there is considerable time which does not have to be allocated by the timetable, on the other hand there is the added problem of combinations of subjects taken by the students. Three principal approaches to the problem have been considered recently, they are:

1) CONFLICT MATRICES - exemplified by Cole (1964) and Wood (1968)

2) 3D BOOLEAN ARRAYS - exemplified by Almond (1965-9) and Yule (1968)

3) ITERATIVE (Linear Programming)-exemplified by Gotlieb (1963) and Lions (1966)

1) The CONFLICT MATRICES technique involves counting the number of subjects conflicting with each other subject, and then using the total number of conflicts per subject as the criterion of ordering. The timetable is then built up one stage at a time until all subjects have been scheduled.

i) Cole (1964) used this approach in the preparation of university examination timetables, subject to the constraints that no student should be required to sit more than one paper per period and the total number of periods taken should be a minimum, as well as a number of optional supplementary conditions concerned with the order of precedence of the papers and accommodation.

The basic weapon in the solution of the problem was an 'incompatibility' table in the form of a binary matrix of N x N array lists, where N is the highest subject number used. 'Incompatibility' is said to exist when two subjects are taken by the same student. The complete table of incompatibilities can be drawn up from the student/subject lists.

As each student's list of subjects is read-in; a record of incompatibilities in the form of bit markers is made in the appropriate positions of the 9 words allocated to each subject - this allocation is partially determined by the optimization of the characteristics of the machine, an Elliott 803, in which each word contains 39 bits. 8 full words and 28 bits of the ninth are required to record the incompatibilities for each subject, the spare 11 bits are used for counting (tape 2).

A check is made that there is no multiple recording of incompatibilities.

The input data requires four tapes. Tape 1 records the subjects taken by each student; this is the longest and most complicated tape, hence the incorporation of correction symbols in the input routine. The format is as follows:

$$m_1, m_2, \blacksquare \ldots\ldots\ldots\ldots * , m_p /k *$$

m_n student number, end of list marker, k (optional) number of students taking this combination - initialized at 0.

Tape 2 records three items through three integers in rows.

u subject number

v number of papers

w code for timelocation: 2 non-consecutive

1 consecutive

0 arbitrary

Tape 3 records precedence definition by using a pair of integers to indicate that A must not preecede subject B.

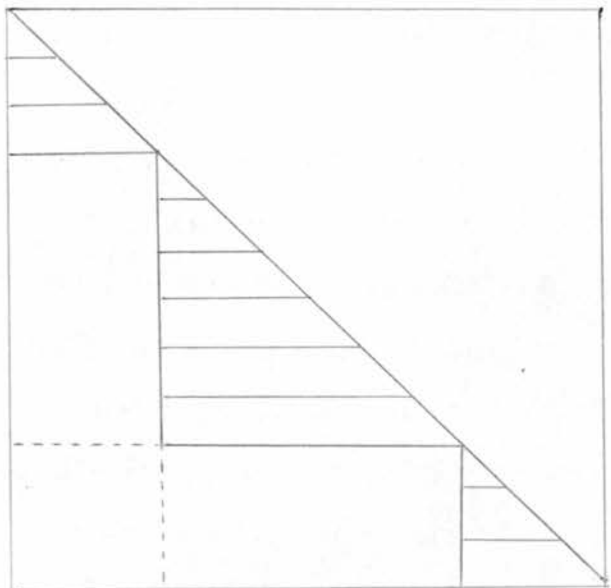
Tape 4 records period definition - the list of subjects which must appear in odd periods (a.m.)

Each tape is terminated by an agreed symbol.

The first stage in transformation of the incompatibilities matrix into a timetable is accomplished by a 'grand sort'. Cole records that experience indicates that very satisfactory ordering of subjects for subsequent allocation in timetabling is achieved on the basis of the total number of incompatibilities per subject 'p'; and sub-ordering by'consecutive' bits 'q', number of papers 'r', original ordering of subjects 's'; thus, a composite word corresponding to each subject is built up, the N such words are sorted into order of magnitude.
The second stage of the solution to the timetabling problem, the allocation of subjects to periods, consists of a search through the lists for criteria of precedence, incompatibility, and accommodation constraints. A subject which satisfies all conditions is now included in w, incompatibilities are recorded in w', the paper count is accordingly amended.
A search for the next permissible subject is made, and the process continued until subject N has been examined. The set of permissible subjects in w is now put out, each together with the current value of the paper count v'. w' is now cleared and the papers to be allocated in the next period are determined in the same way. The process is terminated when all the paper counts v have been reduced to zero.

If a subject whose papers must not be in consecutive periods appears at the end of the table, this may lead to an excessive number of periods used; this may be avoided by the intoduction of dummy subject numbers and by making the offending subject incompatible with them. This step yields a higher priority in the 'grand sort', but will not affect the other conditions.
The equipment used was the Elliott 803 of Leicester University with a 4096 word store and an addition time of 576μsec.
This enables 340 subjects to be handled with a maximum of 15 papers per subject. To process the total would take hours of processing time, but for 340 subjects with an average of $1\frac{1}{2}$ papers per subject and no condition that 'paper A must not precede paper B' the time taken 90 minutes, however the procedure for allocating 57 papers, into 14 periods, with 34 very incompatible subjects was $12\frac{1}{2}$ minutes.
The distribution of times and subjects, in the timetable produced, was bell-shaped: this type of distribution has the advantage that the subjects which have the largest number of candidates and hence prove most difficult to allocate are placed towards the beginning of the examination period. By dealing with large classes earliest, most time is allowed for

to face page 72



Partitioning of Conflict Matrix

Wood (1968)

marking the papers which yield the largest number of scripts. Cole's solution was one of the earliest applications in the U.K. of computers to the problem of academic timetabling. His work provided the foundation for others, notable among whom was Wood (1968) at Manchester.

ii) Wood's (1968) expressed aim is the generalized formulation of the timetabling problem. He uses a larger number - 1323 papers, compared with Cole's 340, and includes room locations in his scheme. His prime innovation is in circumventing the short-coming of the conflict-matrix for larger universities - the enormous amount of storage required, for example the full conflict-matrix for 1000 subjects would require a prohibitive 990,000 locations. Wood's method for circumventing this obstacle is as follows: Since the conflict matrix will be symmetrical, only the upper triangular half needs to be stored - for 1000 subjects, 499,500 locations will be required. The next step is completely pragmatic; by using the fact that the matrix may be divided by faculties (or departments) and very few candidates will take subjects from more than one faculty, there will be commensurately fewer non-zero eleements in the off-diagonal blocks: thus, if subjects which are taken in more than one faculty are repeated in each faculty in which they occur, the elements outside the diagonal blocks can be removed completely, so only the triangular diagonal blocks need be stored. These are stored in a uni-dimensional array; another array 'row' is used to point the position corresponding to the diagonal element of each row, in the same way as a 2-dimensional array is stored by the Atlas Autocode computer. Another ploy which further reduces storage requirement is the approximation of each number by a 6 bit character, this figure is chosen because one word in Atlas consists of 48bits. The maximum number which can be stored is 63, but this is sufficient to indicate that two subjects are highly incompatible.

Wood admits that the effectiveness of this manoeuvre depends upon the extent to which the subject can be partitioned into nearly conflict-free groups. In practice, the largest faculty contains over 500 subjects which cannot readily be divided, although each block of the conflict matrix can be accumulated in the store sepately from the data for the faculty.

While the construction of the timetable requires all the blocks of the conflict matrix, and in practice this would require an excessive amount of storage; since the matrix has a majority of zero elements, once the matrix has been accumulated it can be stored much more efficiently as lists. Thus, for each subject, a list is formed of the subjects with which it clashes, and the number of student conflicts. The subject number can be represented by 18 bits, the number of conflicts is already represented by 6 bits; so each conflict can be held in a half-word.

Wood notes that the average number of conflicts per subject is about 15, the tests for 1000 subjects now require only 7500 words. The conflict lists for those subjects which are required to take place simultaneously, as well as those which are repeated in more than one faculty can now be merged. The list of conflicts for each subject is printed and can prove valuable in making pre-arrangements.

The constructed timetable is constrained by three fundamental conditions:

1) No candidate is required to sit 2 examinations at the same time.
2) All candidates for a subject must be in the same room.
3) Each room has a limited seating capacity.

An extra guide-line is that the number of occasions on which candidates have two examinations in 24 hours is kept to a minimum-when necessary afternoon and following morning sessions are selected in preference to morning and afternoon of the same day. The course of the solution runs parallel with that enunciated by Cole. The ordering in which the subjects are fitted into the timetable is determined by the number of clashes which a subject has, which is regarded as a good indicator of the difficulty of finding a suitable period; in addition to the number of candidates, which restricts the rooms which are suitable.

Those subjects are first selected which must take place in the largest room, the subjects are then assigned to the timetable in descending order of the number of subject clashes. The conditions according to which assignation takes place are that subject's' must not clash with another subject already assigned to that period, there must also be a room of sufficient capacity.

The ideal choice period is one for which the subject is conflict free for both the preceding and the following periods, since this implies that no candidate for the subject has another examination which takes place in these periods. If there is more than one such period, a secondary criterion is used which aims to minimize the effect of this assignment on the choice of periods available for later subjects.

A skewed distribution of periods is sought in order to provide maximum time for marking; specifically, the highest density of periods is sought in the first 10 days of the 15 day examination diet. The effect on the timetable of varying the number of periods or rooms were studied, and some timetables were successfully computed when as few as 24 periods (12 days) were specified. This had the important practical result that there was no need to extend the examination diet, as was feared under manual operation.

If during construction, a subject is encountered which cannot be fitted into the timetable, the binary words for the subject are printed and the programme terminated. Wood reports that inspection of the binary arrays printed as the construction proceeds reveals the subjects which cause the difficulty, in which case the recommended policy is to pre-assign the difficult subjects and restart timetable construction.

As each subject is allocated to the timetable, the two binary words indicating the periods which are conflict-free and the periods for which room is available, are printed out together with the period and the room selected. This, it is claimed, 'gives a good insight into the changing situation as the construction of the timetable proceeds.'

Once computed, the timetable can be printed out as a list of subjects in numerical order with the time and place of each - from this the candidate can immediately find when and where his examinations are held. Lists are also produced showing the state of every room for that period, these enable the administrators to send the appropriate papers to each room, and can be displayed in the room to indicate the seats allocated to each subject.
Whilst the names of the candidates are not required in the construction of the timetable, by including them in the data, lists can be produced of the candidates taking each subject, as required at the time of the examination for checking purposes and later by the examiners for recording their marks.
Each symbol in the students' names is represented on input by an integer in the range 0 to 63 so that it can be stored as a 6 bit character - 3 words (24 characters) are allowed for a name. Since the lists vary for different subjects, to allocate a safe maximum area to each would have been wasteful, chain lists were therefore used whereby the items for each list are not stored consecutively, but are connected by means of a link which can be extended as required.
Wood implemented his proposals in 1967 at the University of Manchester.

The 5730 students who took 1323 different papers were allocated times and places for their examinations within 30 periods and with a maximum requirement of 18 rooms. The subjects were divided into four blocks as follows:

Economics & Education	250	subjects
Arts & Music	560	subjects
Science	399	subjects
Law, Medicine, Theology	114	subjects

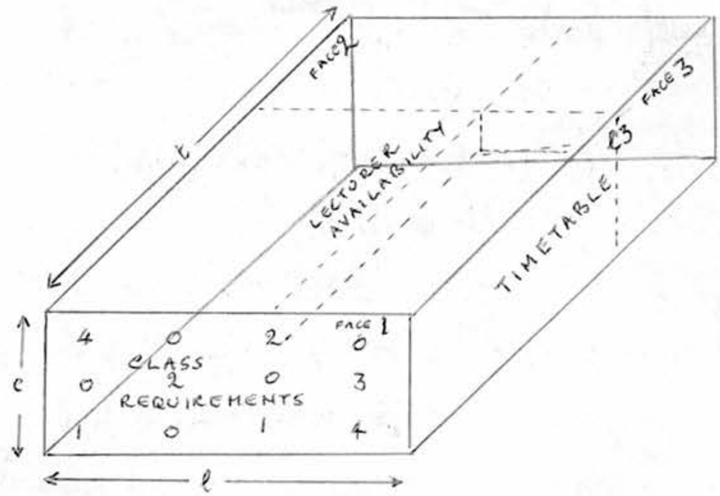
5 minutes computing time was required to transform data for each group into conflict matrices which were then re-arranged as lists.

Construction of the timetable took 10 minutes, of which half was the transfer of information between drums and core store; Atlas core store is of 16,000 words, which is considerably smaller than the area of store which was frequently accessed. The production of the name lists also required about 10 minutes computing time, the printing being performed off-line.

Positive improvements in the organization of the examinations were achieved. Notably, that over 99% of places were occupied, compared with the 85% accommodation utilization achieved when the timetable was drawn-up by hand. This had the important consequence that the marking could be begun sooner and the results became available earlier, and the important economic result as it showed that no further building for examination accommodation would be justified. Also, the distribution of candidates' examination improved; the number of occasions on which candidates had to take two examinations in 24 hours was halved; there was an even greater reduction in the number of students sitting two subjects in one day.

One cannot but recognize Wood's ingenuity in manipulating the input data of examination timetables so as to bring within bounds the storage requirements of the conflict matrix approach when applied to large numbers. On the other hand, however, one cannot agree that he has achieved his avowed aim of a 'generalized formulation of the scheduling problem;' indeed his method appears primarily pragmatic and lacks the methodological clarity, which is the distinctive feature of Almond's approach. Of course, one may argue that there are 'indivisibilities' like faculties, in any scheduling problem - no doubt this is true to a certain extent; but one cannot accept as criterial a programming solution which requires so much discretionary manual manipulation not only

<u>Almond (1965)</u>



Face 1 CLASS REQUIREMENTS MATRIX (Class x Lecturer)
number of hours lecturer 'l' is to meet class 'c' each week

Face 2 LECTURER AVAILABILITY MATRIX (Lecturer x Time)
Boolean Matrix: coefficients F (false) when lecturer is free; T (true) lecturer unavailable

Face 3 TIMETABLE MATRIX (Class x Time)
Coefficient in row 'c', Column 't' name of lecturer 'l' meeting class 'c' at time 't'

prior to computer processing, but also in response to failure in allocation, as well as depending so heavily for its viability upon a relatively high degree of compartmentalization of the data to be manipulated.

2) 3D BOOLEAN ARRAYS

iii) Almond (1965-9) expounds her method in two articles - the latter represents a generalization of the procedure outlined in the first. Both papers, however, focus upon the problem of lecture timetabling rather than the allocation of examinations. The proffered solution to the timetabling problem involves the use of three 2-dimensional Boolean arrays, which can be represented graphically as the 3 rectangular faces of a brick.

The problem is formulated thus: Given a set of lecturers available and a number of classes requiring instruction which must be brought together at certain times; these requirements may be represented in this way:

<u>face 1</u> Class Requirements Matrix (lecturer x class) the number of hours lecturer 'l' is to meet class 'c' each week.

<u>face 2</u> Lecturer Availability Matrix (lecturer x time) with coefficients F (false) when a lecturer is free, T (true) when lecturer is unavailable.

<u>face 3</u> Timetable (class x time) the coefficient in row 'c' column is the name of lecturer 'l' meeting class 'c' at time 't'.

Initially, the matrix on face 3 is null: however, by successive up-dating of the class requirements matrix (face 1) through the allocation to each subject of a suitable lecture hour (face 2) the matrix in face 3 is transformed from the null matrix to the solution - the timetable.

The algorithm dictates firstly, consideration of all the entries in the requirements matrix one by one, and secondly, allocation to each of a suitable lecture hour. Different versions of the timetable may be obtained by scanning the requirements matrix in different directions.

The method involves three procedures:

1) Allocation
2) Conditions Alteration
3) Copying of Initial Matrices

1) Allocation searches for a suitable lecture time which avoids hours already filled and satisfies the preconditions.

The ten pre-conditions which were satisfied by the results were for the most part met by entries in the initial Lecturer Availability Matrix (L.A.M.) or by the Allocation Procedure (A.P.).

These pre-conditions may be grouped into three categories:

i) <u>externally imposed fixed pre-conditions</u>, lecturers with classes at fixed times in other departments (L.A.M.), undergraduates have no lectures on Wednesday afternoon (A.P.)

ii) <u>externally imposed movable pre-conditions</u>, all members of staff have one free day each week (L.A.M.) lectures to postgraduates last for two consecutive hours and can begin at only two possible times - 10.30 a.m. or 2.30 p.m. (A.P.)

iii) <u>pre-conditions imposed by preferences</u>, all lectures should be given in the morning in preference to the afternoon (this is achieved by numbering the hours of the week wo that the morning hours are always filled first), senior members of staff do not like 9.30 a.m. lectures (L.A.M.), a lecturer should not meet the same undergraduate class twice in any half-day, but morning and afternoon meetings are acceptable (A.P.), if possible no one should be asked to lecture for three consecutivehours (A.P.), lecturers may ask for extra free hours (L.A.M.).

There is the additional condition that classes are split into two or three groups for exercises.

2) Alter Conditions procedure is called in when an allocation fails and carries out a series of manoeuvres in an attempt to find a solution. Restarting with different conditions after an allocation failure represents an important advance over the method used by Wood which required human intervention in this eventuality: though it must be admitted that Almond has more variable parameters. The classes are reordered so that the one proving most difficult will be inserted in a blank timetable, if this is unsuccessful various preconditions are waived in reverse order of priority. If this manoeuvre fails, the procedure only now prints a postmortem and brings the programme to a halt.

3) Copy Initial Matrices. After an alteration in conditions, the Initial Requirements and Initial Lecturer Availability matrices are

recopied into the Current Requirements and the Current Lecturer Availability matrices. The timetable matrix is made null, the allocation procedure is then re-entered.

The second application was the construction of a faculty timetable rather than the timetable for a single department, as in the first application. The methodology remains the same. Face 1 now represents a course requirements matrix. Face 2 still deals with lecturer availability, and the resultant timetable is still to be found on Face 3. Additionally, a 2 dimensional Boolean array - the conflicts matrix, which lists groups of courses whose lecture times must not conflict is included in order to allow students to choose from certain subjects. The number of lectures allocated at each hour can be limited by the number of lecture theatres available.

The information generated by the production of the timetable was used to produce lists of lectures which take place at certain times as well as lists of all possible combinations of the three subject course requirement for each of a student's first four semesters. This was extended by an extra programme which took as input data the 4 lists of 3 courses and also any prerequisite course in the semesters 2 and 4, and produces a list of all possible groups of 12 courses which a student could study during his first

4 semesters, using the given timetable; this is in line with Holz (1963) procedure at M.I.T.

Almond's (1965) work was used by Yule (1968) as a basis from which to extend the scope of her method. He proposed the replacement of Class Requirements and Timetable Matrices by a File of Requirement lines in order to overcome the assumptions that all lectures are of the same length, and each lecture is given by one lecturer to one class. He also seeks to deal with the problems posed by the non-availability of classes, lecturers and rooms as well as the allocation of rooms. The spread of lectures over the week, a limitation on the lecturers' daily load, as well as a procedure to deal with impossible solutions also claim his attention. His proposals were implemented at Sussex University and Brighton College of Technology. Processing was conducted on an I.C.T. 301.

Apparently in reply to Yule, in 1969 Almond published her second paper which proposed to offer a co-ordinated timetable for the whole faculty incorporating lecture room allocation and timetabling of later years. The availability of lecturers is not considered directly because of the negligible change in courses from year to year, which, in any case, are independent of staff changes.

Almond also suggests that on registration day the computer could be used to check that each student selects compatible courses and to ensure that classes and laboratories do not get overfull. The printing of individual timetables for each student is also proposed as a basis for a conflict matrix which could be used to produce examination timetables.

The programmes JT74 or JT74XP could be used for the collection of the data required. The factor frequency count facility in JT74 already compiled totals of various categories of students as well as offering the display of lists of students falling within these categories.

Almond, in concert with Wood, finds it more economical to store the conflict matrix in the form of a list; this offers the added advantage that the position in the list can indicate the priority of the non-conflict as well as giving added room for manoeuvre in the allocation process. Now that Almond too is confronted with the more complex problem of timetabling a larger administrative unit; she resorts to the pre-ordering of courses which may prove very difficult to allocate. This pre-ordering is conducted according to identifiable ordering associated with a weight factor.

This weight factor is associated with each course and the courses are then sorted in order of decreasing weight. Mention as possible items for inclusion in the weight factor are:

i) number of essential non-conflicting courses

ii) number of students

iii) number of lecture and laboratory hours

These items can be combined in various ways to produce suitable weight factors and different orderings of the courses for allocation will result in alternative versions of the timetable. Alternative timetables can also be generated by including different lists of possible starting times in order of preference for single periods and multiple periods of various lengths.

The algorithm searches through the times in the given order until one is found which satisfies all the desired restrictions on room occupancy, non-conflict, even spread of the course through the week. If no suitable time can be found, the desirable, but non-essential no-conflict restrictions can be relaxed one by one, beginning with the least important. After a restriction is relaxed then the times of the week are again tried in order, in the hope that a suitable one will be found. Essential no-conflict restrictions are not relaxed but an omission message is printed and the programme passes to the next period. A note is kept of the number of omissions and neglected non-conflict restrictions for each course and these numbers can be introduced into the weighting factor for a second attempt, in which the courses are relisted in a new order which gives priority to those which proved difficult, and a second timetable is produced.

Almond recommends that the choice of timetable should be the result of a survey of several timetables; the optimum timetable is described as the one which offers 'best possible selection of combinations of courses for students.' Whilst it is desirable that an optimum should be sought, especially when the criterion is based upon consumer acceptability; by leaving the choice of an optimum to the discretion of a human administrator, this indicates that the criteria of excellence in university timetabling have not been precisely formulated. Had this been achieved, the computer itself could easily have ordered the timetables produced according to the pre-determined criteria. This step would not only have brought consistent criteria to bear in the evaluation of a number of products but would have helped to crystallize the definition of the optimum university timetable. Once defined, it would become very much easier to specify requirements which could be expected to bring about a closer approach to this aim; of course, one may contend that each university has individual requirements, and to a certain extent this is undeniable, but one cannot deny that above the individual characteristics of universities there are criteria of pupil-teacher ratios, optimum times of the day for learning, and an optimum spacing of instruction and examination periods which transcend these individual requirements. Once criteria such as these have been concretized and incorporated into a list of criteria then the most prudent course to be taken by the timetabler would become clearer.

3) ITERATIVE (Linear Programming)

The 3 dimensional array also formed the basis of the approach used by Gotlieb (1963) in the generation of school timetables. Whilst designed to cater for a different educational level, there are many features of similarity in organization between the two approaches.

Each point of the 3 dimensional array, represents the meeting of a particular class, with a particular teacher, at a particular hour of the day. Array elements are divided into 2 groups 0 or (Non-Zero)1. Zero signifies the impossibility of the class and teacher meeting at that hour; Non-zero signifies the possibility that the class and teacher may meet at that hour. Initially, the third array is filled with 1 to indicate the possibility that any teacher can meet any class at that hour. It is recalled that in Almond the matrix on face 3 is initially nul and that only as the solution emerges is the possibility of specific meetings mentioned. As Gotlieb's calculations proceeds 1 is transformed to 0 according to set rules and predetermined conditions so that by the end of the computation, at each hour it is possible for each teacher to meet only one class and for each class to meet only one teacher and each teacher can meet each class a number of times predetermined for that teacher and that class. The timetable, as with Almond, is then inherent in the resulting 3 dimensional array. A notable point is that Gotlieb claims that each array is effectively square in that any seeming excess of rows or columns can always be eliminated; this contrasts with Almond who visulises a brick.

The examination of the n x n arrays in the course of timetable construction has two stages;

i) Feasibility Test - the existence of at least one possible schedule is determined (Assignment problem).

ii) Matrix Reduction - any nonzero element which does not belong to some possible schedule is changed to zero.

Provided stage i has been successful, stage ii can be entered.
For stage ii Gotlieb proposed the method of the 'tight-set search', whose theoretical basis is a theorem by Hall (1935) on the existence of sets of distinct representatives. The tight-set search is quite efficient when n is small ($n \leq 10$), but since it requires of the order of 2^n operations, it rapidly becomes impractical once n exceeds about twenty. Many schools have over 50 classes and up to 100 are not unknown - this is certainly true of universities.
Lions (1966) therefore proposed the use of the Hungarian Method for Feasibility Test which involves the Iterative Application of procedure called 'Expand'. Procedure Expand searches the linkages between rows and columns and may be regarded as developing a tree structure which eventually shows a minimum path between the designated row of the array which had been found to have no solution element, and some column of the array which has no solution element.
The minimum path is then retraced to rearrange the original partial solution to create the new partial solution. If 'Expand' fails to generate a new partial solution, then it can be shown that it will never be possible to find a solution to the array; although it may be possible to find a partial solution by choosing another row. Any row or column with a solution element in the first partial solution is

certainly represented in the second.

A partial solution for the array is given in the vector solution whose Jth element gives the row number of the solution element in column J if one exists or else is blank. Two working areas are needed: the vectors 'reference' and 'rowlist' which are initially cleared to blanks. Rowlist contains a list of rows to be searched on a first-in-first-served basis. The first member of rowlist is the designated row into which Expand is to intoduce a solution element. The procedure may be divided into two phases:

i) forwards phase - the searching of various rows selected from rowlist.

ii) backwards phase - the retracing and rearranging of the partial solution.

When an array row is searched, its elements are examined one by one. The first phase of Expand is complete when either a non-zero element is encountered in a column J for which solution J is blank, all rows entered in rowlist have been searched but no element satisfying the first condition has been encountered. As soon as the first condition is fulfilled, the existence is demonstrated of the next partial solution which is actually constructed by the second phase of Expand. If the second condition occurs, then the task of

Expand is not feasible.

In the above we have seen the operational definition of a mathematically optimum solution to a timetabling problem: 'a minimum path between a designated row of the array and some column of the array which has no solution element.' This is the first time in this survey that we have found a solution which has been so clearly optimal according to some fixed criterion, although Cole (1964) does mention that his method will generate a solution which is 'minimal' in the sense that the examinations could not be allocated in less time periods.

Another quest for an optimum, this time from the point of view of the administrator, is made by Lawrie (1969) who proposes the 'replacement of an objective function by the production of a number of alternative integer solutions as having more relevance to the criteria normally quoted by headmasters as constituting a "good" timetable.' He repudiates the 'micro' approach adopted by other workers who deal in units of the teacher, the class, and the event; as a proponent of this approach he cites Lions (1967). Instead, he proposes a 'macro' approach formulated in terms of such considerations as the distribution over the week of classes in various subjects, the distribution over the week of free periods for staff/pupils, the mix of periods each day between academic and minority time subjects. The method by which Lawrie seeks to achieve this laudable aim is by the use of between four and six 'layouts'. A 'layout' is defined as a statement of the curriculum and its organization for a group of pupils, generally a year group. It is claimed that this method, which is designed to allow a headmaster to express his requirements in the form of layouts, offers important administrative advantages in that details can be suppressed, the curriculum can be planned as a year unit, even in large schools, by providing a compact notation which emphasises the staff requirements by department and omits explicit reference to classes or how pupils are regrouped into sets from period to period. In this way the educational policy of the school can be drawn in broad brush-strokes.

Once layouts have been obtained, the problem of constructing the timetable is dealt with in two stages:

i) Layouts are overlapped with one another in such a way that staffing restrictions are not violated during any period of the week - this results in an 'outline timetable'.

ii) The permutation of outline timetable to meet the requirements on distribution of classes and sets in various subjects over the week.

The problem is visualized as the finding of as many arrangements as there are periods in the week, though they need not necessarily all be different, from these the best is selected.

At first we were impressed by the fact that someone had sought to incorporate into timetable construction such important, but often neglected, features as the distribution of classes through the week, as well as the mix of periods within the day. We saw Lawrie's search for an optimum based upon 'macro' considerations, unimpeded by minutiae, as an innovation which promised a timetable which was both efficient and humane. Unfortunately, however, the result meets neither of these promises. Whilst it may be bold to dismiss the maximization of an objective function, one is also dismissing a valuable impartial criterion of the technical efficiency of the product; when one reads that one of the conditions of the atrangement of the layout is that the total of staff by department to be allocated to teaching duties by his method is <u>less</u> than the teaching staff available, one's worst suspicions are confirmed -

technical optimum is sacrificed from the outset. When one turns to the laudable humane aims, outlined above, one is equally disappointed; the 'educational policy' of the school is ultimately nothing more rigorous than the arbitary will of the headmaster. Now, it is undeniable that headmasters are men of considerable experience, and acumen; but it is impossible for a computer scientist to accept that the opinion of a man, unsubstantiated by logically coherent reasons, is the 'best' judge of the 'best' timetable, as Lawrie would have us accept. This view is, in fact, tacitly admitted by Lawrie, and the others who accept the policy of the production of many timetables from which a man is to make a choice; this policy is an admission that the criteria by which the timetable is to be judged have not been defined in impersonal terms. Whilst one may waive the requirement of the maximization of an objective function of technical efficiency, if it can be demonstrated that the 'macro' approach does maximize something, and that the programme is demonstrably optimal in some other respect. This has not been achieved in this case; Lawrie's proposal, therefore, cannot be regarded as a 'success', let alone as the 'best' solution to the timetabling problem. He has only focussed attention upon an acknowledged problem.

Our considerations of the proposed solutions to the academic timetabling problem has ranged over school teaching, university lecturing and university examinations; but has shown the underlying necessity of a conflict matrix of one type or another in the quest for any solution to this problem. Cole uses the matrix and shows its method of working. Wood seeks to circumvent the very large storage requirements of the matrix by the elimination from consideration of non-conflicting areas. Almond and Yule, as well as Gotlieb and Lions ultimately use matrices. Lawrie's 'overlapping of layouts' to produce an 'outline timetable' is a conflict matrix by another name. The problem facing educational administrators so far as the matrix is concerned, is the handling of the large storage requirements for its representation - the problem becomes acute when large numbers of classes, lectures, or examination papers have to be dealt with. In this respect one may contrast the approaches of Lawrie and Wood. The latter sought to bring the matrix within bounds by the elimination of redundant value-elements, from the programming point of view; and by the use of auxiliary equipment to expand the computer's storage capability, in the technical sphere. He reports that whilst the core store of the Atlas used was <u>16,000</u> words, 'this was considerably smaller than the area of store which was frequently accessed' - this expansion was achieved by the use of drums. The drawback of this method was that half the time takento construct the timetable, ten minutes, was taken up by the transfer of information between drums and core store - this was a waste. Lawrie, by contrast, side-steped the problem by considering a method unencumbered by 'details'; in effect this rendered the size of the conflict matrix manageable but failed to exploit the full potentialities of computers even in their number-crunching role. By leaving so much of the timetabling to the human operator, he does not make use of the decision making capability of computers in the administrative sphere - this is a waste not only of time but also of available facilities.

From the purely pragmatic point of view, it may well be that with the development of computers with ever larger storage capacities the aim of circumventing the problems posed by the handling of large matrices which have confronted timetablers to date will yield under the weight of technical progress; though of course, manoeuvres such as Wood's for the elimination of redunant elements, and proposals such as Lion's for the implementation of more efficient processing technique in the mathematical sphere will be valued in further expanding the scope and complexity of timetabling problems which may profitably be handled by computers.

In our review, we have considered the proposed solutions according to the criterion of efficiency, interpreted in terms of the computing method used and the solution produced. We noted in the linear programming solution that if this metbod did not yield a solution to the problem then no solution was possible, in addition to the fact that this method maximized an objective function and so yielded a 'minimum path' solution - here was a method which yielded an unequivocal optimum according to a concrete criterian; and contrasted this approach with Lawrie's. Consideration of the solution offered inevitably involves an examination of its acceptability. At the moment, as we have already seen, no objective definition of the optimum timetable has been evolved; all rely upon gaining a good opinion of those whose job it is to decide between the approaches. Appeal can be based upon two grounds - economic, or social. Almond advocates a socially orientated approach, expressed by her regarding a solution as good as one which offers 'the best possible selection of combinations of courses for students'. This consumer orientated approach is parallel to Lawrie's which offered a solution designed to cater for the wishes of the administrator of the organization - the headmaster - who is left to conduct the complex task of balancing the demands of staff and students in terms of 'educational policy' and on this basis he is to judge the worth of a timetable. The 'economic' approach, which seeks the optimum allocation of scarce resources, has already been alluded to with respect to Lions. This approach is also exemplified by Cole who reminds us of the fact that he achieved the solution of a timetabling problem for so many on so small a machine. Wood echoes this by his stretching of the facilities of the C.P.U. and included in his claim for approbation the fact that he produced a timetable which achieved a higher degree of room occupancy. This improved allocation resources had the important financial consequence that the university was saved from further capital investment in additional accommodation for examination purposes. He also incorporated a consumer orientation in his solution; he produced lists to facilitate administration, he spaced examinations for the benefit of students and ordered examinations to allow staff the maximum time for marking - this last development was foreshadowed by Cole.

It appears that Wood's applied approach may be regarded as most nearly realizing the quest for an orchestration of these two approaches into an effecient and humane solution. In the future this quest could be guided towards its goal by greater use of the decision taking ability of computers. It appears that some of the hesitation in formulating a definition of criteria of excellence in timetabling may have been caused by the difficulty in handling the multiple ramifications of these approaches; perhaps the use of Game Theory on the von Neumann & Morgenstern model may prove useful. Once formulated, existing techniques could be more systematically evaluated and subsequently improved. Given advances in programme techniques and improved C.P.U. capability, a greater number and variety of variables could be incorporated into more sophisticated programmes, which in turn could be more effectively evaluated and improved - in this way real progress could be achieved in the 'temporal ordering of events subject to constraints'.

7. CONCLUSION

We stated that this project sought to lay the foundations for an integrated system of data processing. We recall also that data is processed in order to:

1) keep detailed facts for records
2) produce operating documents
3) analyse facts

We have exemplified these three activities in applications within the sphere of university administration. In all three programmes facts are stored, either about students or about bibliographies. As yet, no operating documents of the class list type have resulted from the two earlier timetables because of limitation of facilities. Lists in the form of select bibliographies are produced by the archiving programme, JT74 will display on the consoles lists of students who possess certain attributes - this employs the same principles as are involved in the production of hard copy lists; with the wider availability of remote access consoles, displays may well replace the circulation of hard copy lists.

The same programme also illustrates the analysis of the information stored into 'informative reports' it is apposite to cite the factor frequency count facility as an elementary statistical survey.

The programme irreguelrities log illustrated the way in which improvements in operation can be incorporated through observing and seeking to rectify past errors.

Data processing involved:

1) ORIGINATION

2) PROCESSING

3) OUTPUT

The collection of data for conversion into information has been manifestly illustrated. The development of the conversational mode programmes which collect the three possible types of data in proportions specified by the requirements of the situation, forms the back-bone of this project. Verification facilitates economical storage, as in the case of categorized data; in the case of personal data verification is conducted by the person less qualified, the respondent. The collection of data with card input is illustrated in the archiving programme. The two aspects of processing, rearrangement and file manipulation are also illustrated. The former is the express purpose of the archiving programme and of the selective display facilities in JT74. File processing in the form of statistical analysis is handled by the factor frequency count.

Two of the three types of output are illustrated. 'Historical reports' could cover virtually all the output from the first two programmes. No forecasts have been made. 'Action documents' could describe the output of the archiving programme as well as the selective displays in JT74.

From a technical point of view, this project has foccussed upon the storage and retrieval of information. Storage has almost always been of the indexed sequential type, as already discussed. FORTRAN has been used as the method of communication because of its suitability to use on IBM equipment and also because of its wide applicability. The two earlier programmes were developed under FORTRAN IID, the archiving programme does incorporate features of FORTRAN IV in the display routine. The method of pattern matching and of file organization employed has already been discussed.

It emerges, therefore, that a combination of the techniques described coupled with an extension to system facilities could take over the task of student record-keeping. Provision has been made for the collection, storage, manipulation, and display of information. The archiving programme could provide a valuable service in the library and the tutorial room, where users could be supplied with complete bibliographies on chosen subjects with negligeble delay and a degree of completeness dependent only upon the zeal of the person whose job it is to compile the bibliography at the outset. This quest to group items which possess common features would of course be applicable to the student selection procedure.

Besides their immediate applications, these programmes can offer v valuable information which can be instrumental in the further computerization of university affairs. The former programmes can already collect information necessary for the construction of university teaching and examination timetables. Although the data may be collected in new ways, through the remote access consoles for example, the subsequent analysis is no different from that required for information obtained by more traditional methods. Although the descriptions have been directed specifically towards university applications, many others spring to mind; the interrogation routines in the former programmes could be applied to the collection of data in many spheres, onemay mention passport control, census returns as well as the non-accounting section tax returns. The archiving programme would of course be applicable to the cross referencing of criminal records.

As has been mentioned, the scope of the above programmes, especially those involving on-line operations has been circumscribed by the limitations of the system. One of the most valuable potential extensions, the operation of which has only been simulated in the generalized data processing programmes, is the 'arbiter' facility. This innovation would involve the keying in of data by typists at many consoles, when a reply was entered which was other than a mere keying error, the reply could readily be passed to the 'arbiter' who would give an authoritative classification of the difficult reply. Processing efficiency would be increased by the elimination of incorrect inputs, savings in wages would be achieved as only one person of 'expert' status would be required to handle the whole process of data collection - this same person may also be the same as the one who supervises the processing and the output production.

This project has outlined some of the possible applications of computers in a university, and has exemplified the manner in which the basic techniques of storage and retrieval can be adapted in order to meet these requirements.

Select Bibliography

Almond, M. Algorithm for Constructing University Timetables

Computer Journal Vol. 8 1965-6 p 331 ff

Almond, M. A University Faculty Timetable

Computer Journal Vol. 12 1969 p 215 ff

Barraclough, E. D. The Application of a Digital Computer to the Construction of Timetables

Computer Journal Vol. 8 1965 p 136 ff

Caskey, M.B.E. Major W.S. Accounting for the Soldier's Pay

Computer Journal Vol. 5 1962 p 258 ff

Cole, A. J. The Preparation of Examination Timetables Using a Small Store Computer

Computer Journal Vol. 7 1964 p 117 ff

Cress, P., Dirksen, P., Graham, W. Fortran IV with Watfor

Prentice - Hall 1968

Gregory, R. H. & van Horn, R. L. Automatic Data Processing Systems. Chatto & Windus 1963

IBM System/360 Operating System - Data Management

IBM Operating System/360 - Concepts and Facilities

IBM/ System/360 - Fortran IV Language

IBM System/360 Model 44 Programming System - Guide to System Use for Fortran Programmes

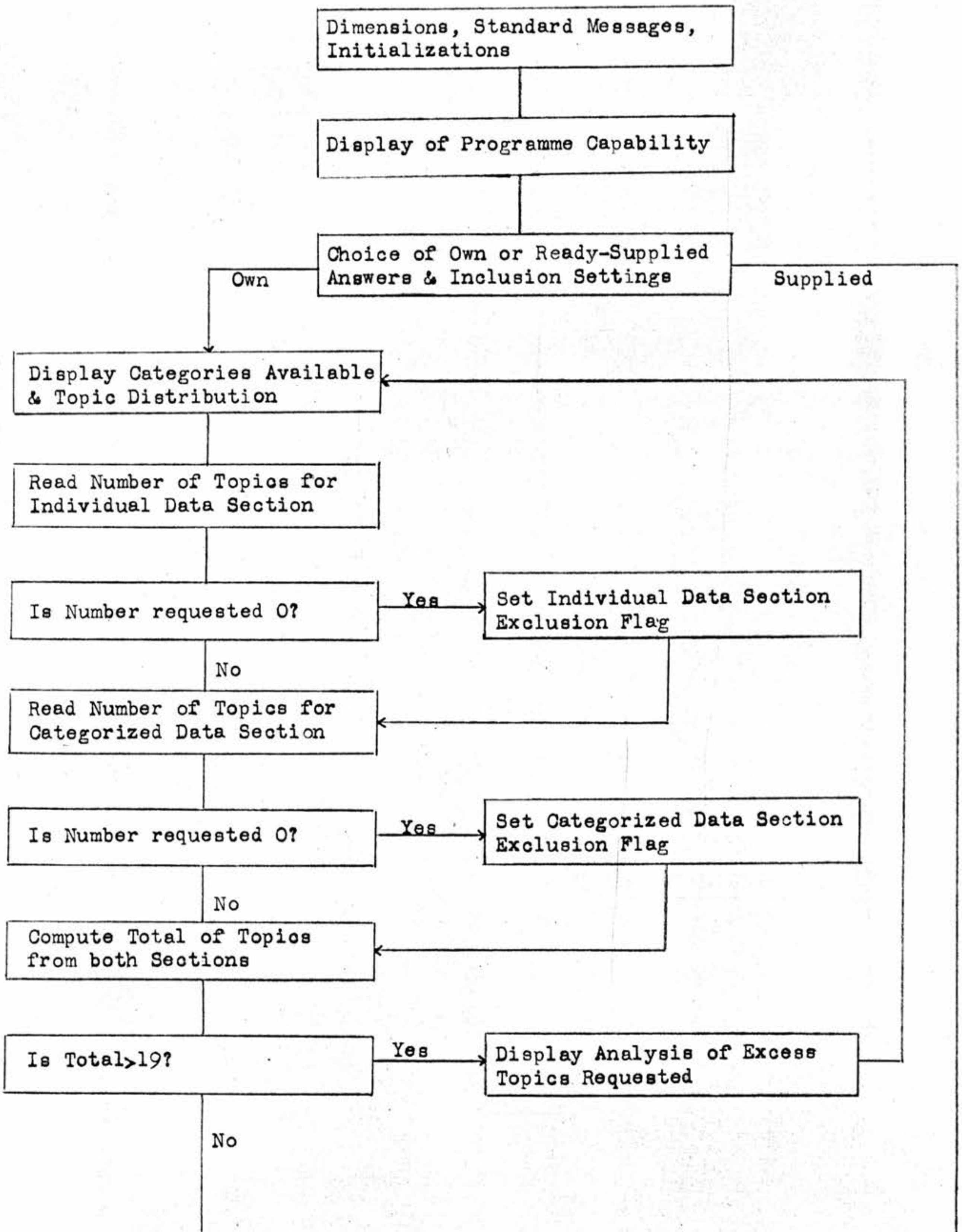
Karplus, W.J. On-Line Computing - McGraw - Hill 1967

Lions, J. Matrix Reduction Using the Hungarian Method for the Generation of School Timetables.
Communications of the Association for Computing Machinery
Vol. 9 1966 p 349 ff

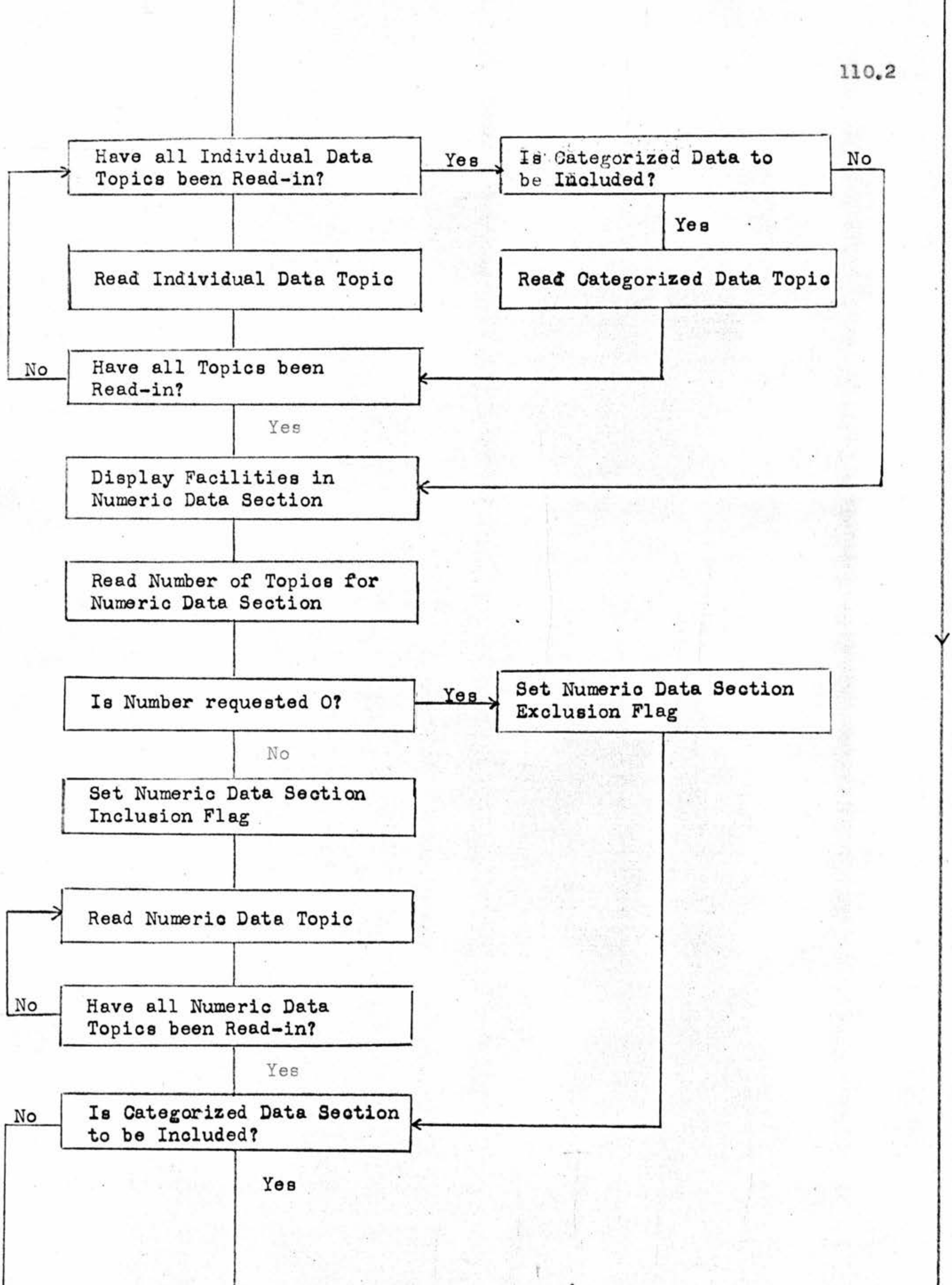
Wood, D.C. A System for Computing University Timetables
Computer Journal Vol. 11 1968 p 41 ff

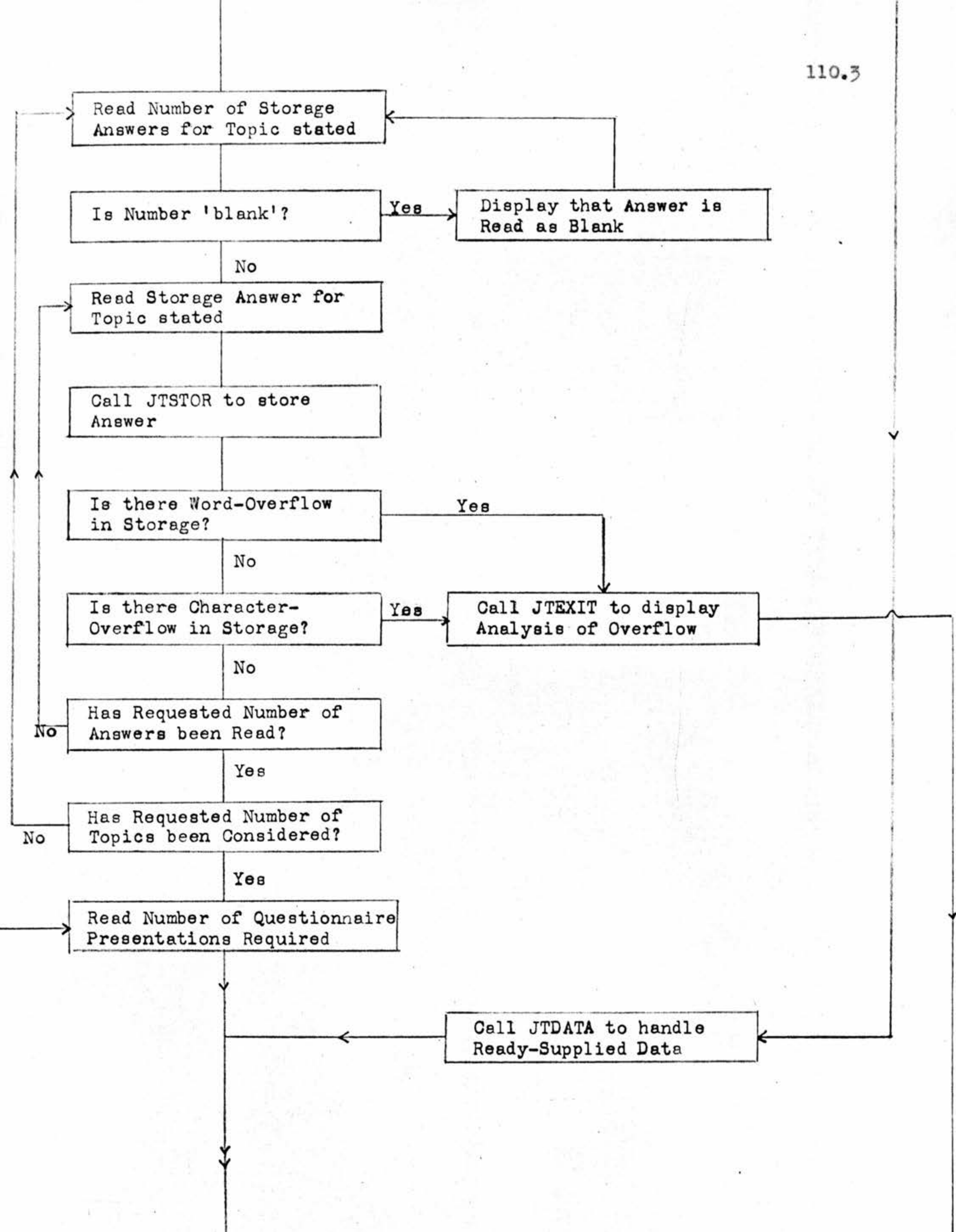
APPENDIX 1

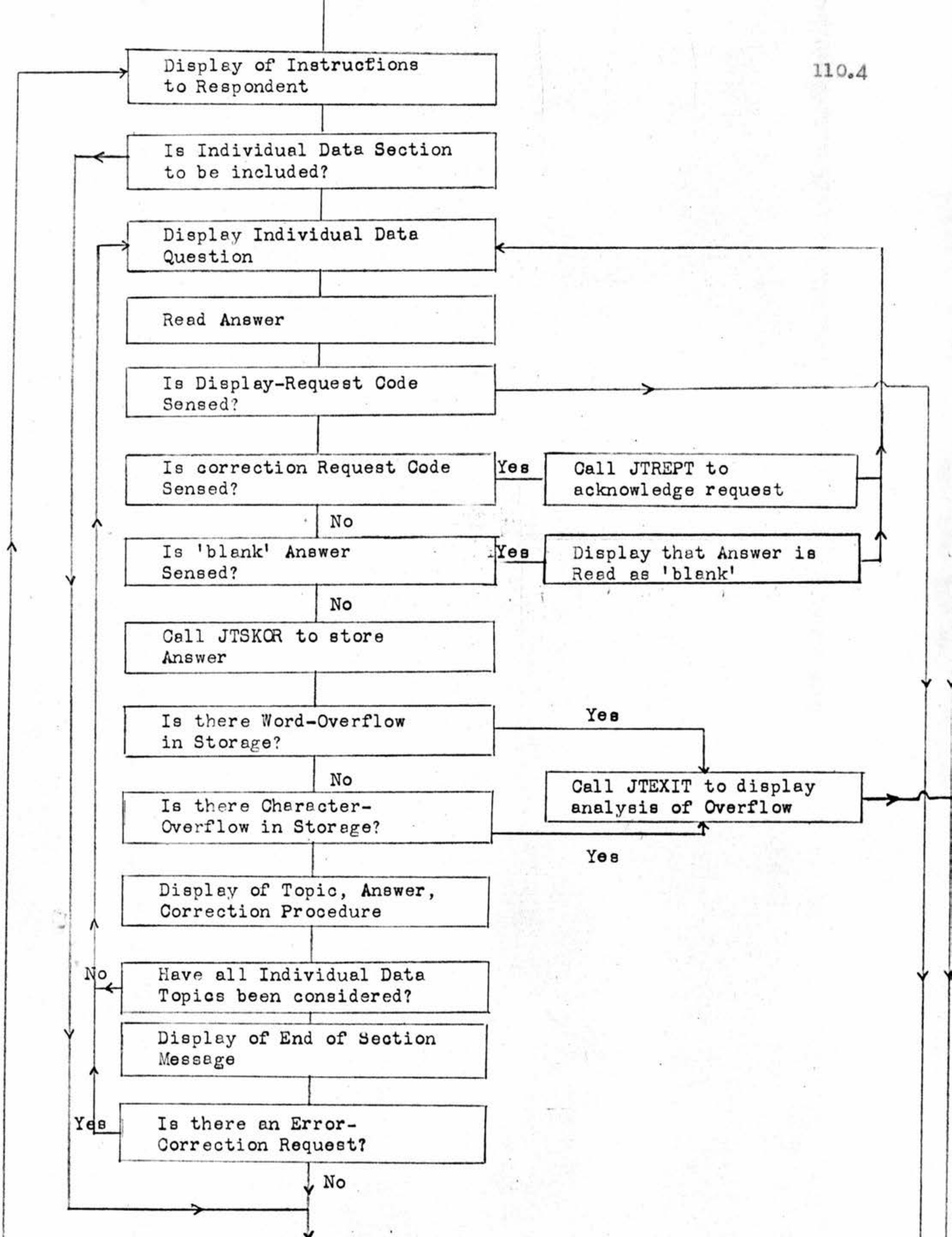
i) Block Diagram of the Generalized Data Processing Programme JT74XP



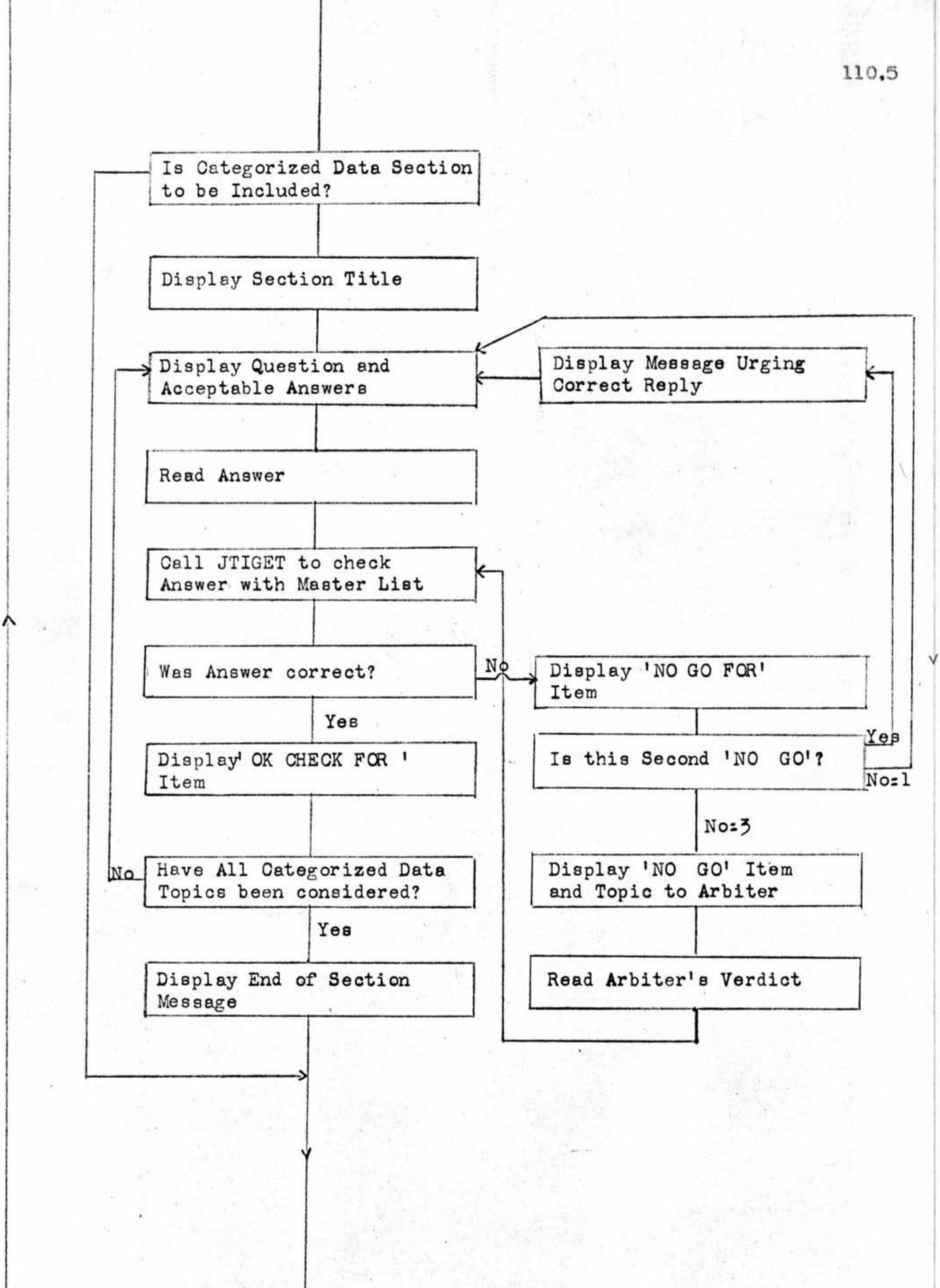
Dimensions, Standard Messages, Initializations
Display of Programme Capability
Choice of Own or Ready-Supplied Answers & Inclusion Settings
Own
Supplied
Display Categories Available & Topic Distribution
Read Number of Topics for Individual Data Section
Is Number requested 0?
Yes
Set Individual Data Section Exclusion Flag
No
Read Number of Topics for Categorized Data Section
Is Number requested 0?
Yes
Set Categorized Data Section Exclusion Flag
No
Compute Total of Topics from both Sections
Is Total>19?
Yes
Display Analysis of Excess Topics Requested
No

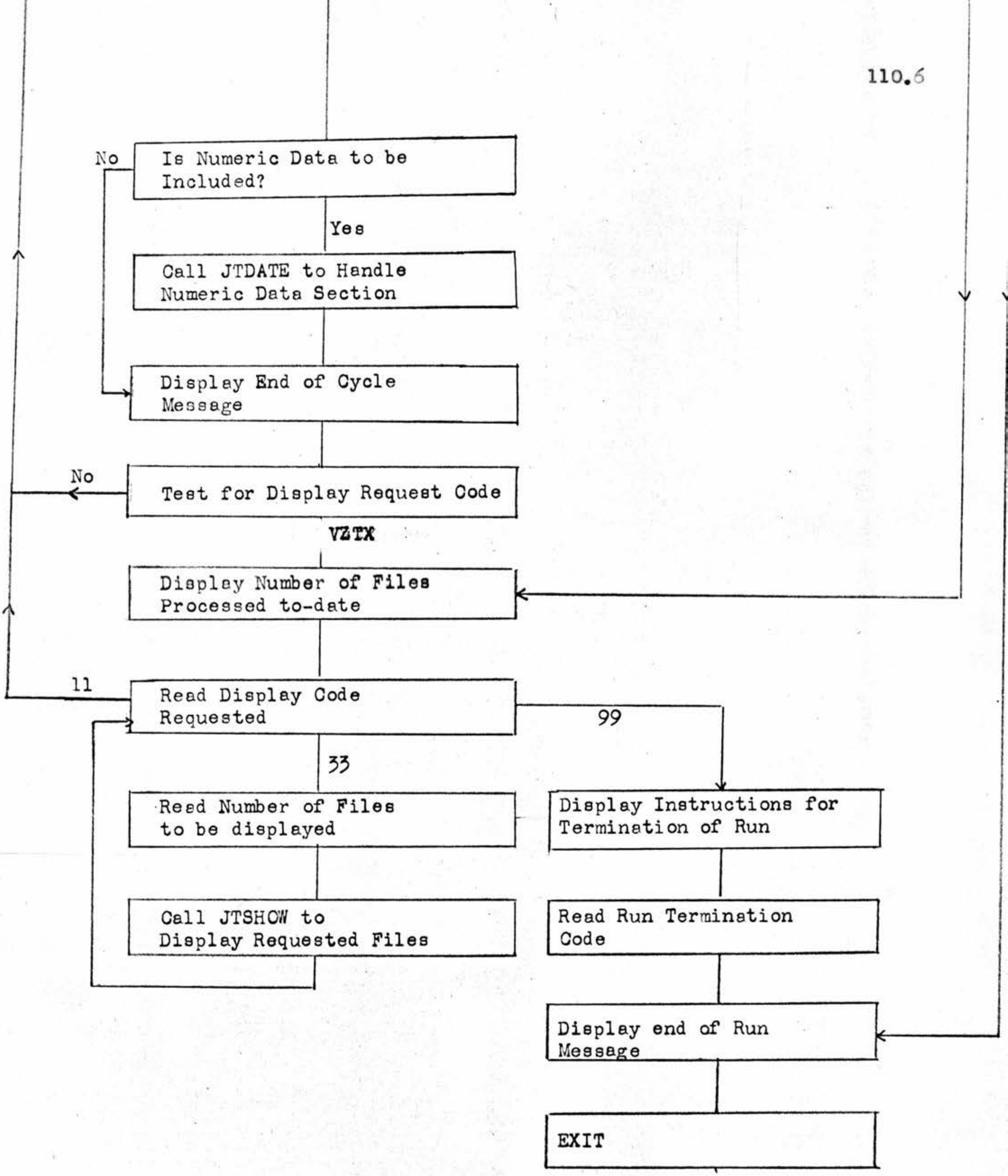

Have all Individual Data Topics been Read-in?
Yes
Is Categorized Data to be Included?
No
Yes
Read Individual Data Topic
Read Categorized Data Topic
No
Have all Topics been Read-in?
Yes
Display Facilities in Numeric Data Section
Read Number of Topics for Numeric Data Section
Is Number requested 0?
Yes
Set Numeric Data Section Exclusion Flag
No
Set Numeric Data Section Inclusion Flag
Read Numeric Data Topic
No
Have all Numeric Data Topics been Read-in?
Yes
No
Is Categorized Data Section to be Included?
Yes


Read Number of Storage Answers for Topic stated
Is Number 'blank'?
Yes
Display that Answer is Read as Blank
No
Read Storage Answer for Topic stated
Call JTSTOR to store Answer
Is there Word-Overflow in Storage?
Yes
No
Is there Character-Overflow in Storage?
Yes
Call JTEXIT to display Analysis of Overflow
No
Has Requested Number of Answers been Read?
No
Yes
Has Requested Number of Topics been Considered?
No
Yes
Read Number of Questionnaire Presentations Required
Call JTDATA to handle Ready-Supplied Data

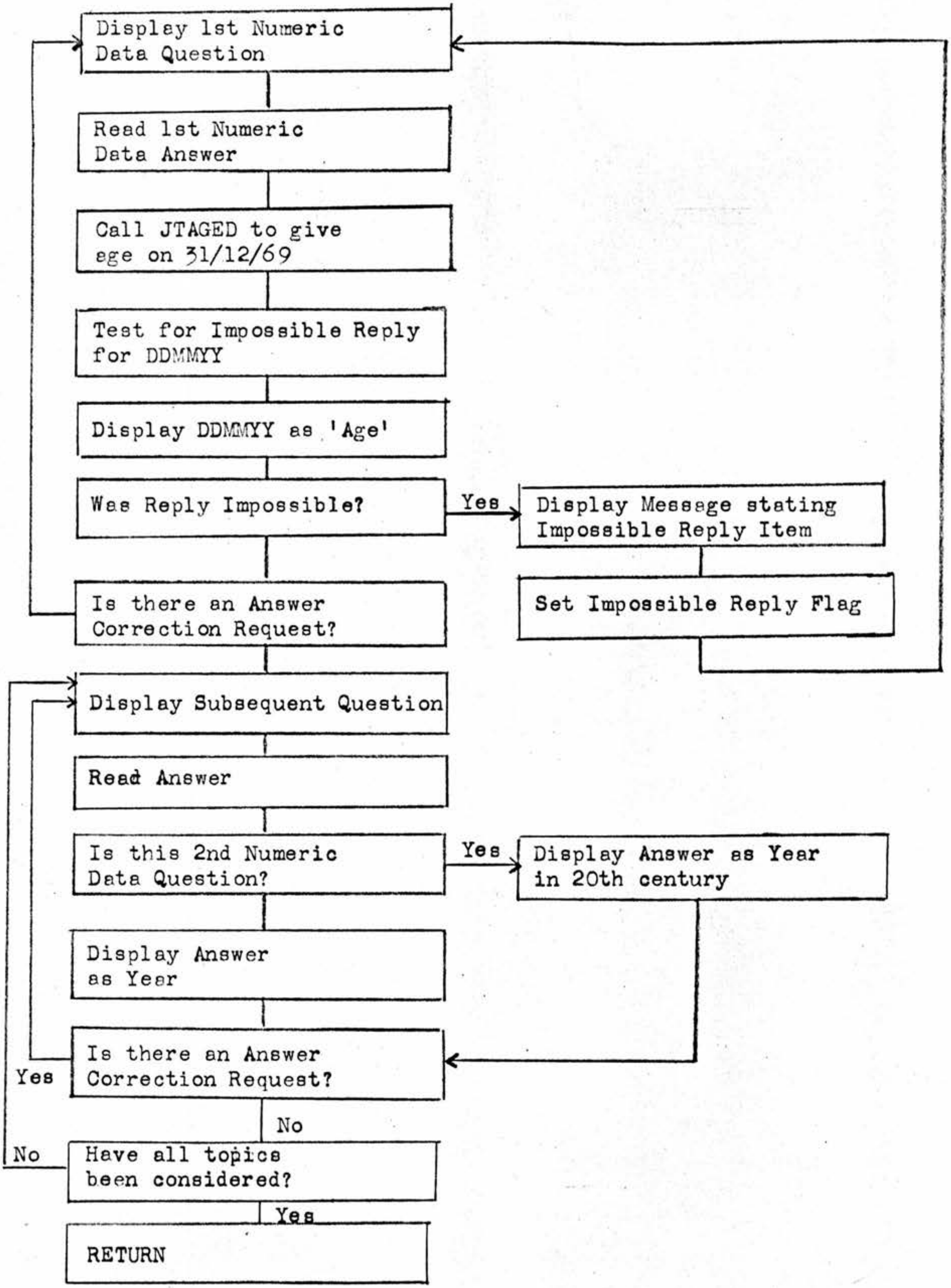


Display of Instructions to Respondent
Is Individual Data Section to be included?
Display Individual Data Question
Read Answer
Is Display-Request Code Sensed?
Is correction Request Code Sensed?
Yes
Call JTREPT to acknowledge request
No
Is 'blank' Answer Sensed?
Yes
Display that Answer is Read as 'blank'
No
Call JTSKOR to store Answer
Is there Word-Overflow in Storage?
Yes
No
Call JTEXIT to display analysis of Overflow
Is there Character-Overflow in Storage?
Yes
Display of Topic, Answer, Correction Procedure
No
Have all Individual Data Topics been considered?
Display of End of Section Message
Yes
Is there an Error-Correction Request?
No


Is Categorized Data Section to be Included?
Display Section Title
Display Question and Acceptable Answers
Display Message Urging Correct Reply
Read Answer
Call JTIGET to check Answer with Master List
Was Answer correct?
No
Display 'NO GO FOR' Item
Yes
Display' OK CHECK FOR ' Item
Is this Second 'NO GO'?
Yes
No:1
No:3
No
Have All Categorized Data Topics been considered?
Display 'NO GO' Item and Topic to Arbiter
Yes
Display End of Section Message
Read Arbiter's Verdict

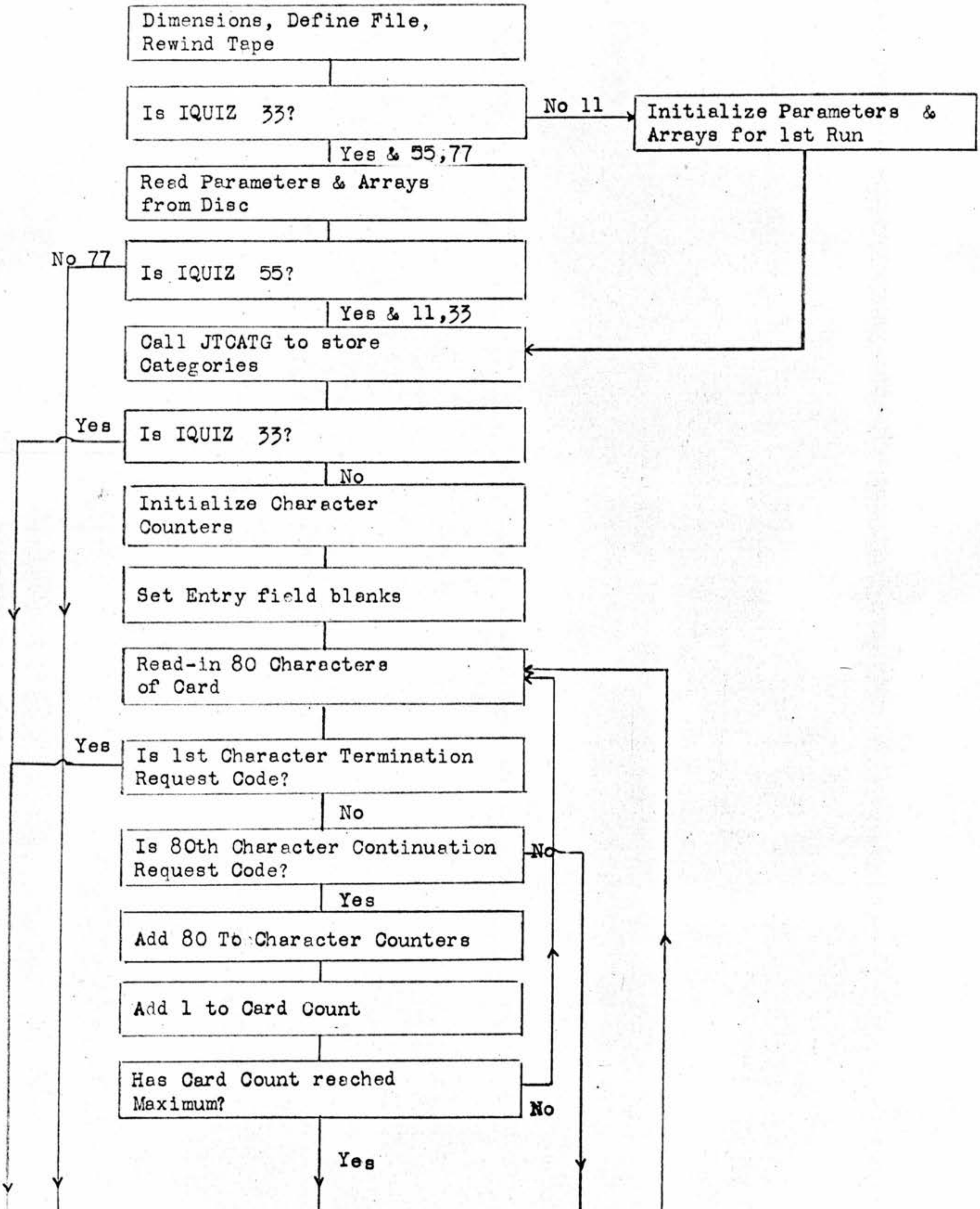

Is Numeric Data to be Included?
No
Yes
Call JTDATE to Handle Numeric Data Section
Display End of Cycle Message
Test for Display Request Code
No
VZTX
Display Number of Files Processed to-date
Read Display Code Requested
11
99
33
Read Number of Files to be displayed
Call JTSHOW to Display Requested Files
Display Instructions for Termination of Run
Read Run Termination Code
Display end of Run Message
EXIT

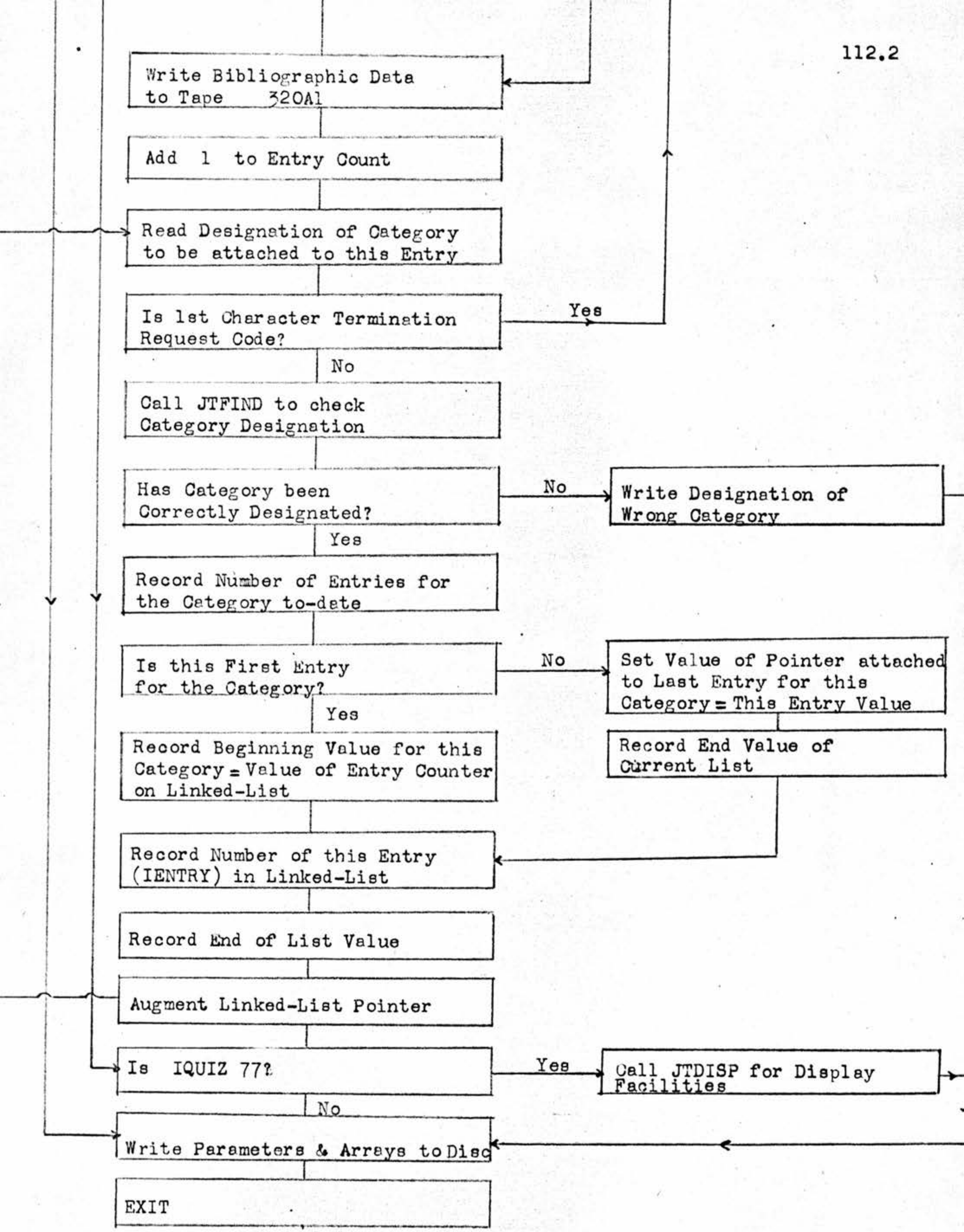
ii) Block Diagram of the Numeric Data Sub-routine JTDATE



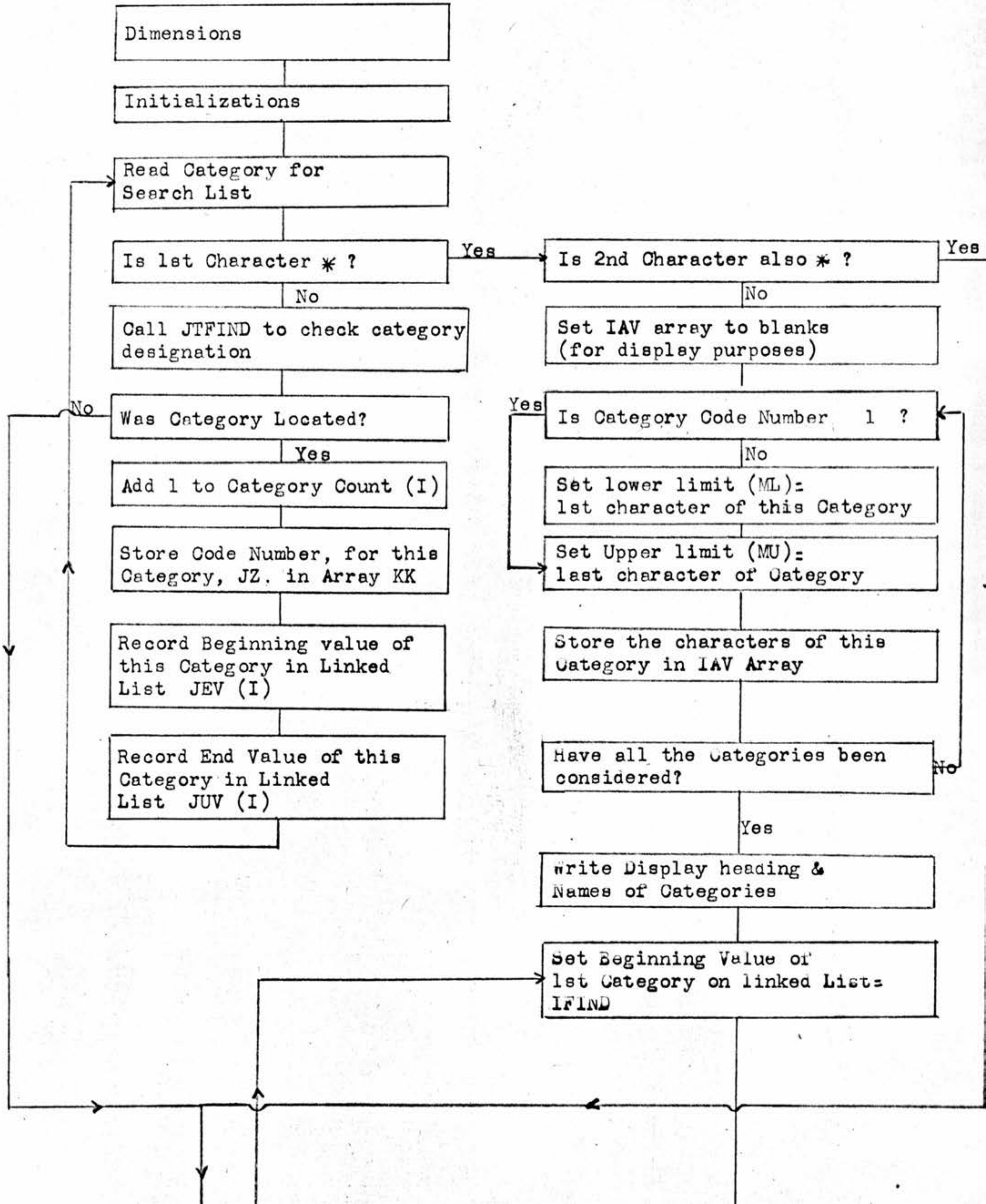
Display 1st Numeric Data Question
Read 1st Numeric Data Answer
Call JTAGED to give age on 31/12/69
Test for Impossible Reply for DDMMYY
Display DDMMYY as 'Age'
Was Reply Impossible?
Yes
Display Message stating Impossible Reply Item
Set Impossible Reply Flag
Is there an Answer Correction Request?
Display Subsequent Question
Read Answer
Is this 2nd Numeric Data Question?
Yes
Display Answer as Year in 20th century
Display Answer as Year
Is there an Answer Correction Request?
Yes
No
Have all topics been considered?
No
Yes
RETURN

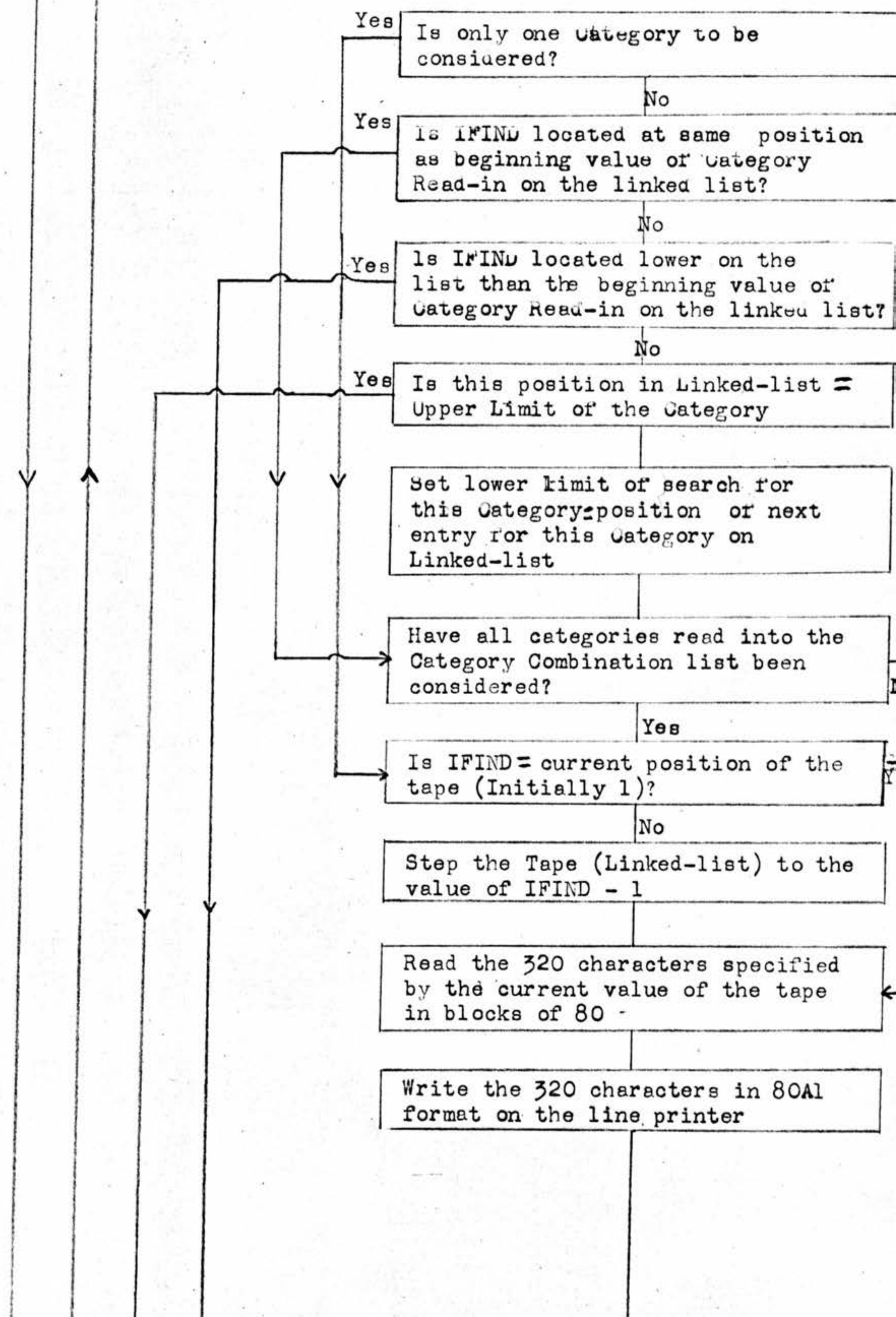
iii) Block Diagram of the Archiving Programme

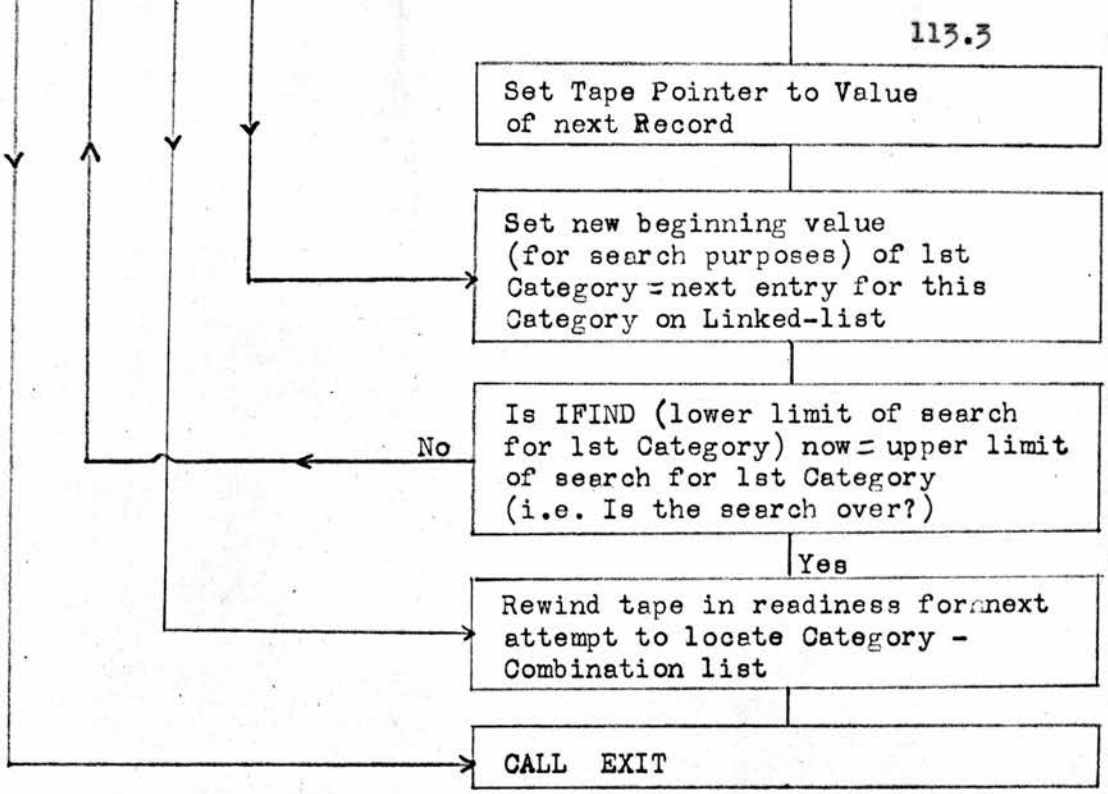

Dimensions, Define File, Rewind Tape
Is IQUIZ 33?
No 11
Initialize Parameters & Arrays for 1st Run
Yes & 55,77
Read Parameters & Arrays from Disc
No 77
Is IQUIZ 55?
Yes & 11,33
Call JTCATG to store Categories
Yes
Is IQUIZ 33?
No
Initialize Character Counters
Set Entry field blanks
Read-in 80 Characters of Card
Yes
Is 1st Character Termination Request Code?
No
Is 80th Character Continuation Request Code?
No
Yes
Add 80 To Character Counters
Add 1 to Card Count
Has Card Count reached Maximum?
No
Yes


Write Bibliographic Data to Tape 320A1
Add 1 to Entry Count
Read Designation of Category to be attached to this Entry
Is 1st Character Termination Request Code?
Yes
No
Call JTFIND to check Category Designation
Has Category been Correctly Designated?
No
Write Designation of Wrong Category
Yes
Record Number of Entries for the Category to-date
Is this First Entry for the Category?
No
Set Value of Pointer attached to Last Entry for this Category = This Entry Value
Record End Value of Current List
Yes
Record Beginning Value for this Category = Value of Entry Counter on Linked-List
Record Number of this Entry (IENTRY) in Linked-List
Record End of List Value
Augment Linked-List Pointer
Is IQUIZ 77?
Yes
Call JTDISP for Display Facilities
No
Write Parameters & Arrays to Disc
EXIT

iv) Block Diagram of the Archiving Programme's
Display Sub-routine JTDISP


Dimensions
Initializations
Read Category for Search List
Is 1st Character * ?
Yes
No
Call JTFIND to check category designation
Was Category Located?
No
Yes
Add 1 to Category Count (I)
Store Code Number, for this Category, JZ. in Array KK
Record Beginning value of this Category in Linked List JEV (I)
Record End Value of this Category in Linked List JUV (I)
Is 2nd Character also * ?
Yes
No
Set IAV array to blanks (for display purposes)
Is Category Code Number 1 ?
Yes
No
Set lower limit (ML)= 1st character of this Category
Set Upper limit (MU)= last character of Category
Store the characters of this Category in IAV Array
Have all the Categories been considered?
No
Yes
Write Display heading & Names of Categories
Set Beginning Value of 1st Category on linked List= IFIND


Is only one Category to be considered?
Yes
No
Is IFIND located at same position as beginning value of Category Read-in on the linked list?
Yes
No
Is IFIND located lower on the list than the beginning value of Category Read-in on the linked list?
Yes
No
Is this position in Linked-list = Upper Limit of the Category
Yes
Set lower limit of search for this Category=position of next entry for this Category on Linked-list
Have all categories read into the Category Combination list been considered?
Yes
Is IFIND = current position of the tape (Initially 1)?
No
Step the Tape (Linked-list) to the value of IFIND - 1
Read the 320 characters specified by the current value of the tape in blocks of 80
Write the 320 characters in 80A1 format on the line printer



Note IFIND = lower limit of search for common categories, based upon the category read in first into the category combination list. Economy in searching is achieved if the first category read-into this list has the smallest number of entries.

v) Illustration of the Working of the Indexed Sequential List used in the Archiving Programme

ILLUSTRATION of the WORKING of the INDEXED SEQUENTIAL LIST USED in the ARCHIVING PROGRAMME

Number in Array	Array Values	
1	ITOP(IE)	1
2	ITOP(IPX)	9
3	ITOP(IE)	1
4	ITOP(IPX)	7
5	ITOP(IE)	1
6	ITOP(IPX)	13
7	ITOP(IE)	2
8	ITOP(IPX)	17
9	ITOP(IE)	2
10	ITOP(IPX)	11
11	ITOP(IE)	3
12	ITOP(IPX)	19
13	ITOP(IE)	3
14	ITOP(IPX)	15
15	ITOP(IE)	4
16	ITOP(IPX)	23
17	ITOP(IE)	4
18	ITOP(IPX)	0
19	ITOP(IE)	4
20	ITOP(IPX)	27

Number in Array	Array Values	
21	ITOP(IE)	5
22	ITOP(IPX)	0
23	ITOP(IE)	6
24	ITOP(IPX)	0
25	ITOP(IE)	6
26	ITOP(IPX)	0
27	ITOP(IE)	7
28	ITOP(IPX)	0

IAH(L,1) Number of Entries	L Category Number	IAH(L,2) Beginning Value	IAH(L,3) End Value
5	1	1	27
3	2	3	17
4	3	5	23
1	4	21	-
1	5	25	-

<u>Sequence of ITOP Array Values attached to each Category</u>

Category Number	<u>Entry Values</u> Array Values						
1	1	2	3	4			7
	1	9	11	19			27
2	1	2		4			
	3	7		17			
3	1		3	4		6	
	5		13	15		23	
4					5		
					21		
5						6	
						25	

APPENDIX 2

UNIVERSITY OF ST. ANDREWS

Computing Laboratory,
The Mathematics Institute,
North Haugh, St. Andrews,
Fife, Scotland.

30th January, 1969.

Dear Sir,

I am engaged in a project designed to process all student records in this University by computer, and should be most grateful if you would allow us to have the benefit of your experience by completing the attached questionnaire. A sample of the Registration and Coding forms used at your University as well as any other relevant information would be appreciated. I shall be pleased to send you reports on any improvements to our own system which we may discover. May I take this opportunity to thank you, in advance, for your co-operation?

Yours faithfully,

James Hartley-Taylor

QUESTIONNAIRE

Please indicate correct answer by placing a cross [x] in the appropriate box.
If you desire more space for written answers, please do not hesitate to use reverse of sheet.

1. Are the Registration Forms preprinted with any information concerning the record of the student?

YES	
NO	

If YES, a) what is printed?

b) what is the source of the information?

---

2. Is any information added AFTER the student has completed the form?

YES	
NO	

If YES, what information is this?

---

3. Is any intermediate coding required between
   i) filling in of form by student and
   ii) data preparation for computer processing?

YES	
NO	

If YES , is this coding done by:

a) CLERICAL
b) COMPUTING
c) OTHER (please name) staff?

---

4. At what stage is information which is to be fed to the computer checked for errors:

a) DIRECTLY WITH STUDENT

b) CODING

BEFORE	
AFTER	

c) PUNCHING

BEFORE	
AFTER	

d) ANY OTHER TIME (please name)

5. Would you please give round-figure estimates of the proportion of errors which stem from the following factors (if known)?

	% STUDENT	% CODING STAFF	% TOTAL ERRORS
a) Interpretation of Registration Form			
b) Omissions in coding			
c) Incorrect coding			
d) Omissions in punching input			
e) Incorrect punching of input			
f) Others (please name)			
	100	100	100

Any comments on error detection and prevention would be most welcome.

6. What manual returns are prepared to provide checks on the validity of of the computer analysis?

7. Are you planning any changes in the near future?

YES	
NO	

If YES, would you please outline these changes?

APPENDIX 3

i) Listing of the Generalized Data Processing Programme
JT74XP and related Sub-routines

```
  TOTAL MEMORY REQUIREMENTS 000BA4 BYTES
 STORAGE MAP
IBCOM=    AT LOCATION 001000
FIOCS=    AT LOCATION 003928
ADCON=    AT LOCATION 0010BC
MAIN44&   AT LOCATION 004000
MAIN44    AT LOCATION 004000
JTDATA&   AT LOCATION 009A80
JTDATA    AT LOCATION 009A80
JTSTOR&   AT LOCATION 009F78
JTSTOR    AT LOCATION 009F78
JTSKOR&   AT LOCATION 00A2F0
JTSKOR    AT LOCATION 00A2F0
JTEXIT&   AT LOCATION 00A6E0
JTEXIT    AT LOCATION 00A6E0
JTREPT&   AT LOCATION 00A9D0
JTREPT    AT LOCATION 00A9D0
JTIGET&   AT LOCATION 00AC70
JTIGET    AT LOCATION 00AC70
JTDATE&   AT LOCATION 00B330
JTDATE    AT LOCATION 00B330
JTAGED&   AT LOCATION 00C198
JTAGED    AT LOCATION 00C198
JTSHOW&   AT LOCATION 00C7E0
JTSHOW    AT LOCATION 00C7E0
```

```
      C
      C      DATA PROCESSING PROGRAM APPLIED TO MATRICULATION SCHEDULES
      C
      C      DIMENSIONIZATION
      C
+XP          DIMENSION IAB(24),IAC(16),NOTB(40),MEMB(200)
+XP          DIMENSION IAGE(25),JAGE(25),KAGE(25),KYR(50),LXBY(100)
+XP          DIMENSION IAD(80),IAB1(16),IAB2(8)
+XP          DIMENSION IKEY(200),JKEY(500),INZ(20),IAA(6),KDZA(20),KDZB(20)
+XP          DIMENSION NOTA(200),MEMO(500),INA(20),IFAC(20),KDZC(20),KDZD(20)
+XP          DIMENSION  LKEY(25),KKEY(100),MSTOR(120),ISTOR(120)
+XP          DIMENSION LXEY(100),KXEY(100),IND(20),IAX(16),IXXA(40),JXXA(120)
+XP          EQUIVALENCE (IAB1,IAB(1)),(IAB2,IAB(17))
      C
      C      MONTHS OF YEAR
      C
+XP          DATA IAB1/'JANU','ARY ','FEBR','UARY','MARC','H   ','APRI','L   ',
            8'MAY ','    ','JUNE','    ','JULY','    ','AUGU','ST  '/
+XP          DATA IAB2/'SEPT','EMBR','OCTO','BER ','NOVE','MBER','DECE','MBER'/
      C
      C      LIST OF MESSAGES IN RESPONSE TO ANSWERS TO CATEGORIZED DATA
      C
+XP          DATA IAA/'OK C','HECK',' FOR','NO G','O FO','R   '/
      C
      C      MESSAGES IN RESPONSE TO IMPOSSIBLE PERSONAL NUMERIC DATA
      C
+XP          DATA IAX/'YOU ','HAVE',' GIV','EN A','N IM','POSS','IBLE',' REP',
            8'LY F','OR  ','XXXX','XXXX','XXXX','XXXX','XXXX','XXXX'/
      C
      C      INITIALIZATION
      C
+XP          IIA=1
+XP          IIB=1
+XP          I=1
+XP          MY=1
+XP          KK=1
+XP          KI=1
+XP          INY=1
+XP          JNY=1
+XP          KNY=1
+XP          IMY=1
+XP          JMY=1
+XP          KMY=1
+XP          MAP=0
+XP          JAP=0
+XP          J=1
+XP          JX=1
+XP          JPG=1
+XP          JB=1
```

```
+XP          N=0
+XP          NX=0
+XP          NPG=0
+XP          NUMB=0
+XP          NB=0
+XP          KU=1
+XP          JF=1
+XP          JDZA=0
+XP          JDZB=0
+XP          JDZC=0
+XP          MGA=1
+XP          MGB=1
+XP          MGC=1
+XP          IFL=1
+XP          KL=25
+XP          MN=28
+XP          KONTO=0
+XP          NJ=1
+XP          IVY=1
+XP          JVY=1
+XP          KVY=1
+XP          KKK=1
+XP          NRM=2
+XP          IADDA=1
      C
      C      DESCRIPTION OF PROGRAM
      C
+XP          WRITE(6,1)
+XP        1 FORMAT('1DATA PROCESSING PROGRAM APPLIED TO QUESTIONNAIRE DATA ',/
            8,'THE TOPICS  CAN BE SPECIFIED BY THE USER     ',/,'THREE TYPES OF
            8DATA CAN BE PROCESSED ',/,' 1.PERSONAL (LITERAL) DATA-CHECKED WITH
            8 RESPONDENT',/,' 2.CATEGORIZED DATA-CHECKED WITH STORAGE ANSWERS',
            8/,' 3.PERSONAL  NUMERIC DATA-CHECKED WITH RESPONDENT ',/,'A 3 FACT
            8OR FREQUENCY COUNT IS INCLUDED IN 2.      ',/,'FOR SPECIAL FACILIT
            8IES ENTER VZTX AS 1ST QUESTIONNAIRE ANSWER')
+XP          WRITE(6,2)
+XP        2 FORMAT('1TO USE READY-SUPPLIED',//,' 1.TOPICS',/,
            8' 2.STOAGE ANSWERS',/,' 3.INCLUSION SETTINGS',//,
            8'ENTER 10 OTHERWISE 00')
+XP          READ(9,5)JDAT
+XP        5 FORMAT(I2)
+XP          IF(JDAT-10)851,850,851
+XP      850 ISRI=25
+XP          IRI=25
+XP          JRI=28
+XP          IXN=6
+XP          JXN=IXN*4
+XP          ITPC=12
+XP          MRI=IRI+(ITPC*4)
```

```
+XP         NTP=18
+XP         DO 852 II=1,NTP
+XP         READ(5,4)IKA,IKB,IKC,IKD
+XP       4 FORMAT( A4,A4,A4,A4)
+XP         GO TO 8004
+XP     851 WRITE(6,1851)
+XP    1851 FORMAT('1A TOTAL OF 23 TOPICS CAN BE CONSIDERED',//,
           8' 19  CAN BE TREATED AS0',//,' 1.INDIVIDUAL DATA - CHECKED WITH RE
           8SPONDENT  ',//,' 2.CATEGORIZED DATA - CHECKED WITH STORAGE ANSWERS
           8   ',//' A FURTHER 4 TOPICS CAN BE CONSIDERED AS    ',/,
           8' 3.INDIVIDUAL NUMERIC DATA - CHECKED WITH RESPONDENT',////,
           8'NOW INDICATE NUMBER (I2) OF TOPICS OF INDIVIDUAL DATA')
+XP         READ(9,1852)IXN
+XP    1852 FORMAT(I2)
+XP         IF(IXN)6952,6953,6952
+XP    6953 ISKP=17
+XP         GO TO 6963
+XP    6952 IRI=(IXN*4)+1
+XP         ISRI=IRI
+XP         JRI=IRI+3
+XP    6963 WRITE(6,1853)
+XP    1853 FORMAT('INDICATE NUMBER (I2) OF CATEGORIZED DATA TOPICS')
+XP         READ(9,1854)ITPC
+XP    1854 FORMAT(I2)
+XP         IF(ITPC)6855,6852,6855
+XP    6852 KSKP=27
+XP         GO TO 6856
+XP    6855 IF(IXN)6417,6415,6417
+XP    6415 ISRI=1
+XP         IRI=1
+XP         JRI=4
+XP    6417 MRI=IRI+(ITPC*4)
+XP    6856 KTOT=IXN+ITPC
+XP         IF(KTOT-19)1855,1855,1866
+XP    1866 KOT=KTOT-19
+XP         WRITE(6,1867)IXN,ITPC,KOT
+XP    1867 FORMAT('YOUR REQUEST FOR ',I2,' INDIVIDUAL DATA TOPICS & ',I2,' CA
           8TEGORIZED DATA TOPICS',//,'EXCEEDS THE MAXIMUM NUMBER OF TOPICS AV
           8AILABLE (18) BY ',I4,//,'PLEASE GIVE YOUR REVISED REQUIREMENTS NOW
           8')
+XP         GO TO 851
+XP    1855 DO 852  KVC=1,KTOT
+XP         IF(KVC-IXN)1856,1856,2856
+XP    1856 WRITE(6,1857)KVC
+XP    1857 FORMAT('INDICATE INDIVIDUAL DATA TOPIC (4A4) NUMBER ',I2)
+XP         GO TO 3476
+XP    2856 IF(KSKP-27)5856,3399,5856
+XP    5856 WRITE(6,2857)IADDA
+XP    2857 FORMAT('INDICATE CATEGORIZED DATA TOPIC (4A4) NUMBER ',I2)
```

```
+XP          IADDA=IADDA+1
+XP     3476 READ(9,3478)IKA,IKB,IKC,IKD
+XP     3478 FORMAT( A4,A4,A4,A4)
+XP     8004 IAD(IIA)=IKA
+XP          IIA=IIA+1
+XP          IAD(IIA)=IKB
+XP          IIA=IIA+1
+XP          IAD(IIA)=IKC
+XP          IIA=IIA+1
+XP          IAD(IIA)=IKD
+XP          IIA=IIA+1
+XP      852 CONTINUE
+XP          IF(JDAT-10)3399,3410,3399
+XP     3510 IF(ITPC)3410,3410,22
+XP     3410 ICHK=44
+XP          NRM=2
+XP          DO 853 IHA=1,3
+XP          READ(5,3411)JKA,JKB,JKC,JKD
+XP     3411 FORMAT( A4,A4,A4,A4)
+XP          GO TO 3402
+XP     3399 WRITE(6,3400)
+XP     3400 FORMAT('INDICATE NUMBER (I2) OF TOPICS FOR INDIVIDUAL NUMERIC DATA
         8',//,'ITEM 1 RECORDS DDMMYY & DISPLAYS DD MONTH 19YY',/,
         8'ITEM 2 RECORDS YY & DISPLAYS 19YY',/,
         8'ITEMS 3+4 RECORDS YY & DISPLAYS YY')
+XP          READ(9,3401)ICAL
+XP     3401 FORMAT(I2)
+XP          MTOT=ICAL
+XP          IF(ICAL)3408,3407,3408
+XP     3407 ICHK=22
+XP          GO TO 854
+XP     3408 NRM=ICAL-1
+XP          ICHK=44
+XP          JCAL=ICAL*4
+XP          DO 853 IHU=1,MTOT
+XP          WRITE(6,3403)IHU
+XP     3403 FORMAT('INDICATE INDIVIDUAL NUMERIC DATA TOPIC (4A4) NUMBER ',I2)
+XP          READ(9,3404)JKA,JKB,JKC,JKD
+XP     3404 FORMAT( A4,A4,A4,A4)
+XP     3402 IAC(IIB)=JKA
+XP          IIB=IIB+1
+XP          IAC(IIB)=JKB
+XP          IIB=IIB+1
+XP          IAC(IIB)=JKC
+XP          IIB=IIB+1
+XP          IAC(IIB)=JKD
+XP          IIB=IIB+1
+XP      853 CONTINUE
+XP      854 IGAT=10
```

```
+XP         IDAT=10
+XP         IF(JDAT-10)1211,6119,1211
+XP    6119 CALL JTDATA(ITPC,IANS,NQZ,ICNK,ICHK,N,INZ,IXN,IKEY,JKEY,J,LKEY,
           8KKEY,IJ,IWORD,ICHAR)
+XP         GO TO 2999
+XP    1211 IF(ITPC)22,23,22
+XP      22 DO 1000 IJ=1,ITPC
+XP      19 WRITE(6,20)(IAD(IK),IK=IRI,JRI)
+XP      20 FORMAT('1INDICATE NUMBER OF STORAGE ANSWERS(I2) FOR ',4A4)
+XP         READ(9,21)IANS
+XP      21 FORMAT(I2)
+XP         GO TO 894
+XP      23 KSKP=27
+XP         GO TO 69
+XP     894 IF(IANS-1077952576)39,991,39
+XP     991 WRITE(6,930)
+XP     930 FORMAT('YOUR REPLY HAS BEEN READ AS BLANK.',//,'PLEASE CHECK REPLY
           8NEXT TIME')
+XP         GO TO 19
      C
      C     READ-IN OF STORAGE ANSWERS
      C
+XP      39 DO 2000 IA=1,IANS
+XP         WRITE(6,40)(IAD(IK),IK=IRI,JRI)
+XP      40 FORMAT('INDICATE A STORAGE ANSWER FOR ',4A4)
+XP         READ(9,41)(INZ(IW),IW=1,20)
+XP      41 FORMAT(20A4)
+XP         N=N+1
+XP         CALL  JTSTOR(N,INZ,IKEY,JKEY,J,LKEY,KKEY,IJ,IWORD,ICHAR)
+XP         IF(IWORD)900,900,980
+XP     900 IF(ICHAR)2000,2000,980
+XP     980 CALL JTEXIT(IWORD,ICHAR,IAD,KL,MN)
+XP         GO TO 9999
+XP    2000 CONTINUE
+XP         LKEY(IJ)=N
+XP         IRI=IRI+4
+XP         JRI=JRI+4
+XP    1000 CONTINUE
+XP         IRI=ISRI
+XP         JRI=IRI+3
+XP      69 WRITE(6,70)
+XP      70 FORMAT('1INDICATE NUMBER OF QUESTIONNAIRE PRESENTATIONS REQUIRED
           8 (I4)')
+XP         READ(9,71)NQZ
+XP      71 FORMAT(I4)
      C
      C
      C     QUESTIONNAIRE PRESENTATION BEGINS
      C
```

```
      C
      C      DISPLAY OF INSTRUCTIONS
      C
      C
+XP    2999 DO 3000 IJX=1,NQZ
+XP         JF=JF+1
+XP         MUK=0
+XP      89 WRITE(6,90)
+XP      90 FORMAT('1INFORMATION YOU REQUIRE TO OPERATE THE COMPUTER',/'TO MOV
           8E  TO THE NEXT STAGE OF THE PROGRAM AND TO START PRESS THE )SHIFT,
           8 AND    )ENTER, KEYS SIMULTANEOUSLY ',/,'TO ENTER YOUR REPLYO AFTE
           8R PRESSING )SHIFT, AND )ENTER, KEYS SIMULTANEOUSLY YOU WILL SEE EN
           8TER DATA ON THE SCREEN',/,'PRESS THE )SHIFT, AND )ERASE DISPLAY,
           8KEYS SIMULTANEOUSLY TO CLEAR THE SCREEN  NOW ENTER YOUR REPLY',/,'
           8FINALLY PRESS )SHIFT, AND )ENTER, TO MOVE ON TO THE NEXT STAGE, AS
           8 ALWAYS',/,'IF THE COMPUTER IS BUSY YOU MAY HAVE TO SHIFT AND ENTE
           8R SEVERAL TIMES',/,'SIMPLY OVER-TYPE ERRONEOUS CHARACTERS')
+XP         KL=1
+XP         MN=4
+XP         NQ=1
+XP         IF(ISKP-17)1499,4001,1499
+XP    1499 WRITE(6,499)
+XP     499 FORMAT('1PERSONAL DATA SECTION',//,'CHECK THAT YOUR REPLY HAS BEEN
           8 CORRECTLY RECORDED')
+XP         KXN=IXN
+XP         DO 4000 MJX=1,KXN
+XP      99 WRITE(6,100)(IAD(IK),IK=KL,MN)
+XP     100 FORMAT('1PLEASE GIVE YOUR ',4A4)
+XP         NX=NX+1
+XP    1909 READ(9,101)(INA(IV),IV=1,20)
+XP     101 FORMAT(20A4)
+XP         IF(INA(1)+437656601)1101,1102,1101
+XP    1102 NX=NX-1
+XP         GO TO 7000
+XP    1101 IF(INA(1)+404232217)102,103,102
+XP     103 CALL JTREPT(IAD,KL,MN)
+XP         KXN=KXN+1
+XP         NSKP=37
+XP         GO TO 1909
+XP     102 IF(INA(1)-1077952576)110,994,110
+XP     994 WRITE(6,1910)(IAD(IK),IK=KL,MN)
+XP    1910 FORMAT('YOUR REPLY HAS BEEN READ AS BLANKO',//,'PLEASE GIVE YOUR '
           84A4)
+XP         KXN=KXN+1
+XP         NSKP=37
+XP         GO TO 1909
+XP     110 CALL JTSKOR(NX,INA,NOTA,MEMO,JX,IWORD,ICHAR,NSKP)
+XP         IF(IWORD)909,909,981
+XP     909 IF(ICHAR)120,120,981
```

```
+XP     981 CALL JTEXIT(IWORD,ICHAR,IAD,KL,MN)
+XP         GO TO 9999
+XP     120 WRITE(6,130)(IAD(IK),IK=KL,MN),(INA(IV),IV=1,20)
+XP     130 FORMAT('1YOUR ',4A4,' HAS BEEN RECORDED ASO',//,20A4)
+XP         IF(MJX-KXN)1131,4441,1131
+XP    1131 WRITE(6,132)
+XP     132 FORMAT('0IF THIS IS WRONG ENTER XXXX IN REPLY TO NEXT QUESTION',/,
           8'OTHERWISE  CONTINUE NORMALLY')
+XP    4441 LXEY(IJX)=NX
+XP         KL=KL+4
+XP         MN=MN+4
+XP    4000 CONTINUE
+XP         WRITE(6,4003)
+XP    4003 FORMAT('OPERSONAL DATA SECTION HAS NOW BEEN COMPLETED',//,'TO PROC
           8EED ENTER NEXT',//,'OTHERWISE ENTER XXXX FOR CORRECTION')
+XP         READ(9,4004)(INA(1))
+XP    4004 FORMAT(A4)
+XP         IF(INA(1)+404232217)4001,783,4001
+XP     783 NSKP=37
+XP         CALL JTREPT(IAD,KL,MN)
+XP         GO TO 1909
+XP    4001 IF(KSKP-27)4002,189,4002
      C
      C     CATEGORIZED DATA SECTION-CHECKED WITH MASTER LIST
      C
+XP    4002 WRITE(6,4012)
+XP    4012 FORMAT('1CATEGORIZED DATA SECTION',//,'YOUR REPLIES WILL BE CHECKE
           8D WITH THE MASTER LIST OF ACCEPTABLE REPLIES')
+XP         KL=ISRI
+XP         MN=ISRI+3
+XP         NJ=1
+XP         DO 5067 IXZ=1,ITPC
+XP     140 IERROR=0
+XP     150 WRITE(6,160)(IAD(IK),IK=KL,MN)
+XP     160 FORMAT('1PLEASE GIVE YOUR ',4A4,' THE ACCEPTABLE REPLIES ARE AS FO
           8LOWSO')
+XP         IF(IERROR-1)1669,1779,1779
+XP    1779 NJ=NJ-1
+XP    1669 NRP=LKEY(NJ)
+XP         IF(NJ-1)1602,1603,1602
+XP    1603 MRP=1
+XP         GO TO 1674
+XP    1602 MRP=LKEY(NJ-1)+1
+XP    1674 DO 1650 IXI=MRP,NRP
+XP         IF(IXI-1)1621,1620,1621
+XP    1620 ILO=1
+XP         GO TO 1622
+XP    1621 ILO=IKEY(IXI-1)+1
+XP    1622 IHI=IKEY(IXI)
```

```
XP       87 WRITE(6,88)(JKEY(MH),MH=ILO,IHI)
XP       88 FORMAT(20A4)
XP     1650 CONTINUE
XP          NJ=NJ+1
XP          READ(9,161)(INA(JW),JW=1,20)
XP      161 FORMAT(20A4)
XP          IF(INA(1)+437656601)1162,7000,1162
XP     1162 CALL JTIGET(N,NQ,JKEY,IKEY,LKEY,KKEY,INA,J,IGET,IANS,IXN,MAP,
           8MSTOR,ISTOR,JAP,JDZA,JDZB,JDZC,KDZA,KDZB,KDZC,MGA,MGB,MGC,JF)
      C
      C     JTIGET SEARCHES FOR STORAGE ANSWER, SIGNALS OUTCOME OF SEARCH
      C            AND RECORDS THE ANSWER GIVEN IF ACCEPTABLE
      C
XP      169 IF(IGET)1990,1990,170
XP      170 WRITE(6,180)(IAA(IG),IG=1,3),(IAD(IK),IK=KL,MN),(INA(JW),JW=1,20)
XP      180 FORMAT(3A4,'  ',4A4,'  ',20A4)
XP          GO TO 210
XP     1990 WRITE(6,1991)(IAA(IG),IG=4,6),(IAD(IK),IK=KL,MN),(INA(JW),JW=1,20)
XP     1991 FORMAT(3A4,'  ',4A4,'  ',20A4)
XP          IERROR=IERROR+1
XP          IF(IERROR-2)150,1992,3992
XP     1992 WRITE(6,1993)IERROR,(IAD(IK),IK=KL,MN)
XP     1993 FORMAT('0YOU HAVE GIVEN',I2,' UNACCEPTABLE REPLIES FOR ',4A4,//,'P
           8LEASE GIVE ACCEPTABLE REPLY THIS TIME')
XP          GO TO 150
XP     3992 WRITE(6,3993)(IAD(IK),IK=KL,MN),(INA(JW),JW=1,20)
XP     3993 FORMAT('1THE PROBLEM IS NOW PASSED TO THE ARBITER FOR CLASSIFICATIO
           8N',//,'ERROR FOR ',4A4,//,'INPUT  = ',20A4,//,'PLEASE CLASSIFY CORR
           8ECTLY')
XP          READ(9,4994)(INA(JW),JW=1,20)
XP     4994 FORMAT(20A4)
XP          GO TO 1162
XP     9000 CONTINUE
      C
      C     RECORD THAT THIS TOPIC HAS BEEN CONSIDEREDO MOVE ON TO NEXT
      C
XP      210 NQ=NQ+1
XP          KL=KL+4
XP          MN=MN+4
XP     5067 CONTINUE
XP          WRITE(6,5068)
XP     5068 FORMAT('0CATEGORIZED DATA SECTION HAS NOW BEEN COMPLETED')
XP      189 IF(ICHK-44)220,289,220
      C
      C     PERSONAL NUMERIC DATA SECTION-CHECKED WITH RESPONDENT
      C
XP      289 WRITE(6,5069)
XP     5069 FORMAT('1PERSONAL NUMERIC DATA SECTION',//,'CHECK THAT YOUR REPLIE
           8S HAVE BEEN CORRECTLY RECORDED')
```

```
4XP        CALL JTDATE(IAD,IDAY,MTHS,IYRS,IAGE,JAGE,KAGE,IMY,JMY,
          8KMY,KYR,KK,IYST,IAB,IYC,IVY,JVY,KVY,KKK,IAC,IAX,NRM)
     C
     C     JTDATE HANDLES ALL PERSONAL NUMERIC DATA
     C
4XP    220 WRITE(6,221)
4XP    221 FORMAT('1THANK YOU FOR YOUR COOPERATION',//,'YOU HAVE NOW COMPLETE
          8D THE QUESTIONNAIRE ',//,'ENTER NEXT TO SHOW INSTRUCTIONS      ')
4XP        KONTO=KONTO+1
4XP        READ(9,3543)IVX
4XP   3543 FORMAT(A4)
4XP        IF(IVX+437656601)3000,7000,3000
4XP   3000 CONTINUE
     C
     C     BRANCH FOR SPECIAL FACILITIES
     C
4XP   7000 WRITE(6,7001)KONTO
4XP   7001 FORMAT('1',I4,' FILES HAVE BEEN PROCESSED TO-DATE',//,'THE FOLLOWI
          8NG FACILITIES ARE AVAILABLE',/,' 1.CONTINUE QUESTIONNAIRE PRESENTA
          8TION ENTER 11 (I2)  ',/,' 2.DISPLAY ALL PROCESSED FILES         EN
          8TER 33 (I2)  ',/,' 3.TERMINATE RUN                         ENTER 99
          8(I2)')
4XP        READ(9,7850)MARK
4XP   7850 FORMAT(I2)
4XP        IF(MARK-33)3000,7007,9888
4XP   7007 WRITE(6,7008)KONTO
4XP   7008 FORMAT('THERE ARE ',I4,' FILES ARE AVAILABLEO ENTER THE NUMBER
          8 YOU REQUIRE (I4)')
4XP        READ(9,7009)NFL
4XP   7009 FORMAT(I4)
4XP        IFL=1
4XP        CALL JTSHOW(NFL,IXN,ITPC,IAD,NOTA,MEMO,MSTOR,IKEY,JKEY,IAGE,JAGE,
          8KAGE,KI,IAB,KYR,ISKP,KSKP,ICHK,ISTOR,MAST,IYC,IAC,IFL,IRI,JRI,NRM)
4XP        GO TO 7000
     C
     C     JTSHOW DISPLAYS ALL,OR A SPECIFIED NUMBER,OF FILES IN ENTIRETY
     C
4XP   9888 WRITE(6,9009)
4XP   9009 FORMAT('YOU HAVE REQUESTED TERMINATION OF THE RUNO',//,'IF YOU WIS
          8H TO PROCEED WITH THIS COURSE ENTER 9999 OTHERWISE 1111')
4XP        READ(9,9001)NAA
4XP   9001 FORMAT(I4)
4XP        IF(NAA-9999)7000,9999,9999
4XP   9999 WRITE(6,9998)
4XP   9998 FORMAT('RUN HAS NOW TERMINATED')
4XP        STOP
4XP        END
```

```
ATA         SUBROUTINE JTDATA(ITPC,IANS,NQZ,ICNK,ICHK,N,INZ,IXN,
           8IKEY,JKEY,J,LKEY,KKEY,IJ,IWORD,ICHAR)
ATA         DIMENSION INZ(20),IKEY(200),JKEY(500),KKEY(100),LKEY(25)
ATA         READ(5,1)ITPC
ATA       1 FORMAT(I2)
ATA         DO 1000 IJ=1,ITPC
ATA         READ(5,2)IANS
ATA       2 FORMAT(I2)
ATA         DO 2000 IA=1,IANS
ATA         READ(5,3)(INZ(IW),IW=1,20)
ATA       3 FORMAT(20A4)
ATA         N=N+1
ATA         CALL JTSTOR(N,INZ,IKEY,JKEY,J,LKEY,KKEY,IJ,IWORD,ICHAR)
ATA    2000 CONTINUE
ATA         LKEY(IJ)=N
ATA    1000 CONTINUE
ATA         READ(5,4)NQZ
ATA       4 FORMAT(I4)
ATA         READ(5,5)IXN
ATA       5 FORMAT(I2)
ATA         READ(5,6)ICNK
ATA       6 FORMAT(I2)
ATA         ICHK=ICNK
ATA         RETURN
ATA         END
```

```
OR        SUBROUTINE JTSTOR(N,INZ,IKEY,JKEY,J,LKEY,KKEY,IJ,IWORD,ICHAR)
OR        DIMENSION IKEY(200),JKEY(500),INZ(20),KKEY(100),LKEY(25)
OR        IWORD=0
OR        ICHAR=0
OR        IF(N-200)22,22,999
OR     22 KG=20
OR    115 IF(INZ(KG)-1077952576)110,105,110
OR    105 KG=KG-1
OR        GO TO 115
OR    110 IKEY(N)=J+KG-1
OR    799 DO 320 LB=1,KG
OR        JKEY(J)=INZ(LB)
OR        J=J+1
OR        IF(J-1000)320,320,995
OR    320 CONTINUE
OR        GO TO 70
OR    999 IWORD=1
OR        GO TO 70
OR    995 ICHAR=1
OR     70 CONTINUE
OR        RETURN
OR        END
```

```
KOR          SUBROUTINE JTSKOR(NX,INA,NOTA,MEMO,JX,IWORD,ICHAR,NSKP)
KOR          DIMENSION INA(20),NOTA(200),MEMO(500)
KOR          IWORD=0
KOR          ICHAR=0
KOR          KG=20
KOR          IF(NSKP-37)115,22,115
KOR       22 JX=JX-KKG
KOR          NX=NNX
KOR          WRITE(6,890)NX
KOR      890 FORMAT('NX=',I4)
KOR          NSKP=97
KOR      115 IF(INA(KG)-1077952576)110,105,110
KOR      105 KG=KG-1
KOR          GO TO 115
KOR      110 NOTA(NX)=JX+KG-1
KOR          IF(NX-200)319,319,999
KOR      319 DO 320 LG=1,KG
KOR          MEMO(JX)=INA(LG)
KOR          JX=JX+1
KOR          IF(JX-500) 320,320,995
KOR      320 CONTINUE
KOR          KKG=KG
KOR          NNX=NX
KOR          GO TO 70
KOR      999 IWORD=1
KOR          GO TO 70
KOR      995 ICHAR=1
KOR       70 RETURN
KOR          END
```

```
XIT          SUBROUTINE JTEXIT(IWORD,ICHAR,IAD,KL,MN)
XIT          DIMENSION IAD(80)
XIT          IF(IWORD)900,900,992
XIT      992 WRITE(6,950)(IAD(IK),IK=KL,MN)
XIT      950 FORMAT('WORD OVERFLOW FOR ',4A4)
XIT          GO TO 999
XIT      900 IF(ICHAR)999,999,993
XIT      993 WRITE(9,960)(IAD(IK),KL,MN)
XIT      960 FORMAT('CHARACTERS OVERFLOW FOR ',4A4)
XIT      999 RETURN
XIT          END
```

```
PT         SUBROUTINE JTREPT(IAD,KL,MN)
PT         DIMENSION IAD(80)
PT         KL=KL-4
PT         MN=MN-4
PT         WRITE(6,104)(IAD(IK),IK=KL,MN)
PT     104 FORMAT('YOU INDICATED THAT YOUR ',4A4,' HAD BEEN WRONGLY
          8 RECORDED',//,'PLEASEE ENTER THE CORRECT INFORMATION NOW')
PT         RETURN
PT         END
```

```
GET         SUBROUTINE JTIGET(N,NQ,JKEY,IKEY,LKEY,KKEY,INA,J,IGET,IANS,IXN,
           8MAP,MSTOR,ISTOR,JAP,JDZA,JDZB,JDZC,KDZA,KDZB,KDZC,MGA,MGB,MGC,JF)
GET         DIMENSION JKEY(500),IKEY(200),KKEY(100),LKEY(25),INA(20)
GET         DIMENSION MSTOR(120),ISTOR(120)
GET         DIMENSION KDZA(20),KDZB(20),KDZC(20)
GET         KONTA=0
GET         IWORD=0
GET         ICHAR=0
GET         MU=1
GET         IF(NQ-1)330,300,330
GET     300 ILOLMT=1
GET         GO TO 320
GET     330 ILOLMT=LKEY(NQ-1)+1
GET     320 IUPLMT=LKEY(NQ)
GET         DO 6000 LMTS=ILOLMT,IUPLMT
GET         IF(LMTS-1)110,100,110
GET     100 JSEEKL=1
GET         GO TO 120
GET     110 JSEEKL=IKEY(LMTS-1)+1
GET     120 JSEEKU=IKEY(LMTS)
GET         MS=1
GET     777 DO 206 JD=JSEEKL,JSEEKU
GET         IF(JKEY(JD)-INA(MS))201,205,201
GET     205 MS=MS+1
GET         IGET=1207
GET     206 CONTINUE
GET         MAP=MAP+1
GET         JAP=JAP+1
GET         IF(LMTS-3)508,508,509
GET     508 IF(LMTS-2)1,2,3
GET       1 JDZA=JDZA+1
GET         KDZA(MGA)=JF
GET         MGA=MGA+1
GET         GO TO 509
GET       2 JDZB=JDZB+1
GET         KDZB(MGB)=JF
GET         MGB=MGB+1
GET         GO TO 509
GET       3 JDZC=JDZC+1
GET         KDZC(MGC)=JF
GET         MGC=MGC+1
GET     509 CONTINUE
GET         MSTOR(MAP)=JD
GET         JVZ=MSTOR(MAP)
GET         ISTOR(JAP)=JVZ-(MS)+2
GET         GO TO 70
GET     999 IWORD=1
GET         GO TO 70
GET     995 ICHAR=1
```

```
SET          GO TO 70
SET      201 IGET=0
SET     6000 CONTINUE
SET       70 RETURN
SET          END
```

```
ATF        SUBROUTINE JTDATE(IAD,IDAY,MTHS,IYRS,IAGE,JAGE,KAGE,IMY,JMY,
          8KMY,KYR,KK,IYST,IAB,IYC,IVY,JVY,KVY,KKK,IAC,IAX,NRM)
ATE        DIMENSION IAD(80),IAGE(25),JAGE(25),KAGE(25),KYR(50),IAB(24)
ATE        DIMENSION IAC(16),IAX(16)
ATE        IA=1
ATE        IZ=4
ATE    210 WRITE(6,211)(IAC(IX),IX=IA,IZ)
ATE    211 FORMAT('1PLEASE GIVE YOUR ',4A4,' IN THE FORM DDMMYY(I2,I2,I2)',//
          8'E.G.  230250   FOR 23 FEBRUARY 1950')
ATE   1211 READ(9,212)IDAY,MTHS,IYRS
ATE    212 FORMAT(I2,I2,I2)
ATE        CALL JTAGED(IDAY,MTHS,IYRS,IAGE,JAGE,KAGE,IVY,JVY,KVY,NYRS,NTHS,
          8NDAY,IRED,IAX,IBLU,IAC)
ATE        MTOS=JAGE(JVY)
ATE        MXTS=MTOS*2
ATE        MYHS=MXTS-1
ATE        WRITE(6,213)(IAC(IX),IX=IA,IZ)
ATE    213 FORMAT('YOUR ',4A4,' HAS BEEN RECORDED ASO')
ATE        WRITE(6,3213)(IAGE(IVY)),(IAB(IP),IP=MYHS,MXTS),(KAGE(KVY))
ATE   3213 FORMAT('0',I2,' ',2A4,'  19',I2)
ATE        IVY=IVY+1
ATE        JVY=JVY+1
ATE        KVY=KVY+1
ATE        WRITE(6,214)NYRS,NTHS,NDAY
ATE    214 FORMAT('OTHIS REPRESENTS AN AGE  ON  31 DECEMBER 1969 OF',I3,
          8' YEARS ',I3,' MONTHS ',I3,' DAYS')
ATE        IF(IBLU-99)1215,389,1215
ATE   1215 WRITE(6,215)
ATE    215 FORMAT('OIF THIS IS WRONG PLEASE ENTER XXXX OTHERWISE ENTER NEXT')
ATE        READ(9,216)NEXT
ATE    216 FORMAT(A4)
ATE        IF(NEXT+708450333)389,218,389
ATE    389 IRED=47
ATE        WRITE(6,390)(IAC(IX),IX=IA,IZ)
ATE    390 FORMAT('1YOUR ',4A4,' HAS BEEN NOTED AS WRONGLY RECORDED',//,'NOW
          8ENTER THE  CORRECT REPLY TO QUESTION')
ATE        GO TO 210
ATE    218 CONTINUE
ATE        DO 1000 IY=1,NRM
ATE        IA=IA+4
ATE        IZ=IZ+4
ATE   2213 WRITE(6,2313)(IAC(IX),IX=IA,IZ)
ATE   2313 FORMAT('1PLEASE GIVE YOUR ',4A4,'IN NUMERIC FORM (I2)',//,'EG 67 F
          8OR 1967 ',//,'    04 FOR SNR HONS YR')
ATE   1213 READ(9,254)IYST
ATE    254 FORMAT(I2)
ATE        KYR(KK)=IYST
ATE        KK=KK+1
ATE        IF(IY-1)265,255,265
```

```
TE      255 IYC=IYST
TE          WRITE(6,240)(IAC(IX),IX=IA,IZ),IYST
TE      240 FORMAT('YOUR ',4A4,' HAS BEEN RECORDED AS 19',I2)
TE          IF(69-IYST)1240,1299,1299
TE     1240 WRITE(6,1241)(IAX(KV),KV=1,10),(IAC(IX),IX=IA,IZ)
TE     1241 FORMAT('0THIS MEANS ',10A4,' '4A4)
TE          GO TO 1001
TE      265 WRITE(6,247)(IAC(IX),IX=IA,IZ),IYST
TE      247 FORMAT('YOUR ',4A4,' HAS BEEN RECORDED AS ',I2)
TE          IF(10-IYST)2240,1299,1299
TE     2240 WRITE(6,2241)(IAX(KV),KV=1,10),(IAC(IX),IX=IA,IZ)
TE     2241 FORMAT('0THIS MEANS ',10A4,'  ',4A4)
TE          GO TO 1001
TE     1299 WRITE(6,241)
TE      241 FORMAT('0IF THIS IS WRONG PLEASE ENTER XXXX OTHERWISE ENTER NEXT')
TE          READ(9,242)INEXT
TE      242 FORMAT(A4)
TE          IF(INEXT+708450333)1001,1000,1001
TE     1001 KK=KK-1
TE          WRITE(6,4987)(IAC(IX),IX=IA,IZ)
TE     4987 FORMAT('YOUR ',4A4,' HAS BEEN NOTED AS WRONGLY RECORDRD',//,'NOW E
         8NTER THE CORRECT REPLY TOO')
TE          GO TO 2213
TE     1000 CONTINUE
TE      289 RETURN
TE          END
```

```
ED        SUBROUTINE JTAGED(IDAY,MTHS,IYRS,IAGE,JAGE,KAGE,IVY,JVY,KVY,
         8NYRS,NTHS,NDAY,IRED,IAX,IBLU,IAC)
ED        DIMENSION IAGE(25),JAGE(25),KAGE(25),IAX(16),IAC(16)
ED        IBLU=11
ED        IF(IRED-47)2,1,2
ED      1 IVY=IVY-1
ED        JVY=JVY-1
ED        KVY=KVY-1
ED      2 NDAY=31-IDAY
ED        IF(31-IDAY)398,3,3
ED    398 WRITE(6,399)(IAX(KX),KX=1,10),(IAC(IU),IU=1,4)
ED    399 FORMAT('0',10A4,' DD=DAYS ',4A4)
ED        IBLU=99
ED      3 NTHS=12-MTHS
ED        IF(12-MTHS)498,4,4
ED    498 WRITE(6,499)(IAX(KX),KX=1,10),(IAC(IU),IU=1,4)
ED    499 FORMAT(10A4,' MM=MONTHS ',4A4)
ED        IBLU=99
ED      4 NYRS=69-IYRS
ED        IF(69-IYRS)598,600,600
ED    598 WRITE(6,599)(IAX(KX),KX=1,10),(IAC(IU),IU=1,4)
ED    599 FORMAT(10A4,' YY=YEARS  ',4A4)
ED        IBLU=99
ED    600 IAGE(IVY)=IDAY
ED        JAGE(JVY)=MTHS
ED        KAGE(KVY)=IYRS
ED        IRED=11
ED        RETURN
ED        END
```

```
      SUBROUTINE JTSHOW(NFL,IXN,ITPC,IAD,NOTA,MEMO,MSTOR,IKEY,JKEY,IAGE,
     8JAGE,KAGE,KI,IAB,KYR,ISKP,KSKP,ICHK,ISTOR,MAST,IYC,IAC,IFL,IRI,
     8JRI,NRM)
      DIMENSION IAD(80),NOTA(200),MEMO(500),IKEY(200),JKEY(500)
      DIMENSION IAGE(25),JAGE(25),KAGE(25),IAB(24),KYR(50),MSTOR(120)
      DIMENSION ISTOR(120),IAC(16)
      IF(MAST-98)1,600,1
  600 IF(IFL-1)602,1,602
  602 IF(IXN-1)1600,2,603
  603 ICH=IXN-1
      NZ=(IFL*IXN)-ICH
      IMY= IFL
      JMY= IFL
      KMY= IFL
      KI=(IFL*2)-1
      GO TO 1600
    2 NZ=IFL
      IMY=IFL
      JMY=IFL
      KMY=IFL
      KI=(IFL*2)-1
 1600 IF(ITPC-1)799,3,604
    3 MSTUD=ITPC
      KSTUD=ITPC
      GO TO 799
  604 JCH=ITPC-1
      MSTUD=(IFL*ITPC)-JCH
      KSTUD=(IFL*ITPC)-JCH
      GO TO 799
    1 NZ=1
      IMY=1
      JMY=1
      KMY=1
      KI=1
      MSTUD=1
      KSTUD=1
  799 DO 800 ISTUD=IFL,NFL
  808 WRITE(6,80)ISTUD
   80 FORMAT('1DATA FROM FILE',I4,' IS AS FOLLOWS')
      IF(ISKP-17)81,701,81
   81 KL=1
      MN=4
      DO 700 JZ=1,IXN
      IF(NZ-1)21,20,21
   20 ILO=1
      GO TO 22
   21 ILO=NOTA(NZ-1)+1
   22 IHI=NOTA(NZ)
      WRITE(6,88)(IAD(IK),IK=KL,MN),(MEMO(JF),JF=ILO,IHI)
```

```
HOW      88 FORMAT(4A4,' ',20A4)
HOW         KL=KL+4
HOW         MN=MN+4
HOW         NZ=NZ+1
HOW     700 CONTINUE
HOW     701 IF(KSKP-27)702,751,702
HOW     702 KL=IRI
HOW         MN=JRI
HOW         DO 750 MT=1,ITPC
HOW         KHI      =MSTOR(MSTUD)
HOW         MSTUD=MSTUD+1
HOW      24 KLO= ISTOR(KSTUD)
HOW         KSTUD=KSTUD+1
HOW         WRITE(6,89)(IAD(IK),IK=KL,MN),(JKEY(JR),JR=KLO,KHI)
HOW      89 FORMAT(4A4,' ',20A4)
HOW         KL-KL+4
HOW         MN=MN+4
HOW     750 CONTINUE
HOW     751 IF(ICHK-44)800,752,752
HOW     752 IA=1
HOW         IZ=4
HOW         IXY=JAGE(JMY)
HOW         IXB=IXY*2
HOW         IXA=IXB-1
HOW         WRITE(6,754)(IAC(IX),IX=IA,IZ),IAGE(IMY),(IAB(IR),IR=IXA,IXB),
           8KAGE(KMY)
HOW     754 FORMAT(4A4,' ',I2,' ',2A4,' 19',I2)
HOW         IMY=IMY+1
HOW         JMY=JMY+1
HOW         KMY=KMY+1
HOW         DO 753 KZ=1,NRM
HOW         IA=IA+4
HOW         IZ=IZ+4
HOW         IF(KZ-1)738,740,738
HOW     738 WRITE(6,739)(IAC(IX),IX=IA,IZ),(KYR(KI))
HOW     739 FORMAT(4A4,' ',I2)
HOW         GO TO 772
HOW     740 WRITE(6,741)(IAC(IX),IX=IA,IZ),(KYR(KI))
HOW     741 FORMAT(4A4,' 19',I2)
HOW     772 KI=KI+1
HOW     753 CONTINUE
HOW     800 CONTINUE
HOW         RETURN
HOW         END
```

ii) Listing of the Matriculation Schedule Processing Programme JT74 with Additional Sub-routines

```
  TOTAL MEMORY REQUIREMENTS 000BA4 BYTES
 STORAGE MAP
IBCOM=    AT LOCATION 001000
FIOCS-    AT LOCATION 003928
ADCON=    AT LOCATION 0010BC
MAIN44&   AT LOCATION 004000
MAIN44    AT LOCATION 004000
JTRAND&   AT LOCATION 00A630
JTRAND    AT LOCATION 00A630
JTDATA&   AT LOCATION 00A9A0
JTDATA    AT LOCATION 00A9A0
JTRCRD&   AT LOCATION 00AE98
JTRCRD    AT LOCATION 00AE98
JTSTOR&   AT LOCATION 00B1D8
JTSTOR    AT LOCATION 00B1D8
JTEXIT&   AT LOCATION 00B550
JTEXIT    AT LOCATION 00B550
JTREPT&   AT LOCATION 00B840
JTREPT    AT LOCATION 00B840
JTSKOR&   AT LOCATION 00BAE0
JTSKOR    AT LOCATION 00BAE0
JTIGET&   AT LOCATION 00BED0
JTIGET    AT LOCATION 00BED0
JTDATE&   AT LOCATION 00C590
JTDATE    AT LOCATION 00C590
JTAGED&   AT LOCATION 00D3F8
JTAGED    AT LOCATION 00D3F8
JTFACT&   AT LOCATION 00DA40
JTFACT    AT LOCATION 00DA40
JTFILE&   AT LOCATION 00E098
JTFILE    AT LOCATION 00E098
JTSHOW&   AT LOCATION 00EE10
JTSHOW    AT LOCATION 00EE10
```

```
C
C     DATA PROCESSING PROGRAM APPLIED TO MATRICULATION SCHEDULES
C
C     DIMENSIONIZATION
C
      DIMENSION IAB1(16),IAB2(8),IAB(24),IAC(16),NOTB(40),MEMB(200)
      DIMENSION IAGE(25),JAGE(25),KAGE(25),KYR(50),LXBY(100)
      DIMENSION IAD1(16),IAD2(16),IAD3(16),IAD4(16),IAD5(16),IAD(80)
      DIMENSION IKEY(200),JKEY(500),INZ(20),IAA(6),KDZA(20),KDZB(20)
      DIMENSION NOTA(200),MEMO(500),INA(20),IFAC(20),KDZC(20),KDZD(20)
      DIMENSION  LKEY(25),KKEY(100),MSTOR(120),ISTOR(120)
      DIMENSION LXEY(100),KXEY(100),IND(20),IAX(16),IXXA(40),JXXA(120)
      EQUIVALENCE (IAD1,IAD(1)),(IAD2,IAD(17)),(IAD3,IAD(33))
      EQUIVALENCE (IAD4,IAD(49)),(IAD5,IAD(65))
      EQUIVALENCE (IAB1,IAB(1)),(IAB2,IAB(17))
C
C     TOPICS AVAILABLE FOR CONSIDERATION
C
C     PERSONAL DATA (LITERAL)O CHECKED WITH RESPONDENT
C
      DATA IAD1/'NAME',',INI','TS,T','ITLE','TERM',' ADD','RESS',
     8'+TEL','LAST',' EDU','C ES','TRMT','FATH','ER*S',' NAM','E   '/
      DATA IAD2/'FATH','ER*S',' PRO','FSSN','HOME',' ADD','RESS','+TEL',
C
C     CATEGORIZED  DATA 0 CHECKED WITH STORAGE ANSWERS (4A4)
C
     8'DOMI','CILE',' ZON','E   ','TERM',' ACC','OMOD','ATN '/
      DATA IAD3/'NATI','ONAL','ITY ','    ','FEE ','CLAS','SIFI','CATN',
     8'GRAN','T SO','URCE','    ','BASI','S OF',' ENT','RY  '/
      DATA IAD4/'FACU','LTY ','    ','    ','DEGR','EE I','N VI','EW  ',
     8'METH','OD O','F ST','UDY ','SUBJ','ECT ',' THI','S YR'/
      DATA IAD5/'SEXU','AL S','TATU','S   ','MARI','TAL ','STAT','US  ',
     8'XXXX','XXXX','XXXX','XXXX'/
C
C     MONTHS OF YEAR
C
      DATA IAB1/'JANU','ARY ','FEBR','UARY','MARC','H   ','APRI','L   ',
     8'MAY ','    ','JUNE','    ','JULY','    ','AUGU','ST  '/
      DATA IAB2/'SEPT','EMBR','OCTO','BER ','NOVE','MBER','DECE','MBER'/
C
C     TOPICS FOR PERSONAL NUMERIC DATAO CHECKED WITH RESPONDENT
C
      DATA  IAC/'DATE',' OF ','BIRT','H   ','YEAR',' OF ','ENTR','Y   ',
     8'YEAR',' OF ','COUR','SE  '/
C
C     LIST OF MESSAGES IN RESPONSE TO ANSWERS TO CATEGORIZED DATA
C
      DATA IAA/'OK C','HECK',' FOR','NO G','O FO','R   '/
C
```

```
C     MESSAGES IN RESPONSE TO IMPOSSIBLE PERSONAL NUMERIC DATA
C
      DATA IAX/'YOU ','HAVE',' GIV','EN A','N IM','POSS','IBLE',' REP',
     8'LY F','OR  ','IN Y','OUR ','DATE',' OF ','BIRT','H   '/
C
C     INITIALIZATION
C
      I=1
      MY=1
      KK=1
      KI=1
      INY=1
      JNY=1
      KNY=1
      IMY=1
      JMY=1
      KMY=1
      MAP=0
      JAP=0
      J=1
      JX=1
      JPG=1
      JB=1
      N=0
      NX=0
      NPG=0
      NUMB=0
      NB=0
      KU=1
      JF=1
      JDZA=0
      JDZB=0
      JDZC=0
      MGA=1
      MGB=1
      MGC=1
      IFL=1
      KL=25
      MN=28
      KONTO=0
      NJ=1
      IVY=1
      JVY=1
      KVY=1
      KKK=1
C
C     DESCRIPTION OF PROGRAM
C
      WRITE(6,1)
```

```
    1 FORMAT('1DATA PROCESSING PROGRAM APPLIED TO MATRICULATION FORMS',/
     8,'THE TOPICS  AVAILABLE ARE IN PRE-SET ARRAYS ',/,'THREE TYPES OF
     8DATA CAN BE PROCESSED ',/,' 1.PERSONAL (LITERAL) DATA-CHECKED WITH
     8 RESPONDENT',/,' 2.CATEGORIZED DATA-CHECKED WITH STORAGE ANSWERS',
     8/,' 3.PERSONAL  NUMERIC DATA-CHECKED WITH RESPONDENT ',/,'TYPES 1+
     82 ARE LINKED BUT ARE INDEPENDENT OF TYPE 3',/,'FOR SPECIAL FACILIT
     8IES ENTER VZTX AS 1ST QUESTIONNAIRE ANSWER')
C
C     BRANCH FOR READY SUPPLIED STORAGE ANSWERS AND INCLUSION SETTINGS
C
      WRITE(6,9)
    9 FORMAT('1 TO USE READY-SUPPLIED STORAGE ANSWERS AND INCLUSION SETT
     8NGS',//,'  ENTER 10 OTHERWISE 00')
      READ(9,1217)IDAT
 1217 FORMAT(I2)
      IF(IDAT-10)1010,6119,1010
 6119 CALL JTDATA(ITPC,IANS,NQZ,ICNK,ICHK,N,INZ,IXN,IKEY,JKEY,J,LKEY,
     8KKEY,IJ,IWORD,ICHAR)
      GO TO 581
C
C     SETTING OF LOOPS FOR CATEGORIZED DATA SECTION
C
 1010 WRITE(6,10)
   10 FORMAT('1INDICATE NUMBER OF TOPICS FROM LIST BELOW TO BE GIVEN ',/
     8'CATEGORIZED STORAGE ANSWERS (I2)',/,'ENTER 00 TO BLOCK TYPE')
      WRITE(6,13)(IAD(IK),IK=25,72)
   13 FORMAT('0 1.',4A4,'  2.',4A4,'  3.',4A4,'  4.',4A4,//,'  5.',4A4,
     8'  6.',4A4,'  7.',4A4,'  8.',4A4,//,'  9.',4A4,' 10.',4A4,' 11.',
     84A4,' 12.',4A4,//,'FACTOR 1 INCORPORATES A 3 FACTOR FREQUENCY COUN
     8T')
      READ(9,11)ITPC
   11 FORMAT(I2)
      IF(ITPC)22,23,22
   22 DO 1000 IJ=1,ITPC
   19 WRITE(6,20)(IAD(IK),IK=KL,MN)
   20 FORMAT('1INDICATE NUMBER OF STORAGE ANSWERS(I2) FOR ',4A4)
      READ(9,21)IANS
   21 FORMAT(I2)
      GO TO 894
   23 KSKP=27
      GO TO 69
  894 IF(IANS-1077952576)39,991,39
  991 WRITE(6,930)
  930 FORMAT('YOUR REPLY HAS BEEN READ AS BLANK.',//,'PLEASE CHECK REPLY
     8NEXT TIME')
      GO TO 19
C
C     READ-IN OF STORAGE ANSWERS
C
```

```
   39 DO 2000 IA=1,IANS
      WRITE(6,40)(IAD(IK),IK=KL,MN)
   40 FORMAT('INDICATE A STORAGE ANSWER FOR ',4A4)
      READ(9,41)(INZ(IW),IW=1,20)
   41 FORMAT(20A4)
      N=N+1
      CALL  JTSTOR(N,INZ,IKEY,JKEY,J,LKEY,KKEY,IJ,IWORD,ICHAR)
      IF(IWORD)900,900,980
  900 IF(ICHAR)2000,2000,980
  980 CALL JTEXIT(IWORD,ICHAR,IAD,KL,MN)
      GO TO 9999
 2000 CONTINUE
      LKEY(IJ)=N
      KL=KL+4
      MN=MN+4
 1000 CONTINUE
C
C     SETTING OF LOOP FOR QUESTIONNAIRE PRESENTATION
C
   69 WRITE(6,70)
   70 FORMAT('1INDICATE NUMBER OF QUESTIONNAIRE PRESENTATIONS REQUIRED
     8 (I4)')
      READ(9,71)NQZ
   71 FORMAT(I4)
C
C     SETTING OF LOOP FOR PERSONAL DATA (LITERAL)
C
      WRITE(6,80)(IAD(IK),IK=1,24)
   80 FORMAT('INDICATE NUMBER OF PERSONAL DATA TOPICS FROM LIST BELOW TO
     8 BE CONSIDERED (I2)',/,'ENTER 00 TO CONFINE PROG TO CATEGORIZED DA
     8TA'//,' 1.',4A4,'  2.',4A4,'  3.',4A4,'  4.',4A4,//,' 5.',4A4,
     8'  6.',4A4)
      READ(9,81)IXN
   81 FORMAT(I2)
      IF(IXN)581,182,581
  182 ISKP=17
      GO TO 2999
  581 KXN=IXN
      IF(IDAT-10)1181,1066,1181
 1066 IF(KXN-6)1181,2281,1181
 2281 WRITE(6,2282)
 2282 FORMAT('1IF EXISTING FILES ARE TO BE DISPLAYED TO RETURNING STUDEN
     8TS ENTER 10',//,'OTHERWISE ENTER 00')
      READ(9,2283)IFRST
 2283 FORMAT(I2)
      IF(IFRST-10)1181,2284,1181
 2284 ICAND=5
      DO 2700 IKX=1,ICAND
      DO 1700 IKL=1,6
```

```
+       NB=NB+1
+       READ(5,5758)(INA(KV),KV=1,20)
+  5758 FORMAT(20A4)
+       CALL JTSKOR(NB,INA,NOTB,MEMB,JB,IWORD,ICHAR,NSKP)
+  1700 CONTINUE
+       LXBY(IKX)=NB
+  2700 CONTINUE
+       GO TO 2999
 C
 C      SETTING OF PERSONAL NUMERIC DATA INCLUSION FLAG
 C
+  1181 WRITE(6,82)(IAC(IX),IX=1,12)
+    82 FORMAT('IF ',4A4,' ',4A4,' ',4A4,' ARE TO BE READ',//,'ENTER  44
       8(I2)  OTHERWISE ENTER 22 (I2)')
+       READ(9,83)ICNK
+    83 FORMAT(I2)
+       ICHK=ICNK
 C
 C      QUESTIONNAIRE PRESENTATION BEGINS
 C
 C
 C
 C
 C      DISPLAY OF INSTRUCTIONS
 C
 C
+  2999 DO 3000 IJX=1,NQZ
+       JF=JF+1
+       MUK=0
+    89 WRITE(6,90)
+    90 FORMAT('1INFORMATION YOU REQUIRE TO OPERATE THE COMPUTER',/'TO MOV
       8E  TO THE NEXT STAGE OF THE PROGRAM AND TO START PRESS THE )SHIFT,
       8 AND    )ENTER, KEYS SIMULTANEOUSLY ',/,'TO ENTER YOUR REPLYO AFTE
       8R PRESSING )SHIFT, AND )ENTER, KEYS SIMULTANEOUSLY YOU WILL SEE EN
       8TER DATA ON THE SCREEN',/,'PRESS THE )SHIFT, AND )ERASE DISPLAY,
       8KEYS SIMULTANEOUSLY TO CLEAR THE SCREEN  NOW ENTER YOUR REPLY',/,'
       8FINALLY PRESS )SHIFT, AND )ENTER, TO MOVE ON TO THE NEXT STAGE, AS
       8 ALWAYS',/,'IF THE COMPUTER IS BUSY YOU MAY HAVE TO SHIFT AND ENTE
       8R SEVERAL TIMES',/,'SIMPLY OVER-TYPE ERRONEOUS CHARACTERS')
+       KL=1
+       MN=4
+       NQ=1
+       IF(ISKP-17)3999,4001,3999
+  3999 IF(IFRST-10)1499,2250,1499
+  2250 WRITE(6,2251)
+  2251 FORMAT('1IS THIS YOUR FIRST REGISTRATION IN THIS UNIVERSITY  '//,'
       8ENTER YES OR NO')
+       READ(9,2252)IYES
+  2252 FORMAT(A4)
```

```
      IF(IYES+389684672)7788,1499,7788
 7788 IF(IYES+437656601)2253,7000,2253
 2253 WRITE(6,2254)(IAD(IK),IK=KL,MN)
 2254 FORMAT('PLEASE GIVE YOUR ',4A4,' IN THE FORMO',//,'SMITH,J,MR.',//,
     8'CHECK EVERY DETAIL OF YOUR DESIGNATION')
      MUK=MUK+1
      READ(9,2255)(INA(IV),IV=1,20)
 2255 FORMAT(20A4)
      MAX=30
      CALL JTRCRD(INA,NOTB,MEMB,JFL,IOK,MAX)
      IF(IOK-100)2256,2257,2256
 2257 WRITE(6,2258)
 2258 FORMAT('YOUR FILE HAS BEEN LOCATED AND IS READY FOR DISPLAY')
      MAST=98
      MFL=JFL
      ISKP=30
      KSKP=27
      ICHK=22
      CALL JTSHOW(MFL,IXN,ITPC,IAD,NOTB,MEMB,MSTOR,IKEY,JKEY,IAGE,
     8JAGE,KAGE,KI,IAB,KYR,ISKP,KSKP,ICHK,ISTOR,MAST,IYC,IAC,JFL)
      MAST=14
      KSKP=47
      ICHK=44
      WRITE(6,2259)
 2259 FORMAT('OIS THE ABOVE RECORD CORRECT ',//,'ENTER YES OR NO')
      READ(9,2449)MYES
 2449 FORMAT(A4)
      IF(MYES+389684672)2264,4402,2264
 4402 CALL JTRAND(NOTA,MEMO,NX,JX,NOTB,MEMB,NB,JB,JFL)
      GO TO 4002
 2256 NPG=NPG+1
      NUMB=NUMB+1
      NSKP=100
      CALL JTSKOR(NPG,INA,IXXA,JXXA,JPG,IWORD,ICHAR,NSKP)
      IF(IWORD)4900,4900,4980
 4900 IF(ICHAR)5557,5557,4980
 4980 CALL JTEXIT(IWORD,ICHAR,IAD,KL,MN)
      GO TO 9999
 5557 WRITE(6,2260)
 2260 FORMAT('YOUR FILE HAS NOT BEEN LOCATED')
      IF(MUK-3)2262,2262,2264
 2262 WRITE(6,2263)
 2263 FORMAT('OIF YOU WISH TO TRY AGAIN ENTER NEXT',//,'OTHERWISE ENTER
     8XXXX AND COMPLETE FULL QUESTIONNAIRE')
      READ(9,2261)NNNN
 2261 FORMAT(A4)
      IF(NNNN+404232217)2253,1499,2253
 2264 WRITE(6,2265)
 2265 FORMAT('PLEASE COMPLETE FULL QUESTIONNAIRE')
```

```
1499 WRITE(6,499)
 499 FORMAT('1PERSONAL DATA SECTION',//,'CHECK THAT YOUR REPLY HAS BEEN
    8 CORRECTLY RECORDED')
     KXN=IXN
     DO 4000 MJX=1,KXN
  99 WRITE(6,100)(IAD(IK),IK=KL,MN)
 100 FORMAT('1PLEASE GIVE YOUR ',4A4)
     NX=NX+1
1909 READ(9,101)(INA(IV),IV=1,20)
 101 FORMAT(20A4)
     IF(INA(1)+437656601)1101,1102,1101
1102 NX=NX-1
     GO TO 7000
1101 IF(INA(1)+404232217)102,103,102
 103 CALL JTREPT(IAD,KL,MN)
     KXN=KXN+1
     NSKP=37
     GO TO 1909
 102 IF(INA(1)-1077952576)110,994,110
 994 WRITE(6,1910)(IAD(IK),IK-KL,MN)
1910 FORMAT('YOUR REPLY HAS BEEN READ AS BLANKO',//,'PLEASE GIVE YOUR '
    84A4)
     KXN=KXN+1
     NSKP=37
     GO TO 1909
 110 CALL JTSKOR(NX,INA,NOTA,MEMO,JX,IWORD,ICHAR,NSKP)
     IF(IWORD)909,909,981
 909 IF(ICHAR)120,120,981
 981 CALL JTEXIT(IWORD,ICHAR,IAD,KL,MN)
     GO TO 9999
 120 WRITE(6,130)(IAD(IK),IK=KL,MN),(INA(IV),IV=1,20)
 130 FORMAT('1YOUR ',4A4,' HAS BEEN RECORDED ASO',//,20A4)
     IF(MJX-KXN)1131,4441,1131
1131 WRITE(6,132)
 132 FORMAT('OIF THIS IS WRONG ENTER XXXX IN REPLY TO NEXT QUESTION',/,
    8'OTHERWISE  CONTINUE NORMALLY')
4441 LXEY(IJX)=NX
     KL=KL+4
     MN=MN+4
4000 CONTINUE
     WRITE(6,4003)
4003 FORMAT('OPERSONAL DATA SECTION HAS NOW BEEN COMPLETED',//,'TO PROC
    8EED ENTER NEXT',//,'OTHERWISE ENTER XXXX FOR CORRECTION')
     READ(9,4004)(INA(1))
4004 FORMAT(A4)
     IF(INA(1)+404232217)4001,783,4001
 783 NSKP=37
     CALL JTREPT(IAD,KL,MN)
     GO TO 1909
```

```
 4001 IF(KSKP-27)4002,189,4002
C
C     CATEGORIZED DATA SECTION-CHECKED WITH MASTER LIST
C
 4002 WRITE(6,4012)
 4012 FORMAT('1CATEGORIZED DATA SECTION',//,'YOUR REPLIES WILL BE CHECKE
     8D WITH THE MASTER LIST OF ACCEPTABLE REPLIES')
      KL=25
      MN=28
      NJ=1
      DO 5067 IXZ=1,ITPC
  140 IERROR=0
  150 WRITE(6,160)(IAD(IK),IK=KL,MN)
  160 FORMAT('1PLEASE GIVE YOUR ',4A4,' THE ACCEPTABLE REPLIES ARE AS FO
     8LLOWSO')
      IF(IERROR-1)1669,1779,1779
 1779 NJ=NJ-1
 1669 NRP=LKEY(NJ)
      IF(NJ-1)1602,1603,1602
 1603 MRP=1
      GO TO 1674
 1602 MRP=LKEY(NJ-1)+1
 1674 DO 1650 IXI=MRP,NRP
      IF(IXI-1)1621,1620,1621
 1620 ILO=1
      GO TO 1622
 1621 ILO=IKEY(IXI-1)+1
 1622 IHI=IKEY(IXI)
   87 WRITE(6,88)(JKEY(MH),MH=ILO,IHI)
   88 FORMAT(20A4)
 1650 CONTINUE
      NJ=NJ+1
      READ(9,161)(INA(JW),JW=1,20)
  161 FORMAT(20A4)
      IF(INA(1)+437656601)1162,7000,1162
 1162 CALL JTIGET(N,NQ,JKEY,IKEY,LKEY,KKEY,INA,J,IGET,IANS,IXN,MAP,
     8MSTOR,ISTOR,JAP,JDZA,JDZB,JDZC,KDZA,KDZB,KDZC,MGA,MGB,MGC,JF)
C
C     JTIGET SEARCHES FOR STORAGE ANSWER, SIGNALS OUTCOME OF SEARCH
C            AND RECORDS THE ANSWER GIVEN IF ACCEPTABLE
C
  169 IF(IGET)1990,1990,170
  170 WRITE(6,180)(IAA(IG),IG=1,3),(IAD(IK),IK=KL,MN),(INA(JW),JW=1,20)
  180 FORMAT(3A4,'  ',4A4,'  ',20A4)
      GO TO 210
 1990 WRITE(6,1991)(IAA(IG),IG=4,6),(IAD(IK),IK=KL,MN),(INA(JW),JW=1,20)
 1991 FORMAT(3A4,'  ',4A4,'  ',20A4)
      IERROR=IERROR+1
      IF(IERROR-2)150,1992,3992
```

```
 1992 WRITE(6,1993)IERROR,(IAD(IK),IK=KL,MN)
 1993 FORMAT('OYOU HAVE GIVFN',I2,' UNACCEPTABLE REPLIES FOR ',4A4,//,'P
     8LEASE GIVE ACCEPTABLE REPLY THIS TIME')
      GO TO 150
 3992 WRITE(6,3993)(IAD(IK),IK=KL,MN),(INA(JW),JW=1,20)
 3993 FORMAT('1THE PROBLEM IS NOW PASSED TO THE ARBITER FOR CLASSIFICATI
     8N',//,'ERROR FOR ',4A4,//,'INPUT  = ',20A4,//,'PLEASE CLASSIFY COR
     8ECTLY')
      READ(9,4994)(INA(JW),JW=1,20)
 4994 FORMAT(20A4)
      GO TO 1162
 9000 CONTINUE
C
C     RECORD THAT THIS TOPIC HAS BEEN CONSIDEREDO MOVE ON TO NEXT
C
  210 NQ=NQ+1
      KL=KL+4
      MN=MN+4
 5067 CONTINUE
      WRITE(6,5068)
 5068 FORMAT('OCATEGORIZED DATA SECTION HAS NOW BEEN COMPLETED')
  189 IF(ICHK-44)220,289,220
C
C     PERSONAL NUMERIC DATA SECTION-CHECKED WITH RESPONDENT
C
  289 WRITE(6,5069)
 5069 FORMAT('1PERSONAL NUMERIC DATA SECTION',//,'CHECK THAT YOUR REPLIE
     8S HAVE BEEN CORRECTLY RECORDED')
      CALL JTDATE(IAD,IDAY,MTHS,IYRS,IAGE,JAGE,KAGE,IMY,JMY,
     8KMY,KYR,KK,IYST,IAB,IYC,IVY,JVY,KVY,KKK,IAC,IAX)
C
C     JTDATE HANDLES ALL PERSONAL NUMERIC DATA
C
  220 WRITE(6,221)
  221 FORMAT('1THANK YOU FOR YOUR COOPERATION',//,'YOU HAVE NOW COMPLETE
     8D THE QUESTIONNAIRE ',//,'SHIFT AND ENTER TO SHOW INSTRUCTIONS')
      KONTO=KONTO+1
 3000 CONTINUE
C
C     BRANCH FOR SPECIAL FACILITIES
C
 7000 WRITE(6,7001)KONTO,(IAD(IK),IK=25,28)
 7001 FORMAT('1',I4,' FILES HAVE BEEN PROCESSED TO-DATE',//,'THE FOLLOWI
     8NG FACILITIES ARE AVAILABLE',/,' 1.CONTINUE QUESTIONNAIRE PRESENTA
     8TION ENTER 11 (I2)  ',/,' 2.DISPLAY ALL PROCESSED FILES        EN
     8TER 33 (I2)  ',/,' 3.DISPLAY FILES BY ',4A4,'   ENTER 55 (I2) ',/,
     8' 4.DISPLAY A NAMED INDIVIDUAL''S FILE   ENTER 66 (I2)  ',/,' 5.DI
     8SPLAY FACTOR FREQUENCY COUNT         ENTER 77 (I2)  ',/,' 6.DISPLAY P
     8ROGRAM IRREGULARITIES LOG  ENTER 88 (I2)  ',/,' 7.TERMINATE RUN
```

```
      8                          ENTER 99 (I2)')
       READ(9,7850)MARK
  7850 FORMAT(I2)
       IF(MARK-33)3000,7007,7851
  7851 IF(MARK-66)7607,7705,7888
  7888 IF(MARK-88)7703,7400,9888
  7007 WRITE(6,7008)KONTO
  7008 FORMAT('THERE ARE ',I4,' FILES ARE AVAILABLEO ENTER THE NUMBER
      8 YOU REQUIRE (I4)')
       READ(9,7009)NFL
  7009 FORMAT(I4)
       IFL=1
       CALL JTSHOW(NFL,IXN,ITPC,IAD,NOTA,MEMO,MSTOR,IKEY,JKEY,IAGE,JAGE,
      8KAGE,KI,IAB,KYR,ISKP,KSKP,ICHK,ISTOR,MAST,IYC,IAC,IFL)
       GO TO 7000
C
C      JTSHOW DISPLAYS ALL,OR A SPECIFIED NUMBER,OF FILES IN ENTIRETY
C
  7607 WRITE(6,7608)
  7608 FORMAT('TO DISPLAY FULL FILES ENTER 10',//,'TO DISPLAY NAMES
      8ENTER 00')
       READ(9,7609)IYEL
  7609 FORMAT(I2)
       IF(IYEL)7620,7610,7620
  7610 IF(IXN)7620,7979,7620
  7620 CALL JTFILE(KDZA,KDZB,KDZC,MGA,MGB,MGC,KONTO,IYEL,
      8NFL,IXN,ITPC,IAD,NOTA,MEMO,MSTOR,IKEY,JKEY,IAGE,
      8JAGE,KAGE,KI,IAB,KYR,ISKP,KSKP,ICHK,ISTOR,MAST,IYC,IAC,IFL)
       GO TO 7000
  7705 IF(IXN)7715,7979,7715
  7715 WRITE(6,7706)
  7706 FORMAT('GIVE THE NAME OF THE INDIVIDUAL WHOSE FILE YOU REQUIRE IN
      8THE FORM    ',//,'SMITH,J,MR.')
       READ(9,7707)(INA(IV),IV=1,20)
  7707 FORMAT(20A4)
       MAX=KONTO*IXN
       CALL JTRCRD(INA,NOTA,MEMO,IFL,IOK,MAX)
       IF(IOK-100)7708,7709,7708
  7708 WRITE(6,1708)(IAA(IG),IG=4,6),(INA(IV),IV=1,10)
  1708 FORMAT(3A4,'  ',20A4)
       WRITE(6,1709)
  1709 FORMAT('OCHECK THAT YOU HAVE GIVEN THE CORRECT DESIGNATION
      8',//,'IF YOU WISH TO TRY AGAIN, OR REQUEST ANOTHER FILE
      8ENTER NEXT',//,'OTHERWISE ENTER XXXX')
       READ(9,1710)NOMEN
  1710 FORMAT(A4)
       IF(NOMEN+404232217)7705,7000,7705
  7709 WRITE(6,7710)(IAA(IG),IG=1,3),(INA(IV),IV=1,20)
  7710 FORMAT(3A4,'  ',20A4)
```

```
      MAST=98
      NFL=IFL
      ISKP=30
      KSKP=40
      ICHK=44
      CALL JTSHOW(NFL,IXN,ITPC,IAD,NOTA,MEMO,MSTOR,IKEY,JKEY,IAGE,
     8JAGE,KAGE,KI,IAB,KYR,ISKP,KSKP,ICHK,ISTOR,MAST,IYC,IAC,IFL)
      MAST=14
      WRITE(6,7711)
 7711 FORMAT('1TO REQUEST ANOTHER FILE ENTER NEXT',//,'OTHERWISE ENTER X
     8XXX')
      READ(9,7712)NEMO
 7712 FORMAT(A4)
      IF(NEMO+404232217)7705,7000,7705
 7979 WRITE(6,7980)
 7980 FORMAT('PERSONAL DATA HAS NOT BEEN COLLECTED',//,'HENCE THIS
     8FACILITY IS NON-OPERATIONAL')
      GO TO 7000
C
C     JTFILE DISPLAYS ALL PROCESSED FILES AS CLASSIFIED BY
C             THE SORTING FACTOR
C
 7703 CALL JTFACT(JDZA,JDZB,JDZC,KONTO,LKEY,JKEY,IKEY,IAD)
      GO TO 7000
C
C     JTFACT DISPLAYS A FACTOR FREQUENCY COUNT
C
 7400 WRITE(6,471)
  471 FORMAT('PROGRAM IRREGULARITIES LOG ',//,'THE FOLLOWING REPORTED TH
     8EMSELVES AS RETURNING STUDENTS',/,'YET COULD NOT LOCATE THEIR EXIS
     8TING FILE UNDER THE NAMES GIVEN BELOW')
      DO 4700 ILL=I,NUMB
      IF(ILL-1)4721,4720,4721
 4720 JLO=1
      GO TO 4722
 4721 JLO=IXXA(ILL-1)+1
 4722 JHI=IXXA(ILL)
      WRITE(6,4788)(JXXA(MZ),MZ=JLO,JHI)
 4788 FORMAT('0',20A4)
 4700 CONTINUE
      GO TO 7000
 9888 WRITE(6,9009)
 9009 FORMAT('YOU HAVE REQUESTED TERMINATION OF THE RUNO',//,'IF YOU WIS
     8H TO PROCEED WITH THIS COURSE ENTER 9999 OTHERWISE 1111')
      READ(9,9001)NAA
 9001 FORMAT(I4)
      IF(NAA-9999)7000,9999,9999
 9999 WRITE(6,9998)
 9998 FORMAT('RUN HAS NOW TERMINATED')
```

```
4          STOP
4          END
```

```
ND          SUBROUTINE JTRAND(NOTA,MEMO,NX,JX,NOTB,MEMB,NB,JB,JFL)
ND          DIMENSION NOTA(200),MEMO(500),NOTB(40),MEMB(200)
ND          IVK=(JFL*6)-5
ND          JVK=JFL*6
ND          DO 320 NB=IVK,JVK
ND          NX=NX+1
ND          IF(NB-1)1319,1319,1321
ND     1319 KG=1
ND          GO TO 1322
ND     1321 KG=NOTB(NB-1)+1
ND     1322 KH=NOTB(NB)
ND          KC=KH-KG
ND          NOTA(NX)=JX+KC
ND          DO 320 JB=KG,KH
ND          MEMO(JX)=MEMB(JB)
ND          JX=JX+1
ND      320 CONTINUE
ND          RETURN
ND          END
```

```
CRD         SUBROUTINE  JTRCRD(INA,NOTB,MEMB,JFL,IOK,MAX,IXN)
CRD         DIMENSION INA(20),NOTB(40),MEMB(200)
CRD         ILOLMT=1
CRD         IUPLMT=MAX
CRD         JFL=0
CRD         DO 6000 LMTS=ILOLMT,IUPLMT,IXN
CRD         JFL=JFL+1
CRD         IF(LMTS-1)110,100,110
CRD     100 ILO=1
CRD         GO TO 110
CRD     110 ILO=NOTB(LMTS-1)+1
CRD     120 IHI=NOTB(LMTS)
CRD         MS=1
CRD         DO 206 JD=ILO,IHI
CRD         IF(MEMB(JD)-INA(MS))201,205,201
CRD     205 MS=MS+1
CRD         IOK=100
CRD     206 CONTINUE
CRD         GO TO 70
CRD     201 IOK=00
CRD    6000 CONTINUE
CRD      70 RETURN
CRD         END
```

```
ACT          SUBROUTINE JTFACT(JDZA,JDZB,JDZC,KONTO,LKEY,JKEY,IKEY,IAD)
ACT          DIMENSION  LKEY(25),JKEY(500),IKEY(200),IAD(80)
ACT          WRITE(6,10)KONTO
ACT       10 FORMAT(I4,' FILES HAVE BEEN PROCESSED TO-DATE')
ACT          WRITE(6,20)(IAD(IK),IK=25,28)
ACT       20 FORMAT('0FACTOR FREQUENCY COUNT FOR ',4A4,' IS AS FOLLOWSO')
ACT          NRP=LKEY(1)
ACT          MRP=1
ACT          DO 1650 IXI=MRP,NRP
ACT          IF(IXI-1)1621,1620,1621
ACT     1620 ILO=1
ACT          GO TO 1622
ACT     1621 ILO=IKEY(IXI-1)+1
ACT     1622 IHI=IKEY(IXI)
ACT          IF(IXI-2)1,2,3
ACT        1 WRITE(6,81)JDZA,(IAD(IK),IK=25,28),(JKEY(JX),JX ILO,IHI)
ACT       81 FORMAT('0',I4,'=FREQUENCY FOR ',4A4,' CLASS 1 ',20A4)
ACT          GO TO 1650
ACT        2 WRITE(6,82)JDZB,(IAD(IK),IK=25,28),(JKEY(JX),JX=ILO,IHI)
ACT       82 FORMAT('0',I4,'=FREQUENCY FOR ',4A4,' CLASS 2 ',20A4)
ACT          GO TO 1650
ACT        3 WRITE(6,83)JDZC,(IAD(IK),IK=25,28),(JKEY(JX),JX=ILO,IHI)
ACT       83 FORMAT('0',I4,'=FREQUENCY FOR ',4A4,' CLASS 3 ',20A4)
ACT     1650 CONTINUE
ACT          RETURN
ACT          END
```

```
ILE        SUBROUTINE JTFILE(KDZA,KDZB,KDZC,MGA,MGB,MGC,KONTO,IYEL,
          8NFL,IXN,ITPC,IAD,NOTA,MEMO,MSTOR,IKEY,JKEY,IAGE,
          8JAGE,KAGE,KI,IAB,KYR,ISKP,KSKP,ICHK,ISTOR,MAST,IYC,IAC,IFL,IRI,
          8JRI,NRM)
ILE        DIMENSION KDZA(20),KDZB(20),KDZC(20)
ILE        DIMENSION IAD(80),NOTA(200),MEMO(500),IKEY(200),JKEY(500)
ILE        DIMENSION IAGE(25),JAGE(25),KAGE(25),IAB(24),KYR(50),MSTOR(120)
ILE        DIMENSION ISTOR(120),IAC(16)
ILE        IGA=MGA-1
ILE        IGB=MGB-1
ILE        IGC=MGC-1
ILE        WRITE(6,10)KONTO
ILE     10 FORMAT(I4,'=TOTAL OF FILES PROCESSED TO-DATE')
ILE        WRITE(6,1)IGA,(IAD(IK),IK=25,28)
ILE      1 FORMAT('0',I4,' OF WHICH ARE FROM ',4A4,' CLASS 1.')
ILE        WRITE(6,2)IGB,(IAD(IK),IK=25,28)
ILE      2 FORMAT('0',I4,' OF WHICH ARE FROM ',4A4,' CLASS 2.')
ILE        WRITE(6,3)IGC,(IAD(IK),IK=25,28)
ILE      3 FORMAT('0',I4,' OF WHICH ARE FROM ',4A4,' CLASS 3.')
ILE        IF(IGA)200,200,100
ILE    100 WRITE(6,102)(IAD(IK),IK=25,28)
ILE    102 FORMAT('1THE FILES FROM ',4A4,' CLASS 1. ARE AS FOLLOWSO')
ILE        IF(IYEL-10)104,103,104
ILE    104 I=IXN
ILE        K=KSKP
ILE        J=ICHK
ILE        IXN=1
ILE        KSKP=27
ILE        ICHK=22
ILE    103 DO 199 IAP=1,IGA
ILE        IFL=KDZA(IAP)
ILE        MAST=98
ILE        IFL=IFL-1
ILE        NFL=IFL
ILE        CALL JTSHOW(NFL,IXN,ITPC,IAD,NOTA,MEMO,MSTOR,IKEY,JKEY,IAGE
          8,JAGE,KAGE,KI,IAB,KYR,ISKP,KSKP,ICHK,ISTOR,MAST,IYC,IAC,IFL,IRI,
          8JRI,NRM)
ILE        IXN=I
ILE        KSKP=K
ILE        ICHK=J
ILE        MAST=14
ILE    199 CONTINUE
ILE    200 IF(IGB)300,300,201
ILE    201 WRITE(6,202)(IAD(IK),IK=25,28)
ILE    202 FORMAT('1THE FILES FROM ',4A4,' CLASS 2. ARE AS FOLLOWSO')
ILE        IF(IYEL-10)204,203,204
ILE    204 I=IXN
ILE        K=KSKP
ILE        J=ICHK
```

```
LE          IXN=1
LE          KSKP=27
LE          ICHK=22
LE      203 DO 299 IBP=1,IGB
LE          IFL=KDZB(IBP)
LE          MAST=98
LE          IFL=IFL-1
LE          NFL=IFL
LE          CALL JTSHOW(NFL,IXN,ITPC,IAD,NOTA,MEMO,MSTOR,IKEY,JKEY,IAGE,
           8JAGE,KAGE,KI,IAB,KYR,ISKP,KSKP,ICHK,ISTOR,MAST,IYC,IAC,IFL,IRI,
           8JRI,NRM)
LE          KSKP=K
LE          ICHK=J
LE          IXN=I
LE          MAST=14
LE      299 CONTINUE
LE      300 IF(IGC)500,500,301
LE      301 WRITE(6,302)(IAD(IK),IK=25,28)
LE      302 FORMAT('1THE FILES FROM ',4A4,' CLASS 3. ARE AS FOLLOWSO')
LE          IF(IYEL-10)304,303,304
LE      304 I=IXN
LE          K=KSKP
LE          J=ICHK
LE          IXN=1
LE          KSKP=27
LE          ICHK=22
LE      303 DO 399 ICP=1,IGC
LE          MAST=98
LE          IFL=KDZC(ICP)
LE          IFL=IFL-1
LE          NFL=IFL
LE          CALL JTSHOW(NFL,IXN,ITPC,IAD,NOTA,MEMO,MSTOR,IKEY,JKEY,IAGE,
           8JAGE,KAGE,KI,IAB,KYR,ISKP,KSKP,ICHK,ISTOR,MAST,IYC,IAC,IFL,IRI,
           8JRI,NRM)
LE          KSKP=K
LE          ICHK=J
LE          IXN=I
LE          MAST=14
LE      399 CONTINUE
LE          IYEL=13
LE      500 RETURN
LE          END
```

iii) Specimen of Ready-Supplied Data

```
/DATA
12
03
LOCAL
NON-LOCAL
OVERSEAS
17
ST. SALVATORS
ST. REGULUS
JOHN BURNET
SOUTHGAIT
HEPBURN
KINNESSBURN
DEANS COURT
ANDREW MELVILLE
UNIVERSITY
MCINTOSH
HAMILTON
ABBOTSFORD
WEST PARK
LODGINGS IN ST.A.
LODGINGS ELSEWHERE
PARENTAL HOME
FLAT
03
BRITISH
COMMONWEALTH
FOREIGN
05
NORMAL
PART TIME
STAFF
OVERSEAS
EXCHANGE
07
SCOTTISH EDUCATION DAPARTMENT
LOCAL EDUCATION AUTHORITY
U.K. UNIVERSITY
RESEARCH COUNCIL
BRITISH COUNCIL
INDUSTRY
OTHER
06
S.C.E.
G.C.E.
H.N.C.
FOREIGN QUALIFICATION
DEGREE
OTHER
04
ARTS
SCIENCE
MEDICINE
DIVINITY
07
HONOURS
ORDINARY
PH.D.
POSTGRADUATE DEGREE
POSTGRADUATE DIPLOMA
UNDERGRADUATE DIPLOMA
```

NON-GRADUATING
02
FULL TIME
PART TIME
16
ACTUARIAL SCIENCE
ANATOMY
APPLIED MATHEMATICS
ASTRONOMY
BIOCHEMISTRY
BIOLOGY
BOTANY
CHEMISTRY
COMPUTATIONAL SCIENCE
GEOLOGY
PURE MATHEMATICS
PHYSICS
PHYSIOLOGY
STATISTICS
THEORETICAL PHYSICS
ZOOLOGY
02
MALE
FEMALE
03
SINGLE
MARRIED
DIVORCED
9999
06
44
HARTLEY-TAYLOR,J,MR.
ANDREW MELVILLE HALL ST.AND 3759
THE COLLEGE,BRIGHTON.7.
J,HARTLEY-TAYLOR,ESQ.
RETIRED
21,CHATSWORTH COURT,PEMBROKE ROAD,KENSINGTON.LONDON.W.8. 01-937 4675
CLARK,P.P,MISS
14,GRANNIE CLARKES WYND*3592
NORTHAMPTON HIGH SCHOOL FOR GIRLS
JOSIAH CLARK,ESQ.
FARMER
25,LOMOND ROAD,LONG BUCKBY,NORTHAMPTON*DAVENTRY 3456
HOLLINGWORTH,V,M,MR.
JOHN BURNETT HALL
GLENALMOND SCHOOL
DR.W,G,HOLLINGWORTH
PHYSICIAN
45,HOPE STREET,ST.ANDREWS*4146
MCLACHLIN,D,M,MISS
UNIVERSITY HALL*ST.AND 4689
CHELTENHAM LADIES COLLEGE
J,G,MCLACHLIN,ESQ.
TEA PLANTER
BOX 45,KANDY,CEYLON
DEXTER,U,C,MR.
9,SHOREHEAD ST.ANDREWS*3129
HOLLAND PARK COMPREHENSIVE
T,R,DEXTER,ESQ,M.P,
CABINET MINISTER
56,MARLEBONE ROAD,LONDON.N.W.1.

iv) Listing of Archiving Programme and Sub-Routines

```
//          JOB           ,JT                  C.L.                  C4
//SYSC02 ACCESS      JTDISC,191='SA45V1'
//SYSC03 ACCESS      JTTAPE,280='SA0038'
//TEST    EXEC       FORTRAN(MAP)
C       ARCHIVING PROGRAM APPLIED TO BIBLIOGRAPHIC RECORDS
C
C
      DIMENSION  ITOP(10000),IWRD(20),IAV(80),INA(7),IKEY(100),JKEY(950)
     1,IAA(400),IAB(80),IAH(100,3),KNT(10),LST(10),INZ(7)
      DATA IBL/'.'/
      DEFINE FILE 2(120,360,L,KS)
      REWIND 3
C
C       READ OPERATION CODE
C
      READ(5,1000)IQUIZ
 1000 FORMAT(I2)
      IF(IQUIZ-33)1111,101,101
C
C       INITIALIZATION FOR FIRST RUN OF PROGRAM
C
 1111 J=1
      N=0
      IDV=1
 1100 DO 12 L=1,100
      IAH(L,1)=0
      IAH(L,2)=0
      IAH(L,3)=0
   12 CONTINUE
      IU=0
      IV=1
      IW=1
      IENTRY=0
      IE=1
      IP=2
  100 CONTINUE
      GO TO 1001
C
C       READ PARAMETER & ARRAY VALUES FROM DISC --SUBSEQUENT RUNS ONLY
C
  101 IDC=1
      READ(2'IDC)IW,IENTRY,N,J,IE
      NC=100
      IGA=2
      READ(2'IGA)((IAH(IY,M),IY=1,NC),M=1,3),(IKEY(IY),IY-1,100),(JKEY(I
     8Y),IY=1,950)
      IG=IE+1
      IDS=21
      READ(2'IDS)(ITOP(IX),IX-1,IC)
C
C       BRANCH FOR DISPLAY OR STORAGE OF NEW CATEGORIES
C
      IF(IQUIZ-55)1001,1001,77
 1001 CALL JTCATG(IKEY,JKEY,INA,N,J)
C
C       BRANCH FOR END OF PROGRAM IF OPERATION CODE = 33
C
      IF(IQUIZ-33)1,9000,1
C
C       READ IN OF BIBLIOGRAPHIC DATA
```

```
C
    1 KL=1
      MN=80
      KOUNT=0
C
C     SET FIELD TO BLANKS
C
      DO 175 JA=1,320
  175 IAA(JA)=IBL
  180 READ(5,7)(IAA(KC),KC=KL,MN)
 7    FORMAT(80A1)
      KOUNT=KOUNT+1
C
C     TEST FOR END OF RUN REQUEST
C
      IF(IAA(KL)-1547714624)200,9097,200
C
C     TEST FOR CONTINUATION OF READ REQUEST
C
  200 IF(IAA(MN)-1547714624)122,111,122
 111  KL=KL+80
      MN=MN+80
C
C     TEST FOR MAXIMUM NUMBER OF CARDS
C
      IF(KOUNT-3)180,180,122
C
C     WRITE BIBLIOGRAPHIC DATA TO TAPE
C
  122 DO 123 K1=1,320,80
      K2=K1+79
  123 WRITE(3) (IAA(JA),JA=K1,K2)
      IENTRY=IENTRY+1
C
C     READ CATEGORIES TO BE ATTACHED TO BIBLIOGRAPHIC DATA
C
  666 READ(5,4)(IAV(JB),JB=1,7)
    4 FORMAT(7A1)
      JC=1
C
C     TEST FOR END OF CATEGORY LIST MARKER
C
  617 IF(IAV(JC)-1547714624)618,1,618
C
C     CHECK CORRECTNESS OF CATEGORY DESIGNATION & RETURN CATEGORY CODE NUMB
C
  618 CALL JTFIND(IAV,JKEY,IKEY,NOGO,IH,N,JZ)
C
C     TEST FOR 'WRONG DESIGNATION' FLAG
C
      IF(NOGO-10)444,410,444
  410 WRITE(6,411)(IAV(JB),JB=1,7)
  411 FORMAT(' ERROR IN DESIGNATION OF CATEGORY ',7A1)
      GO TO 9000
  444 L=JZ
C
C     AUGMENT NUMBER OF ENTRIES RECORD FOR THIS CATEGORY BY 1
C
      IAH(L,1)=IAH(L,1)+1
```

```
C     TEST FOR FIRST ENTRY FOR CATEGORY
C
      IF(IAH(L,1)-1)511,510,511
C
C     RECORD BEGINNING VALUE ON ITCP LIST FOR THIS CATEGORY
C
  510 IAH(L,2)=IE
      ITCP(IE)=IENTRY
C
C     RECORD END VALUE ON ITCP LIST FOR THIS CATEGORY
C
      IAH(L,3)=IE
      IE=IE+2
      GO TO 555
  511 CONTINUE
C
C     RECORD VALUE OF ITCP ARRAY SUBSCRIPT APPROPRIATE FOR RECORDING VALUE
C     LAST ENTRY FOR THIS CATEGORY
C
      IPX=IAH(L,3)+1
C
C     RECORD IN ITCP ARRAY VALUE OF LAST ENTRY FOR THIS CATEGORY
C
      ITCP(IPX)=IE
C
C     RECORD CURRENT VALUE OF LAST ENTRY FOR THIS ARRAY
C
      IAH(L,3)=IE
C
C     RECORD NUMBER OF THIS ENTRY ON ITCP LIST
C
  213 ITCP(IE)=IENTRY
      IE=IE+2
  300 CONTINUE
  555 CONTINUE
      GO TO 666
C
C     TEST IF DISPLAY FACILITIES ARE REQUIRED
C
 9097 IF(IQUIZ-77)9000,77,9000
   77 CALL JTDISP(IAV,JKEY,IKEY,NOGO,IH,N,IAH,ITCP,JZ)
C
C     WRITE PARAMETER & ARRAY VALUES TO DISC FOR STORAGE
C
 9000 IDT=1
      WRITE(2'IDT)IW,IENTRY,N,J,IE
      NT=100
      IDW=2
      WRITE(2'IDW)((IAH(IT,MB),IT=1,NT),MB=1,3),(IKEY(IY),IY=1,100),(JKE
     8Y(IY),IY=1,950)
      IGT=IE+1
      IDZ=21
      WRITE(2'IDZ)(ITCP(IXT),IXT=1,IGT)
      STOP
      END
      SUBROUTINE JTCATG(IKEY,JKEY,INA,N,J)
      DIMENSION IKEY(100),JKEY(950),INA(7)
C
C     READ NUMBER OF CATEGORIES TO BE RECORDED
C
```

```
      READ(5,1)ICTG
 1    FORMAT(I4)
      DO 10 JA=1,ICTG
C
C     READ CATEGORY NAME
C
      READ(5,2)(INA(IA),IA=1,7)
 2    FORMAT(7A1)
      N=N+1
      KG=7
C
C     TEST FOR  AND  ELIMINATE BLANKS
C
  115 IF(INA(KG)-1077952576)110,105,110
 105  KG=KG-1
      GO TO 115
C
C     RECORD POINTER POSITION FOR END OF WORD
C
  110 IKEY(N)=J+KG-1
C
C     TRANSFER INDIVIDUAL CHARACTERS OF CATEGORY NAME TO JKEY ARRAY FOR ST
C
      DO 320 LB=1,KG
      JKEY(J)=INA(LB)
      J=J+1
 320  CONTINUE
 10   CONTINUE
      RETURN
      END
      SUBROUTINE JTFIND(IAV,JKEY,IKEY,NCOC,IH,N,JZ)
      DIMENSION    IAV(80),JKEY(950),IKEY(100)
      DO 700 JZ=1,N
C
C     ESTABLISH LOWER & UPPER LIMITS OF SEARCH
C
      IF(JZ-1)21,22,21
   22 ILOLMT=1
      GO TO 33
   21 ILOLMT=IKEY(JZ-1)+1
   33 IUPLMT=IKEY(JZ)
      MS=1
C
C     TEST FOR MATCH BETWEEN STORAGE LIST & INPUT LIST OF CATEGORIES
C
      DO 206 JD=ILOLMT,IUPLMT
      IF(JKEY(JD)-IAV(MS))201,205,201
 205  MS=MS+1
C
C     SET 'CORRECT DESIGNATION' FLAG
C
      NCOC=0
 206  CONTINUE
      GO TO 798
C
C     SET 'WRONG DESIGNATION' FLAG
C
 201  NCOC=10
 700  CONTINUE
  798 CONTINUE
```

```
      RETURN
      END
      SUBROUTINE JTDISP(IAV,JKEY,IKEY,NCGC,IH,N,IAH,ITOP,JZ)
      DIMENSION IAV(80),JKEY(950),IKEY(100),IAA(400),IAH(100,3),ITOP(1)
      DIMENSION KK(6),JEV(6),JUV(6)
      DATA IBLANK/' '/
  199 I=0
      KL=1
C
C     READ CATEGORIES TO BE INCLUDED IN CATEGORY SEARCH LIST
C
  200 READ(5,201)(IAV(IQ),IQ=1,7)
  201 FORMAT (7A1)
C
C     TEST FOR END OF RUN REQUEST
C
      IF(IAV(1)-1547714624)202,203,202
  203 IF(IAV(2)-1547714624)206,999,206
C
C     JT FIND CHECKS CORRECT DESIGNATION OF CATEGORY & RETURNS CATEGORY CO
C
  202 CALL JTFIND(IAV,JKEY,IKEY,NCGC,IH,N,JZ)
C
C     TEST FOR FAILURE TO LOCATE  CATEGORY
C
      IF(NCGC-10)900,999,900
  900 I=I+1
      KK(I)=JZ
C
C     ESTABLISH LOWER AND UPPER LIMITS OF SEARCH FOR COMMON ENTRIES
C
      JEV(I)=IAH(JZ,2)
      JUV(I)=IAH(JZ,3)
      GO TO 200
  206 DO 395 M=1,80
  395 IAV(M)=IBLANK
      MM=1
      DO 396 M=1,I
  398 MN=KK(M)
      ML=1
      IF(MN.EQ.1) GO TO 397
      ML=IKEY(MN-1)+1
  397 MU=IKEY(MN)
  399 IAV(MM)=JKEY(ML)
      MM=MM+1
      ML=ML+1
      IF(ML.LE.MU) GO TO 399
  396 MM=MM+2
      WRITE(6,394) (IAV(M),M=1,80)
  394 FORMAT('1',' PUBLICATIONS LISTED UNDER CATEGORIES ',80A1,//)
  205 JZ=JEV(1)
      IFIND=ITOP(JZ)
      IF(I.EQ.1)GO TO 300
C
C     COMPARE ENTRIES FOR OTHER CATEGORIES WITH ENTRIES IN FIRST CATEGORY
C
      DO 207 M=2,I
  209 MM=JEV(M)
      IF(IFIND.EQ.ITOP(MM))GO TO 207
      IF(IFIND.LT.ITOP(MM))GO TO 208
```

```
      IF(MM.EQ.JUV(M))GO TO 213
      JEV(M)=ITOP(MM+1)
      GO TO 209
  207 CONTINUE
C
C     TEST IF TAPE IS TO BE STEPPED PRIOR TO READING
C
  300 IF((IFIND-1)*4+1.EQ.KL)GO TO 210
      KU=(IFIND-1)*4
C
C     STEP TAPE WITH DUMMY READ
C
      DO 211 M =KL,KU
  211 READ(3)
C
C     READ THE ENTRY IN BIBLIOGRAPHIC RECORD
C
  210 DO 250 K1=1,320,80
      K2=K1+79
  250 READ(3) (IAA(JA),JA=K1,K2)
C
C     WRITE CONTENTS OF BIBLIOGRAPHIC RECORD FOR DISPLAY
C
      WRITE(6,212)(IAA(JA),JA=1,320)
  212 FORMAT('0',80A1)
C
C     AUGMENT RECORD COUNTERS IN READINESS FOR NEXT SEARCH
C
      KL=4*IFIND+1
  208 JEV(1)=ITOP(JZ+1)
      IJ=JUV(1)
      IF(IFIND.EQ.ITOP(IJ))GO TO 213
      GO TO 205
  213 REWIND 3
      GO TO 199
  999 CALL EXIT
      END
/*
//      EXEC    CLOADER
/*
77
ALGRTHM
*
TECHNQS
*
STNDRDS
TECHNQS
*
EDUCATN
*
STNDRDS
ALGRTHM
TECHNQS
*
TECHNQS
STNDRDS
ALGRTHM
*
STNDRDS
ALGRTHM
```

```
TECNNQS
*
**
/*
/8
```