RECOGNITION OF ENGLISH CHARACTERS

by

SYNTACTIC METHOD OF PATTERN RECOGNITION

By

Md. Mashuqur Rahman

A Thesis

submitted to the Department of Electrical and Electronic Engineering, Bangladesh University of Engineering and Technology, Dhaka, in partial fulfilment of the requirements for the degree

of

MASTER OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING

July 1985.

To

my parents

DECLARATION

I do hereby declare that neither this thesis nor any part thereof has been submitted or is being concurrently submitted in candidature for any degree at any other university.

Candidate

CERTIFICATE OF RESEARCH

Certified that the work presented in this Thesis is the result of the investigation carried out by the the candiate under the supervision of Dr. A. K. M. Mahfuzur Rahman Khan at the Department of Electrical and Electronic Engineering, BUET.,Dhaka.

Supervisor

Candidate

Accepted as satisfactory for partial fulfilment of the requirements for the degree of M. Sc. Engineering in Electrical and Electronic Engineering.

Board of Examiners

(Dr. A.K.M.Mahfuzur Rahman Khan)
Professor & Head
Department of Computer Engineering
B.U.E.T.,Dhaka, Bangladesh.

Chairman
(Supervisor)

(Dr. Syed Fazl-i-Rahman)
Professor & Head
Department of Electrical & Electronic Engineering, B.U.E.T.,Dhaka,Bangladesh.

Member

(Dr. Shamsuddin Ahmed)
Professor & Dean
Faculty of Electrical & Electronic Engineering, B.U.E.T.,Dhaka, Bangladesh.

Member

(Mr. M. I. Ahmed)
Assistant Professor
Department of Electrical & Electronic Engineering, B.U.E.T.,Dhaka, Bangladesh.

Member

(Dr. A. M. Choudhury)
Director (Research)
Space Research & Remote Sensing Organization, Agargaon, Dhaka, Bangladesh.

Member
(External)

ACKNOWLEDGEMENTS

Keen interest of Dr. A.K.M. Mahfuzur Rahman Khan in the field of Pattern Recognition has influenced the author to carry out a research work in this field for the first time in this department. The author expresses his deepest gratitude to Dr. Khan for his constant guidance and supervision to make the idea work.

Thanks are due to Mr. M. I. Ahmed for his all along encouragements and valuable suggestions; without which the idea would never have materialised.

The author also expresses his sincere gratitude to Dr. A.M. Patwari for his keen interest and encouragements for the development of research works in this institute.

Thanks are also due to Mr. M.A. Jalil for typing the thesis and Mr. Mafizur Rahman for drafting and art work.

ABSTRACT

A syntactic pattern recognition scheme for recognising English alphabetic characters of block capital type has been suggested. The syntactic approach choosen as the basis for the analysis not only recognises the character but also gives an idea about the structural description of it.

The scheme operates on the representation of one character at a time. The representation is in the form of a matrix whose entries have levels '0' or '1' corresponding to white or black in the original picture.

The scheme draws the outline of the pattern, then resolves the outline into a number of primitives. The process for resolving the pattern into primitives is automatic and the number of resultant primitives depends upon the number of sharp change in direction of the border line of the pattern. Next step in the scheme is to recognise the primitives and finally a decision is made depending upon the relationships among the primitives.

A FORTRAN IV computer program has been developed to implement the scheme in a IBM 4331 machine.

CONTENTS

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

1.1 GENERAL

The problem of pattern recognition usually denotes a discrimination or classification of a set of processes or events[1]. The set of processes or events to be classified could be a set of physical objects or a set of mental states. The number of pattern classes is often determined by the particular application in mind. For example, in the problem of English character recognition, the problem is of 26 classes. On the other hand, discriminating English characters from Bengali characters is a two class problem.

Pattern recognition is a major area of activity that encompasses the processing of pictorial information obtained from interaction betwean science and society, and need for the people of communicate with the computing machines in their natural mode of communication (the human voice and hand written script)[2]. Pattern recognition scientists are also concerned with the idea of designing and making automata that can hear and understand what we human being say and write, the automata that can speak and make people understand, and the automata that can process pictorial information for human use with more and more efficiency. The research in pattern recognition encompasses the fields of communication and computer science, mathematics and statistics, acoustics, phonetics, linguistics and psycholinguistics, speech pathology, haematology, neurophysiology, and radiology, remote sensing techniques and photogrammetry.

Research and developments on pattern recognition methods and applications may be classified into following groups:

(i) Man-Machine communication

(ii) Bio-medical applications and diagnosing pathological conditions by analysing medical X-rays or cytological slides

(iii) Natural resources estimation and planning in agriculture, forestry, hydrology, geology and environment

(iv) Scientific and Military applications

and (v) Detection of crimes and criminals.

1.2 APPROACHES TO PATTERN RECOGNITION

The many different mathematical techniques used to solve pattern recognition problems may be grouped into two general approaches; namely, the decision theoretic (or statistical) approach and the syntactic (or linguistic) approach.

1.2.1 Decision Theoretic Approach

In the decision theoretic approach the classification is based on a set of selected measurements, extracted from the input pattern[1,2]. These selected measurements are called 'features'. The recognition of each pattern (assignment to a pattern class) is usually made by partitioning the feature space. Once a pattern is transformed through feature extraction, to a point or a vector in the feature space, its characteristics are expressed only by a set of numerical values. The information about the structure of each pattern is either ignored or not explicitly represented in the feature space. Most of the developments in pattern recognition research during the last decade deal with decision theoretic approach. Applications include character recognition, crop classification, medical diagnosis, classification of electrocardiograms etc.

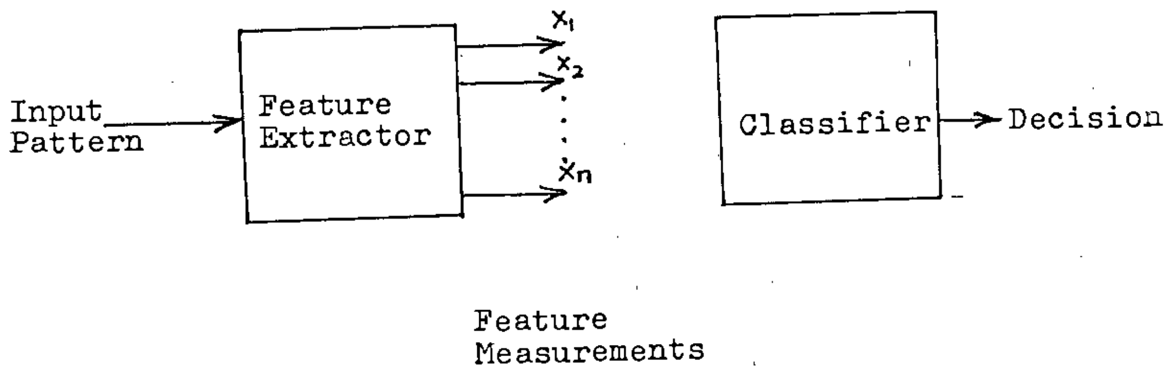


Fig 1.1: Block diagram of a pattern recognition system using decision theoretic approach.

1.2.2 Syntactic Approach

In some pattern recognition problems, the structural information which describes each pattern is important and the recognition process includes not only the capability of assigning the pattern to a particular class (to classify it), but also the capacity to describe specific aspects of the pattern which make it ineligible for assignment to another class[3]. The syntactic approach views patterns as complexes of primitive structural elements, called morphs[4]. A pattern is classified by studying the set of morphs which build up a pattern and studying the relationships among the morphs. This method has been successfully applied in problems of character recognition, chromosome analysis, finger-print classification, X-ray analysis, speech analysis, etc.

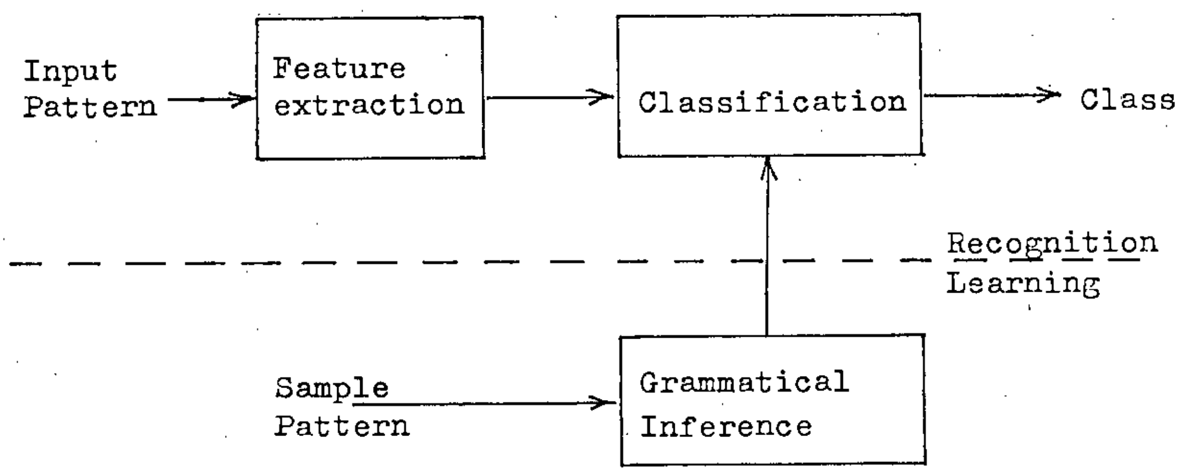


Fig 1.2 : Block diagram of a syntactic pattern recognition problem.

1.3 DICHOTOMY OF SYNTACTICAL AND STATISTICAL APPROACHES:

In the past much has been made of the aparent difference between the two approaches[4]. The stress on the distinction between the two models hides many similarities : most of the pre-processing techniques are usefully applied in both the approaches; feature extraction and selection in decision theoretic approach and morph extraction and selection in syntactic approach are similar in nature.

The basic difference between the two approaches is that in decision theoretic approach the features are a set of numerical measurements on the pattern, subpattern parameters whereas the morphs in syntactic model are sub-patterns themselves. Statistical decision theory focuses entirely on statistical relationships among scalar features ignoring any other structural properties that characterize patterns i.e. decision theoretic approach only classifies the pattern whereas syntactic approach serves classification as well as description of the pattern.

1.4 A COMPARISON OF ANALOG AND DIGITAL TECHNIQUES FOR PATTERN RECOGNITION:

The digital approach to problems in pattern recognition has many advantages[5]. Digital computers provide the user with the capability of performing calculations to essentialy any degree of precision with almost infinite flexibility as regards the type and scope of the problem addressed. Due to the universality of most major programming languages and general availability of digital computing facilities, the user also benefits from both ease of programming and the transferability of software. Last but not the least, the digital computer usually offers the user absolute repeatability on each execution of a given program. These are the advantages which have led to an almost overwhelming preference for the use of digital computers in carrying out calculations relating to pattern recognition.

Analog computer offers workers using low precision high-speed one dimensional or two dimensional linear discriminant analysis a significant advantage in hardware performance (equivalent bits per sec per dollar) over the digital computer in certain limited but important areas[5]. These areas include finger print identification, word recognition, chromosome spread detection, earth resources and land use analysis and broad band radar signal analysis. Although at present the analog computer offers significant advantages in certain fields, this advantages will eventually be overcome by the digital computer[5].

1.5 LITERATURE SURVEY

Over the years, the field of pattern recognition has attracted workers from a variety of areas such as engineering, system theory, statistics, linguistics, psychology, etc., resulting in a vast literature containing abstract mathematical approaches as well as highly pragmatic techniques. The literature is scattered in a large number of journals in several fields. At least three IEEE journals (SMC, COMP and Inf. Th.) regularly publish pattern recognition papers. Some of the important literatures, relevant to the present work, are discussed in this section.

L.N.Kanal has presented[4] an excellent overview of machine pattern recognition discussing the status of various aspects of the field as of 1974 and putting different approaches, techniques and trends in perspective. This paper also provides a very useful bibliography.

A.W.Holt in a paper[6] classifies character recognition machinery with a minimum reference to the specific components used. According to Holt the job of all such machines is to convert a set of data having high information content into a character name having a much lower information content. In this paper he gave a description of single stage, two stage and three stage character recognition machines. He is the first man to use stage concept in classifying character recognition machinery.

In his paper[5] Kendall. Preston Jr. gave a thorough comparison of analog and digital techniques used for pattern recognition. This paper reviews three major catagories (electronic, acoustical and optical) of analog technology used in pattern recognition and predicts the future trends upon the analysis of performance advances which have taken place in both digital and analog fields during the past decade.

W.W.Stallings describes[7] an approach to chinese character recognition, based on a formal model of the pictorial structure of chinese characters. He developed a program which produces a description of a character on two levels : i) the internal structure of each connected part of the character, and ii) the arrangement in two dimensions of the connected parts.

K.S.Fu a pioneer worker in this field has written as well as edited a number of books on pattern recognition. He has also published a number of papers at different times. In a paper[1] Fu has elaborately discussed the different approaches to pattern recognition whereas in another paper[3] he has studied syntactic method of pattern recognition at length. He also discussed[7] the grammatical inference for syntactic pattern recognition basing on a set of sample patterns,

1.6 SCOPE OF THE PRESENT WORK :

It has already been mentioned that, one of the main aspects of pattern recognition is to communicate with the computing machines in the natural mode of communication (the human voice and handwritten script). Research works are going on in different ways of man-machine communication, such as speaker recognition, speech analysis, finger print identification, character recognition, etc. Character recognition has been receiving considerable attention as the result of the phenomenal growth of office automation and the need for translating human language into machine language. Most workers in the field followed decision theoretic approach which is generally attractive only for classification of patterns ignoring structural information. Syntactic approach has been choosen for the present work which classifies the pattern as well as gives a structural description of it. A computer program has been developed to work in steps, first for drawing the outline of the character, next for selecting nodes, third for recognising morphs in between successive nodes and finally for converting characters into digital codes representing the relationship among the morphs which build the character.

CHAPTER 2

SYNTACTIC METHOD

2. SYNTACTIC METHOD

2.1 SYNTACTIC METHOD AS THE BASIS FOR ANALYSIS

Syntactic method has been utilised as the basis for pattern analysis in this work. In syntactic method patterns are considered to be built up out of subpatterns (called morphs) in various ways of composition in the same fashion as phrases and sentences are built up by concatenating words and words are built up by concatenating characters. Obviously, the simplest subpatterns i.e. the morphs should be much easier to recognise than the patterns themselves.

A syntactic pattern recognition system can be considered as consisting of three major parts; namely, preprocessing pattern description or representation and syntax analysis. A simple block diagram of the system is shown in fig 2.1.

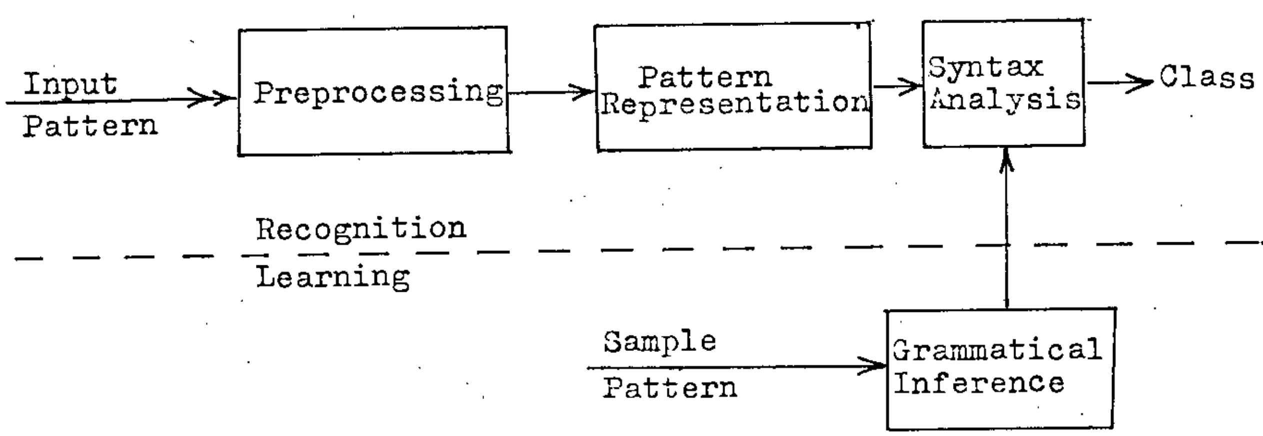


Fig 2.1 : Block diagram of a syntactic pattern recognition system.

2.1.1 Preprocessing

The functions of preprocessing include (i) pattern encoding or approximation, and (ii) filtering, restoration and enhancement. An input pattern is first coded or approximated by some convenient form for further processing. For example, a black and white picture can be coded in terms of a matrix of 0's and 1's. Techniques of filtering, restoration and enhancement will be used to make the pattern clean of noise, and to improve the quality of the coded patterns. At the output of the preprocessor, patterns with reasonably good quality are obtained.

2.1.2 Pattern Representation

Pattern representation stage consists of (i) Pattern segmentation and (ii) Morph extraction. Each preprocessed pattern is now segmented into primitive structures (morphs). The morphs are identified and each pattern is represented by a set of primitives with specified synatactic operations. For example, in terms of concatenation operation, each pattern is represented by a string of concatenated morphs.

2.1.3 Syntax Analysis

In order to have a grammar describing the structural information about the class under study, a grammatical inference machine which can infer grammar from a set of training patterns is to be used. The structural description of a class of pattern under study is learned from actual sample patterns from that class. The pattern under study is now compared with the grammars inferred from sample patterns and given the class to which its grammar matches. If the pattern matches to none of the classes, it is rejected as ambiguous.

2.2 SELECTION OF MORPHS

The first step in formulating a syntactic model for pattern description is the determination of a set of morphs, interms of which the patterns of interest may be described. This will be largely influenced by the nature of the data, the specific application in mind, and the technology available for implementing the system. There is no general principle for selection of morphs. The following requirements usually serve as a guideline for the purpose.

i) The morphs should serve as basic pattern elements to provide a compact but adequate description of the data interms of the specified structural relations

ii) The morphs should be easily extractable and recognizable.

Method used in the next chapter is based on the principles developed above.

CHAPTER 3

DEVELOPMENT OF THE PRESENT METHOD

3. DEVELOPMENT OF THE PRESENT METHOD

3.1 INTRODUCTION

The program developed for the present analysis operates on a representation of one character at a time. The representation is in the form of a matrix whose entries have level '0' or '1' corresponding to white or black pixels in the original picture. The matrix may be obtained by a flying spot scanner or a camera-digitizer combination. The characters to be used may be taken from a number of different sources.

Fig 3.1 shows digitized form of character A; white points are shown as black spaces instead of O's while black points are shown by the level 1.

Partially coloured pixels with 50% or more blacks are considered fully black, and those with less than 50% black are considered fully white. This results in jigjag appearance for all sloping lines and make sharp bend appear curved. However for recognition purposes this may not affect in any way.

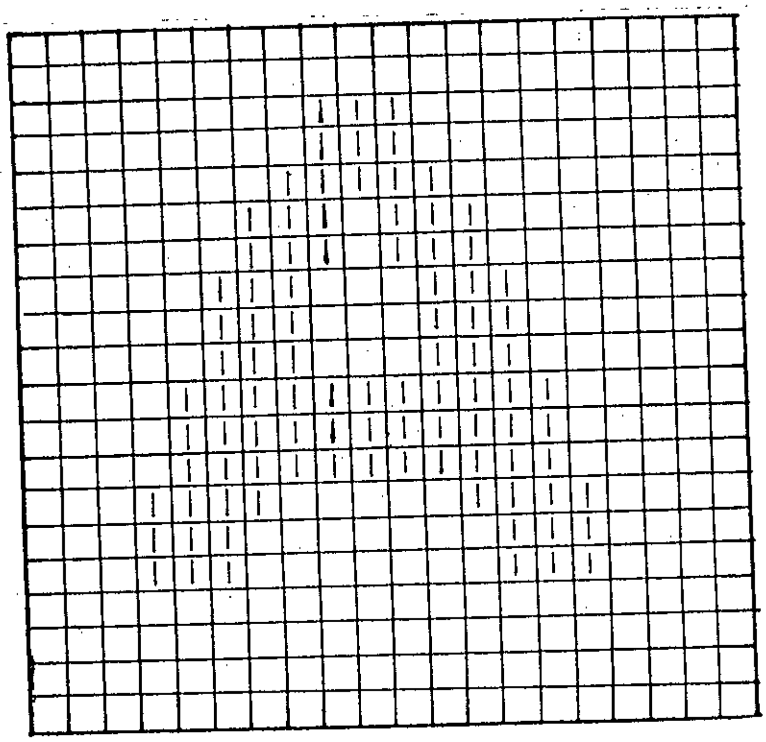


Fig 3.1 : Digitized representation of character A.

3.2 SELECTION OF MORPHS

In syntactic method each input pattern is to be resolved into primitive structural elements, called morphs. The first step in designing a syntactic model is the selection of morphs in terms of which the patterns of interest can be represented. In selecting morphs care must be taken so that they are simple to recognise and they are minimum in number. For the present analysis the following eight simple strokes were choosen to represent all the 26 English characters. To each morph a wide range of deviation is allowed (This has been discussed in details in Chapter 4).

Table 3.1 : Morphs used for recognising English Characters

Morph	Characteristic	Numeric Code
	Horizontal line	1
	Vertical line	2
	Positive Slope line	3
	Negative Slope line	4
	Vertical Concave Curve	5
	Vertical Convex Curve	6
	Horizontal Concave Curve	7
	Horizontal Convex Curve	8

3.3 ANALYSIS OF THE INPUT

The method of analysis consists of the following steps:

1. Tracing the border line of a character
2. Selection of node on the border line
3. Recognising the morph in between the nodes

4. Representing the character by numeric codes corresponding to the combination of morphs.

Figure 3.2 shows a block schematic for the process of analysis.

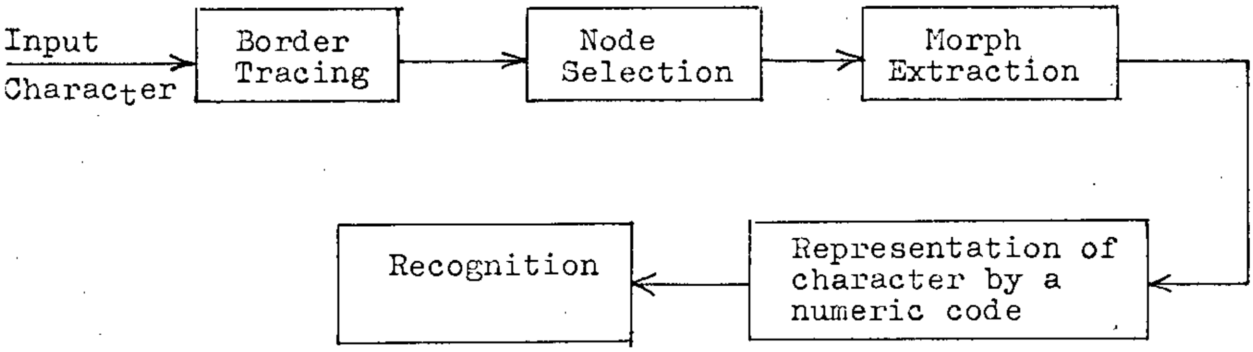


Fig 3.2 : Block diagram showing the steps in present analysis.

The input appears as 0's and 1's given as levels to the m x n matrix elements that represent the pixels making a pattern. Where m is the number of pixels on a horizontal row and n is the number of pixels allowed on a vertical column.

3.3.1 Border Line Tracing

The first step towards pattern representation is to trace the border line of the pattern. For tracing and drawing the border line of a pattern, "Border Following Technique" developed by Dr. Appana C. Chottera and Dr. Shridhar was used[8].

For locating the black points (1's) within a white background (0's) a program is developed (see Appendix - C) for scanning each pixel of the digitized pattern row-wise (or column-wise) to detect a transition from white to black as shown in fig 3.3.

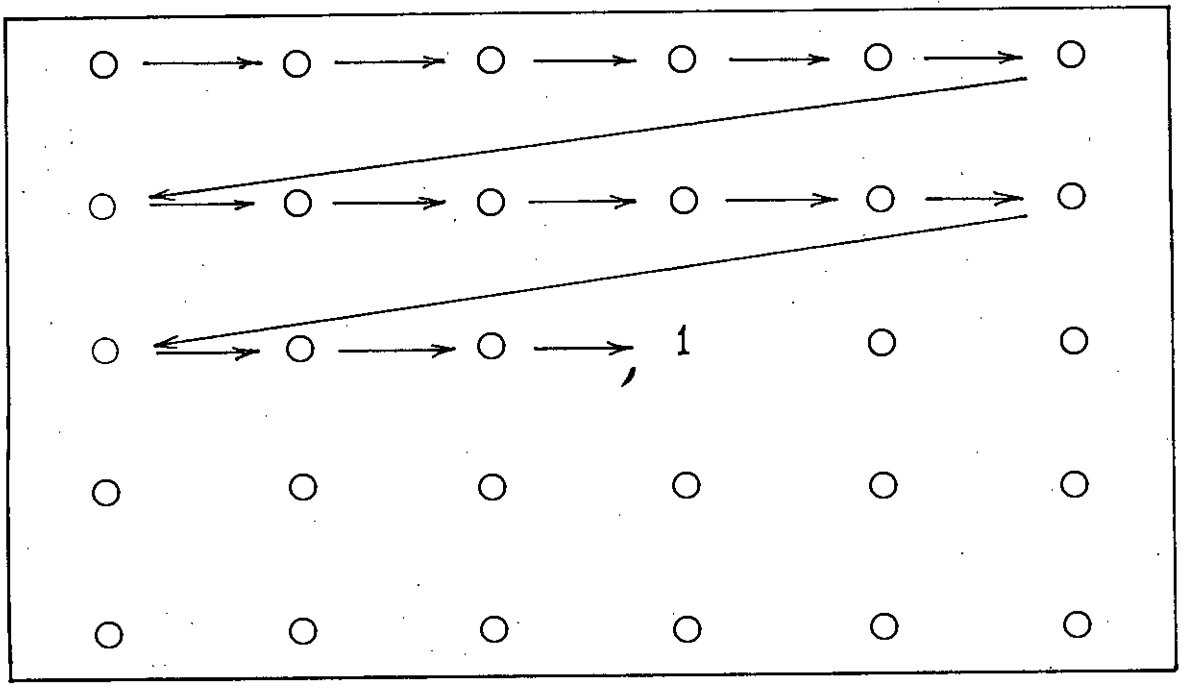


Fig 3.3 : Row-wise scanning in search of a transistion from white to black.

Once the first black point is detected its level is changed from '1' to '2'. Searching for the next black point now starts from the pixel occuring earlier in the scanning process. The scanner moves clock-wise around the black point in search of the second black point (fig 3.4). It is obvious that if the first black point is not an isolated one, the second border point will lie in one of the seven other pixels sorrounding the first one point. The level of the

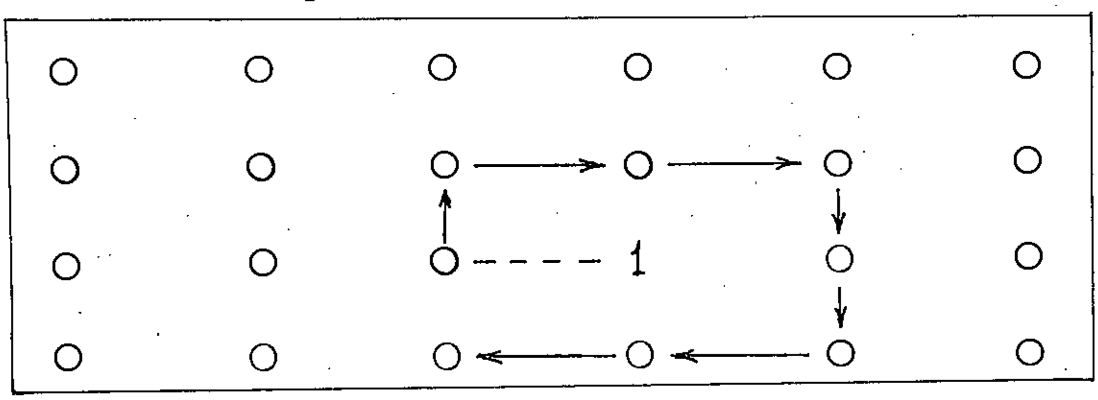


Fig 3.4 : Clock-wise search around the first black point for a second one.

second border point is also changed to '2'. Following the same procedure, search is now made around the second black point to find the third one. In this process all the black points are traced, and their levels are changed to '2'. After tracing the border line, character A of fig 3.1 becomes as shown in fig 3.5.

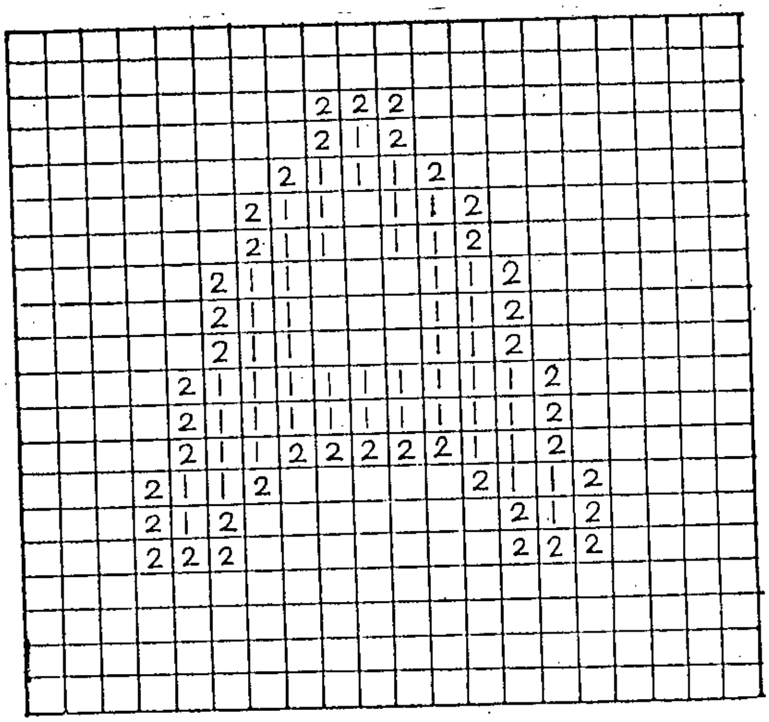


Fig 3.5 : Character A, after its border line is drawn.

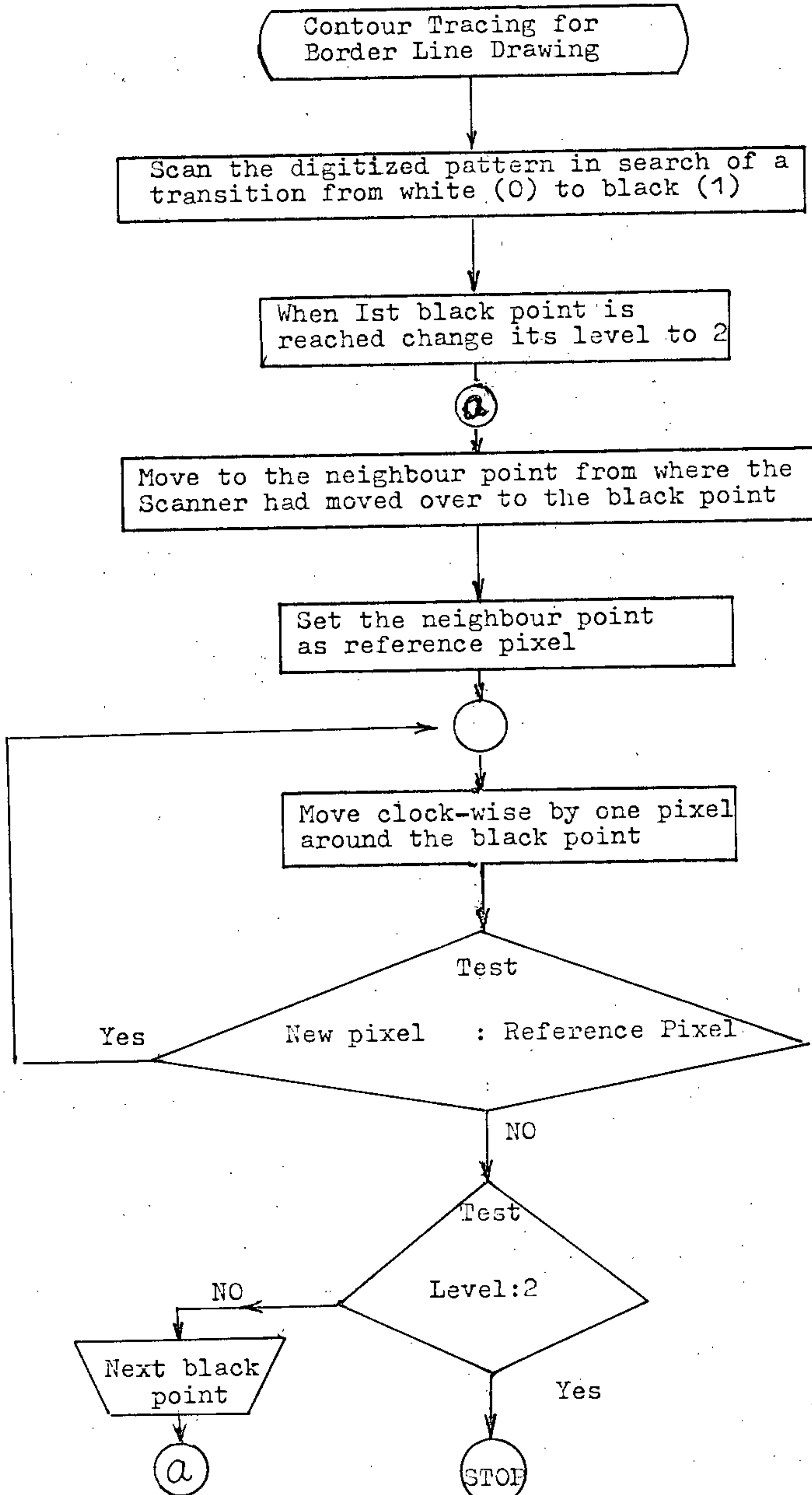


Fig: 3.6 - Flow diagram for Border line drawing.

3.3.2 Selection of Nodes

After drawing the border line of the input character, the task is to find the nodes. Nodes are formed at points where there is a sharp change of direction of 90 degree or less and 270 degree or higher. The conditions under which a border point will become a node are explained below with simple diagrams.

Test 1: Test 1 is performed for checking a sharp bend of 270 degree or more. The scanner moves to the border point (IP1, IP2) from the first neighbour point (NX1, NY1). Now it is checked whether

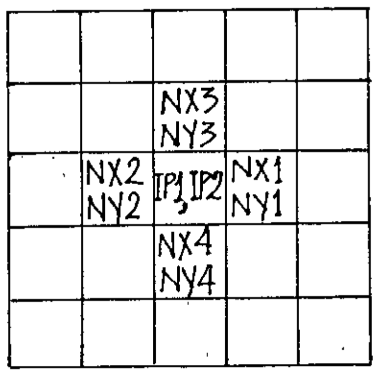


Fig 3.7 : Test 1. for finding a node with a change of direction of 270° or more.

the second neighbour point (NX2, NY2) is a border point. If (NX2, NY2) is not a border point then (IP1, IP2) is not a node (fig 3.7). If the second neighbour point is a border point, the third neighbour point (NX3, NY3) and the fourth neighbour point (NX4, NY4) are checked. If any one of the two is found to be a border point then the border point (IP1, IP2) is a node otherwise test 2 is performed to take final decision.

Test 2: Test 2 finds a sharp bend of 90 degree or less. If test 1 fails the scanner moves from the border point (IP1, IP2) to the first neighbour point (NX1, NX2). Now it is checked whether both the pixels (AX1, AY1) and (BX1, BY1) on the other two sides of (NX1, NY1) are border points. If they are, then the border point (IP1,IP2) is a node having a change of direction of 90 degree or less (fig 3.8).

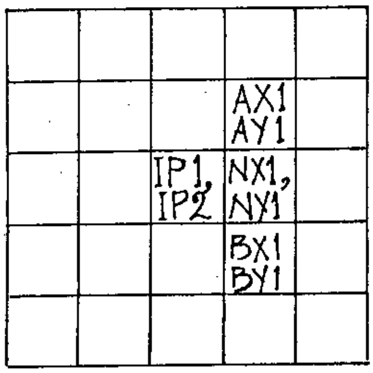


Fig 3.8 : Test 2 for finding a node with a change of direction of 90 degree or less.

If both the tests fail , the border point under consideration is not a node.

For generating co-ordinates of the second, third and fourth neighbour points for test 1 and AX1, AY1, BX1, BY1 for test 2, the following relations are being used :

Border point		:	IP1, IP2
Ist neighbour point		:	NX1, NY1
Set	K1	:	IP1 - NX1
	K2	:	IP2 - NY1

2nd neighbour point : NX2 = IP1 + K1

NY2 = IP2 + K2

3rd neighbour point : NX3 = IP1 + K2

NY3 = IP2 + K1

4th neighbour point : NX4 = IP1 − K2

NY4 = IP2 − K1

For test 2 the co-ordinates are

AX1 = NX1 + K2

AY1 = NY1 + K1

and BX1 = NX1 − K2

BY1 = NY1 − K1

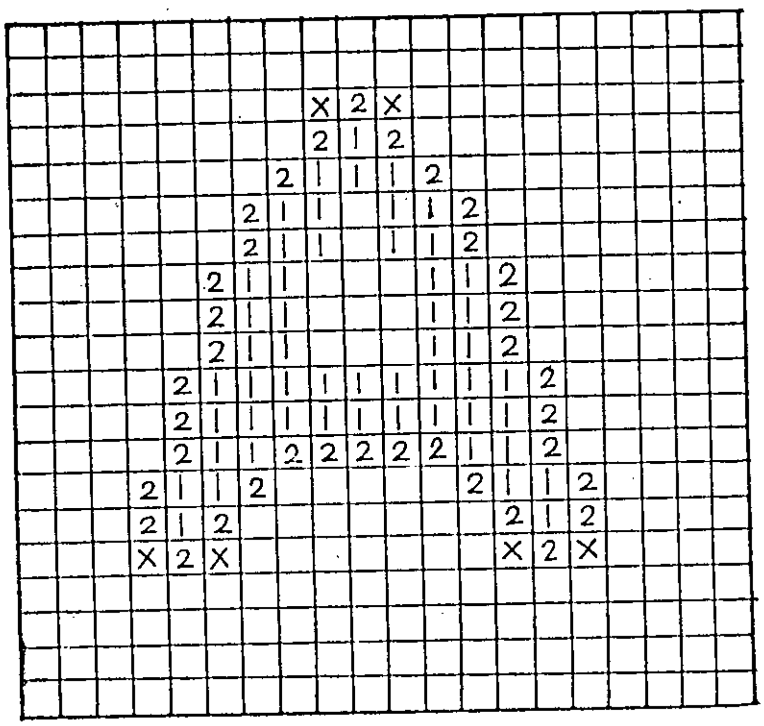


Fig 3.9 : Nodes in character A shown by cross (X).

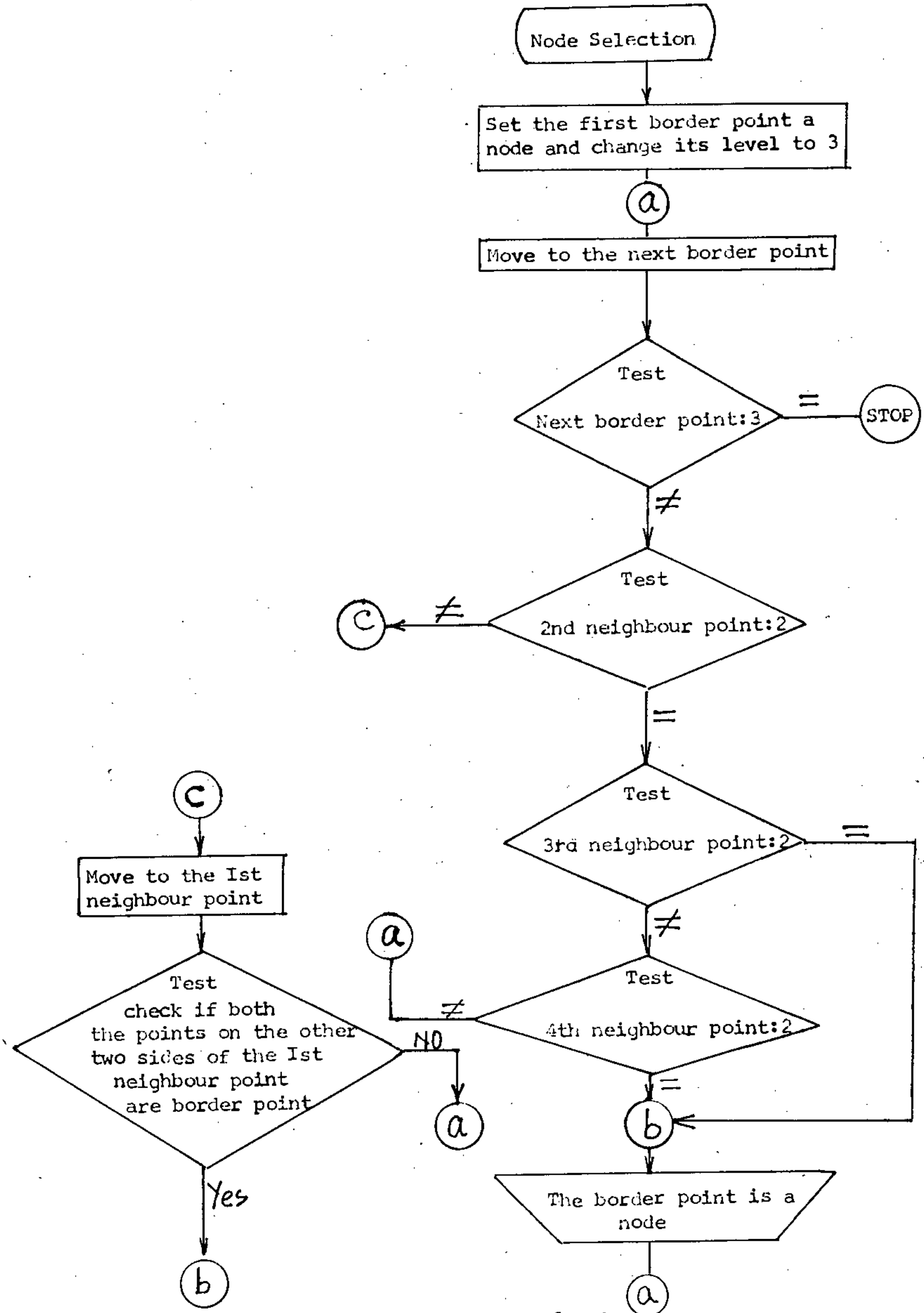


Fig 3.10 : Flow diagram for selection of nodes.

3.3.3 Extraction and Recognition of Morphs

After selecting the nodes next task is to extract the morphs in between nodes and their recognition. The arrays between successive nodes are extracted and they are put under a series of tests for recognition as one of the eight morphs.

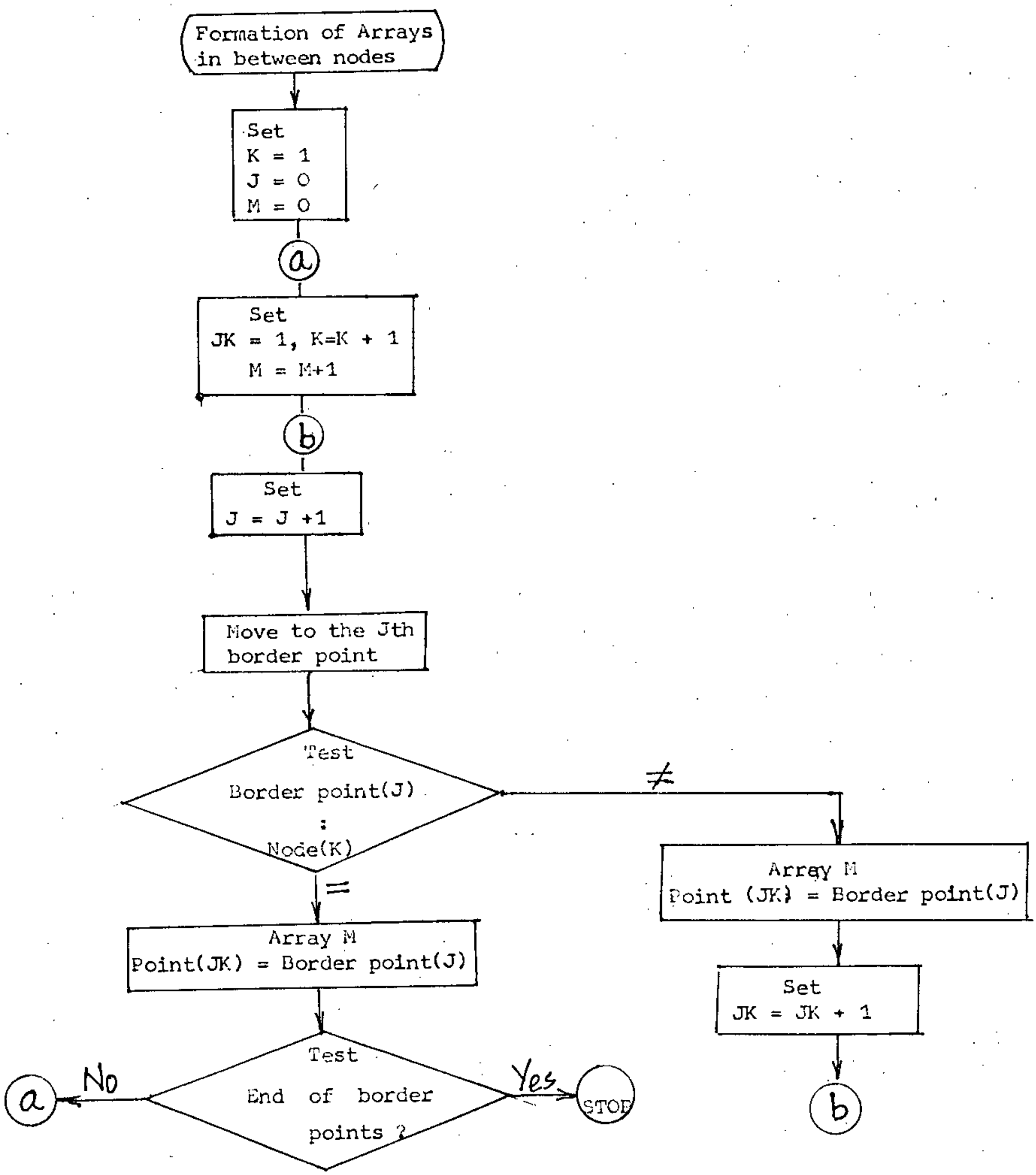


Fig 3.11: Flow diagram for formation of arrays in between nodes.

The arrays formed in between nodes are now tested to be recognised as one of the eight morphs.

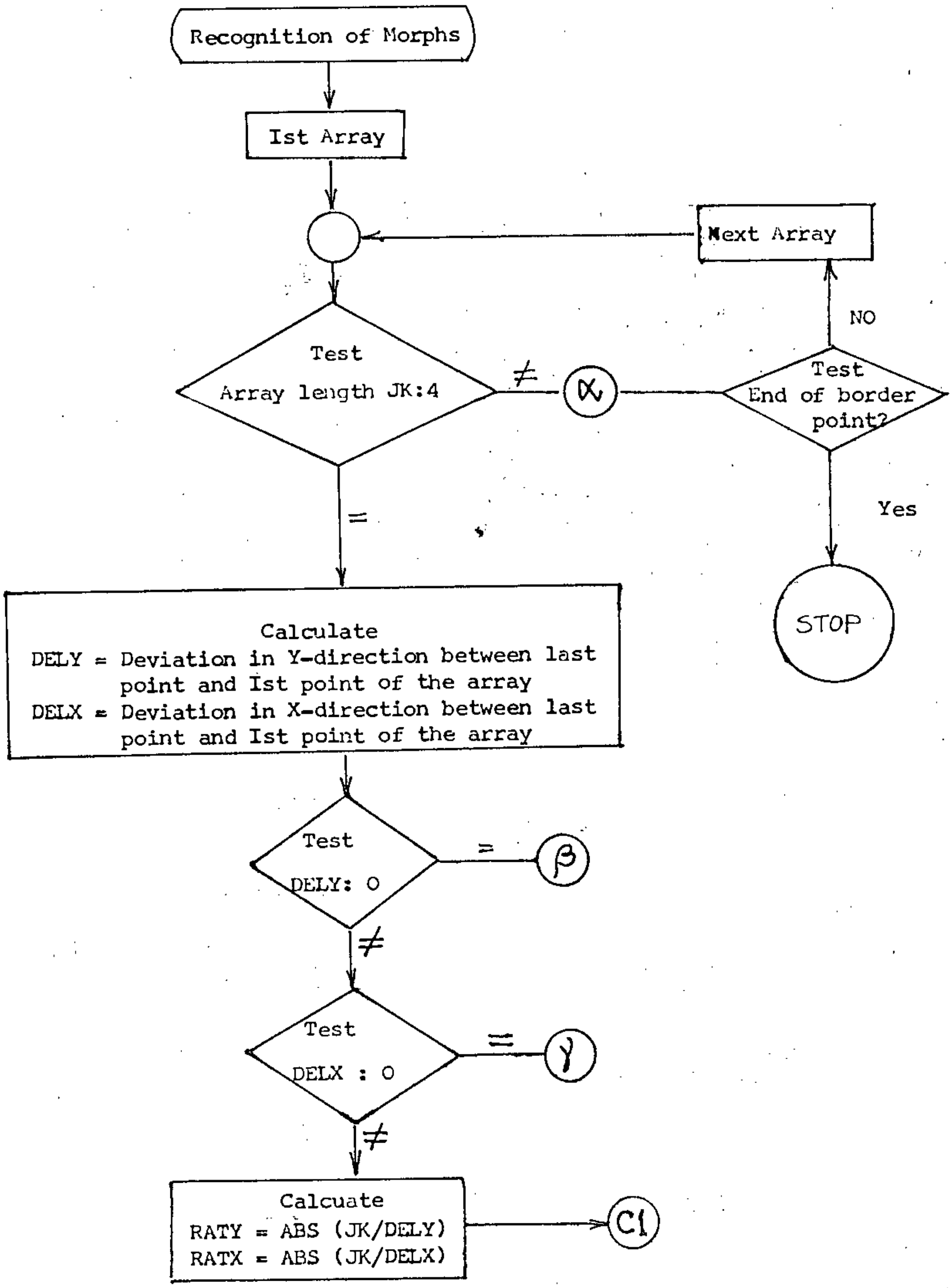


Fig 3.12 : Flow diagram for recognising morphs.

Fig 3.12 (Contd)

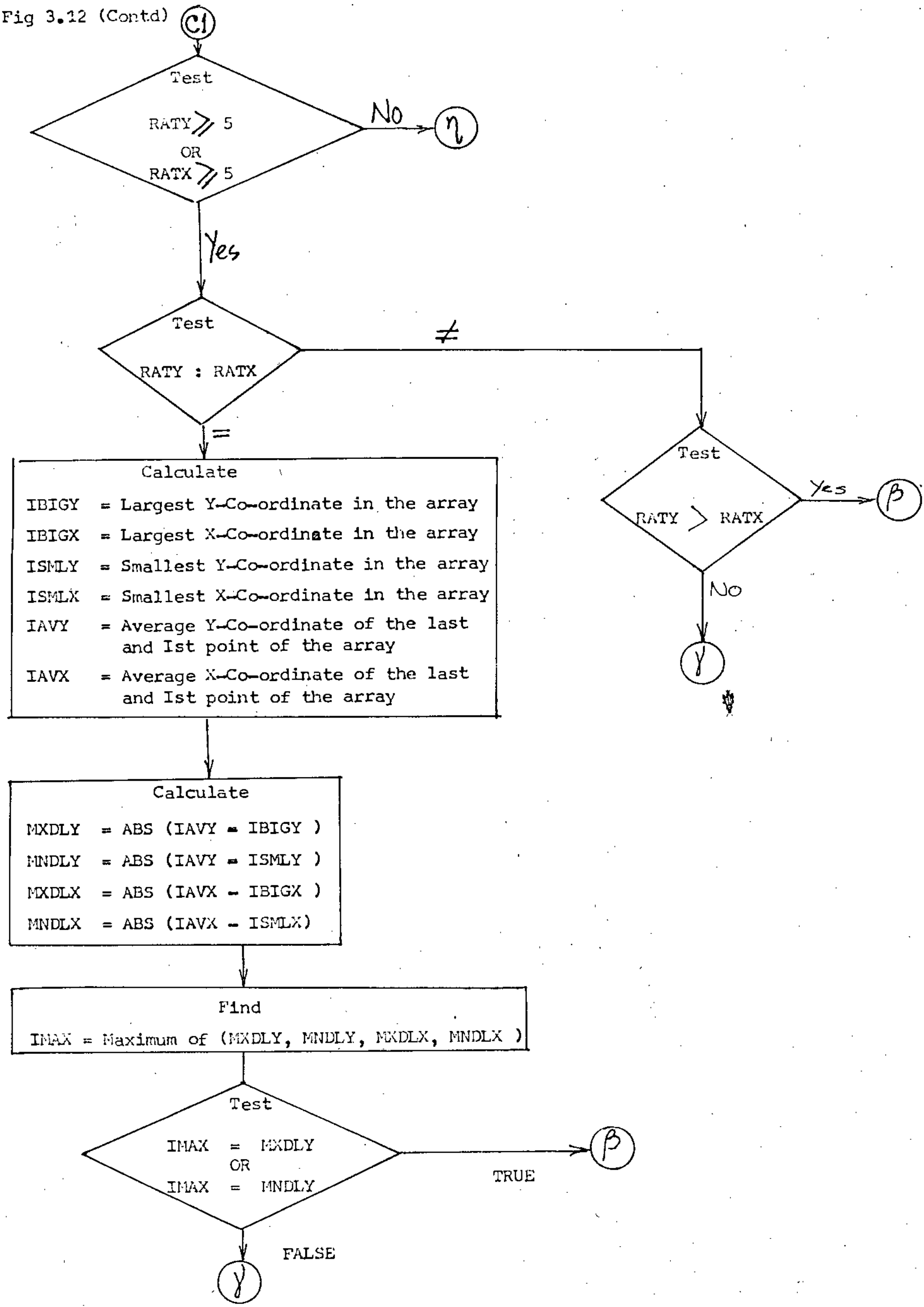


Fig 3.12 (Contd)

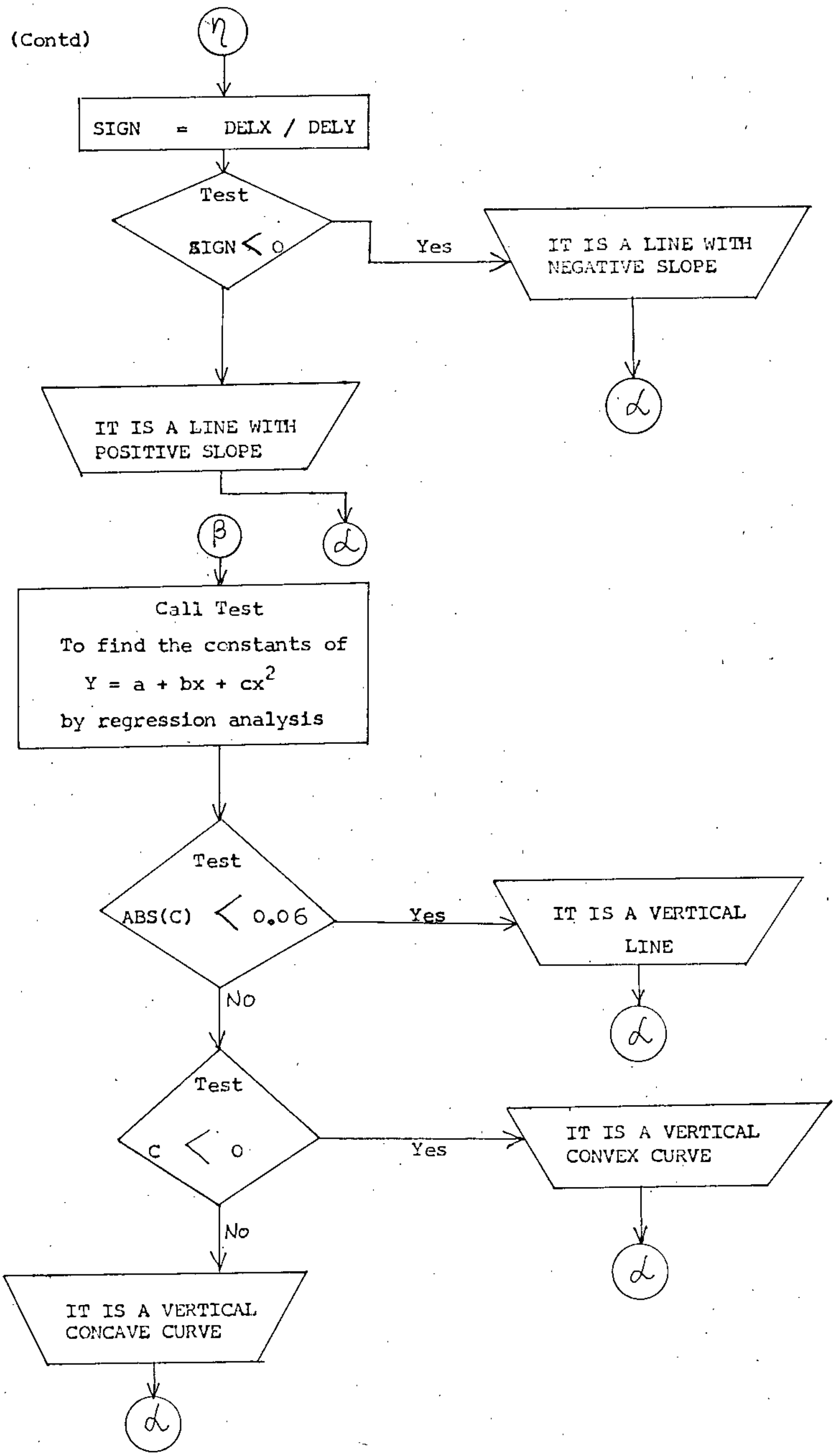
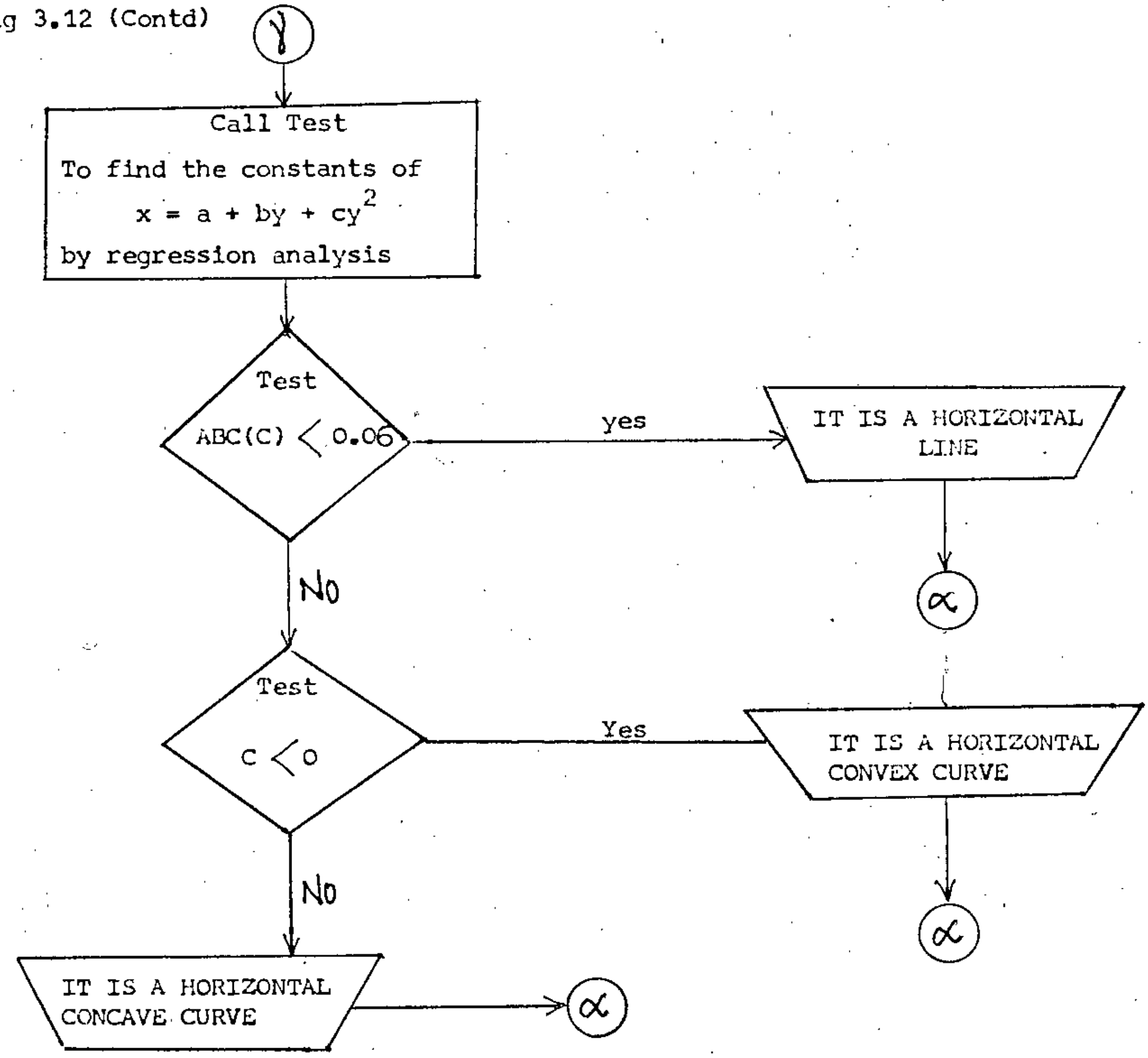


Fig 3.12 (Contd)



The character A of fig 3.1 is broken up into the following morphs

Ist array	:	Rejected
2nd array	:	Positive slope line
3rd array	:	Rejected
4th array	:	Horizontal concave curve
5th array	:	Rejected
6th array	:	Negative slope line

3.3.4 Representation of Characters by Numeric Codes

After resolving each character into morphs, a string of digits is generated showing the relationship among the morphs to built up the character.

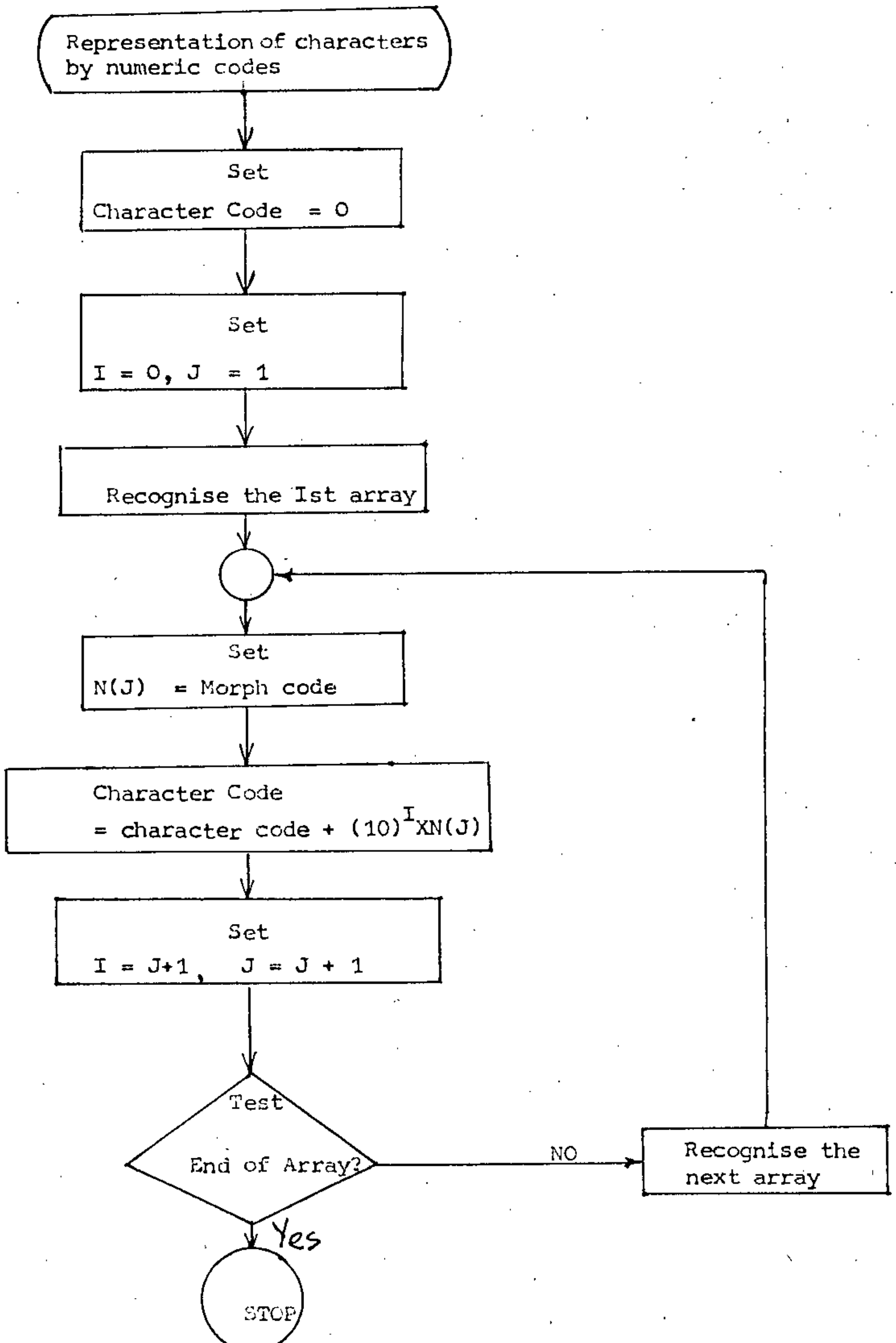


Fig 3.13 Flow diagram representing characters by numeric codes.

The character A of fig 3.1 gives rise to the code 473.

3.4 RECOGNITION OF CHARACTERS

The characters represented by numeric codes are now to be recognised. A dictionary of codes is searched to find a matching code corresponding to the code generated from the input character. The dictionary is developed giving codes corresponding to each acceptable appearance of a character (Appendix-B). The codes are acquired from sample characters taken from different sources.

Most of the characters can be represented uniquely as there are natural differences in structural compositions among the characters. Many of the characters have got more than one code depending on the writing style and quality of the digitized output. In a few cases the same code represents a pair of characters having structural similarities. Elaborate discussions about characters having more than one code and characters having same code are presented in chapter 4.

The dictionary is arranged in ascending order of the values of the codes representing the characters. For the ease of searching the total volume of the dictionary is divided into six subgroups:

Group 1 : Character code values less than 10^2.
Group 2 : Character code values in between 10^2 and 10^3.
Group 3 : Character code values in between 10^3 and 10^4.
Group 4 : Character code values in between 10^4 and 10^5.
Group 5 : Character code values in between 10^5 and 10^6.
Group 6 : Character code values in between 10^6 and 10^8.

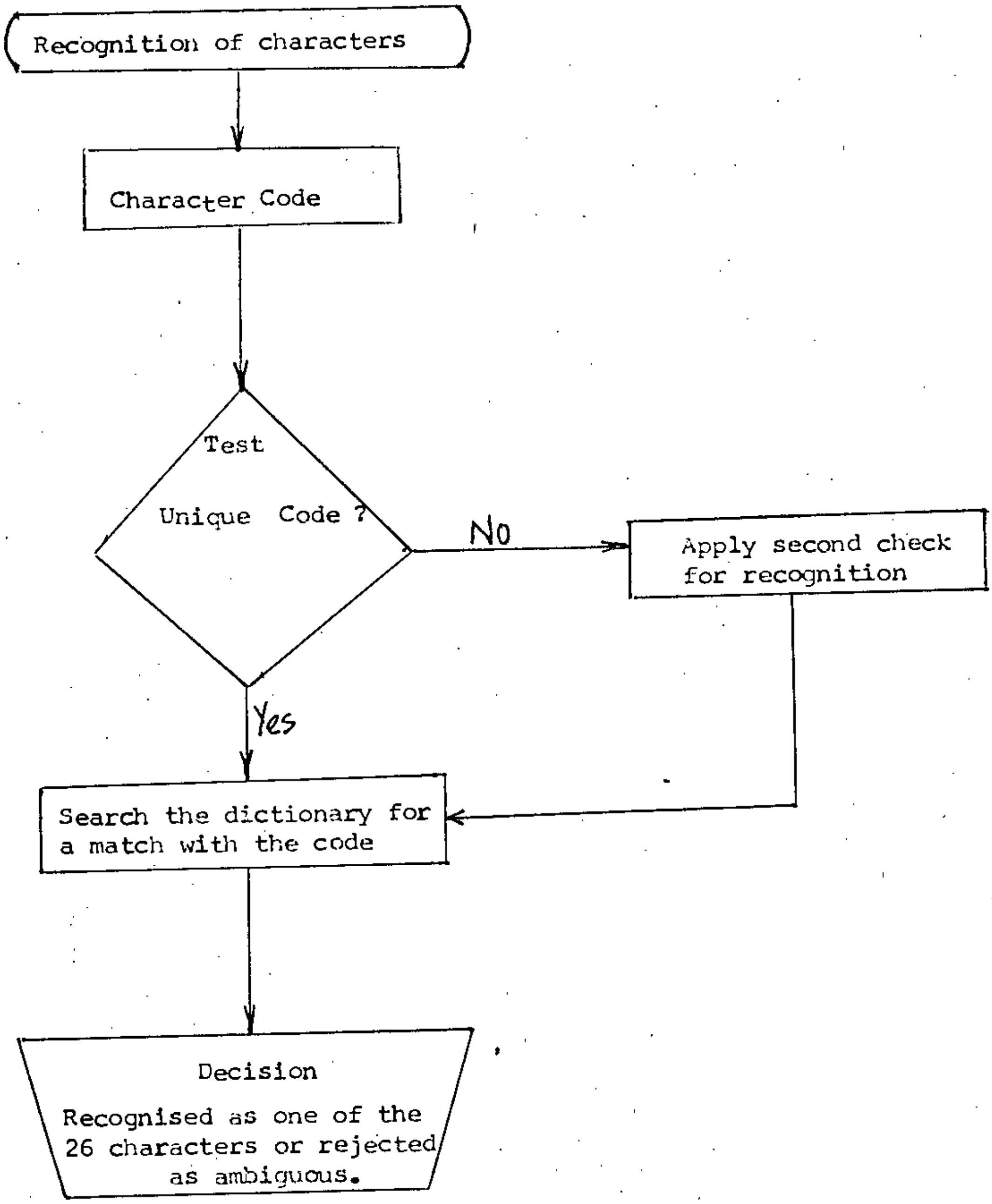


Fig 3.14 : Flow diagram for recognition of characters.

CHAPTER 4

DISCUSSIONS

4. DISCUSSIONS

4.1 GENERAL REMARKS

Recognition of characters is a very involved task as characters of allowable writing styles as well as variations in quality of the digitizer output have to taken into consideration. The program developed has been successfully tested with a number of characters from several different sources. The tests were designed to consider three aspects :

(i) Does the program develop consistent codes for characters of the same font i.e. will two instances of the same character from the same source yield the same code ?

(ii) Does the program work for characters from different sources ?

(iii) Do factors such as character size affect program performance ?

Sources in the above statements relates to different styles and sizes of writing, the result of the test was positive. Test (i) gives 100% accuracy and Test (ii) gives consistent accuracy within allowable writing styles. In test (iii) it has been found that some ambiguity may arise if a stroke length in a character becomes so small as to be comparable to the width of the stroke. In practice there is a remote possibility of such a case to happen.

4.2 GENERATION OF DATA

The present work concentrates on the analysis of digitized characters. For this work, digitized characters in matrix form have been generated manually, the same can be done with a suitable digitizer. Care was taken to make the digitized character the closest possible to the actual digitizer output. Characters of different writing styles were considered and at the same time the errors which were quite natural to the practical digitizer output were taken into consideration. For the present analysis it is assumed that there are no gaps or holes in any of the strokes. This is not always the case, in practical cases gaps and holes may appear in a stroke. This limitation may be overcome by a smoothing operation to fill in the gaps.

4.3 RECOGNITION OF MORPHS

In all eight different morphs were decided to represent the block capitals of English alphabet. In recognising morphs, a wide range of deviation is allowed to each of the eight morphs. To each horizontal or vertical line a deviation of 5:1 (length : deviation) is allowed, this stands for a deviation of about $\pm 12^{\circ}$ from a reference normal. All other straight lines out side these ranges are considered sloping lines, positive or negative depending on the sign of the slope. Figure 4.1 shows a diagrammatic representation of lines and their deviation limits.

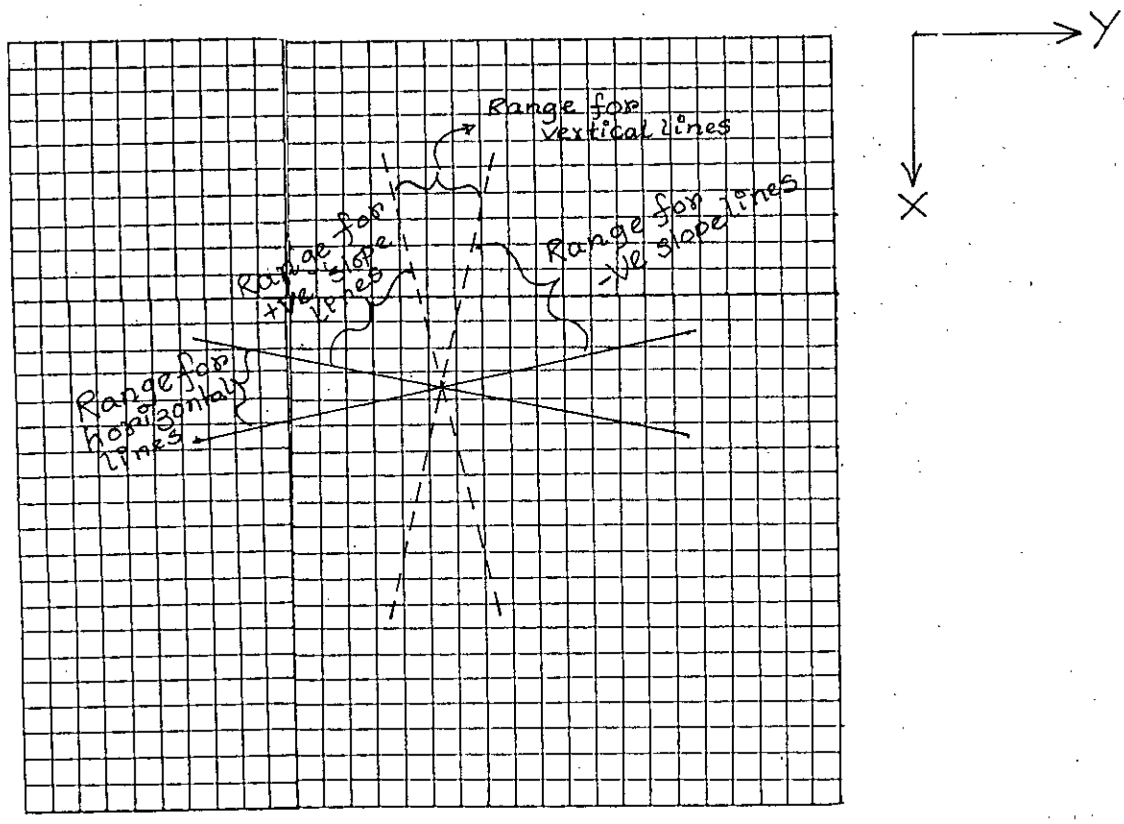


Fig 4.1 : Figure shows ranges for different straight lines.

For separating sloping lines, the ratio of the stroke length to deviation is first tested. If the absolute value of this ratio is less than five, the morph under consideration is a sloping line. Otherwise, the morph is either a curve or a straight line (horizontal or vertical).

For recognising a curve, regression analysis is performed to find the co-efficients of the eqn.:

$$y = a + bx + cx^2$$
or
$$x = a + by + cy^2$$

as the case may be. Now the limiting value of 'C' is set at

0.06. If absolute value of 'C' is greater than 0.06 the morph under consideration is a curve, otherwise, it is a straight line (horizontal or vertical). The limiting value 0.06 allows a wide range of flexibility.

4.4 RECOGNITION OF CHARACTERS

The process for generating numeric codes for characters has already been explained. The numerals in the code not only indicate the morphs which build the character but also shows the relationship among the morphs. As the scanner moves around each character in clock-wise direction, it is possible to make an idea about the structure of the character from its code. For example, character E gives a code 21551 which indicate the following structure(figure 4.2).

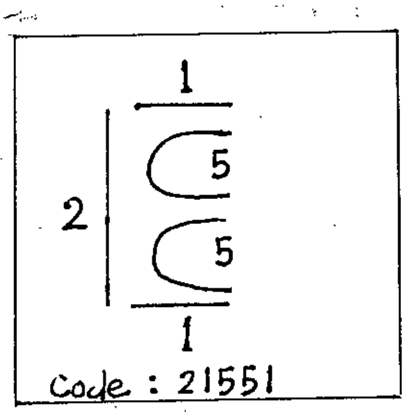


Fig 4.2 : Numeric code giving an idea about the structure.

SOME SPECIAL CASES

Most of the characters can be represented uniquely as there is natural structural differences among the characters. There are as many as four pairs of characters, each of which have got the same code due to structural similarities between them. Many of the characters have got more than one code due to variations in writing style and quality of the digitized data. In this article discussion is made on (1) some characters having more than one code and (2) some other characters having the same code.

(1) <u>Characters having more than one code</u>

Some of the characters having more than one code with their structures are shown in fig 4.3. One point to note here is that to each character deviation to such an extent is always allowed as is allowed to each morph.

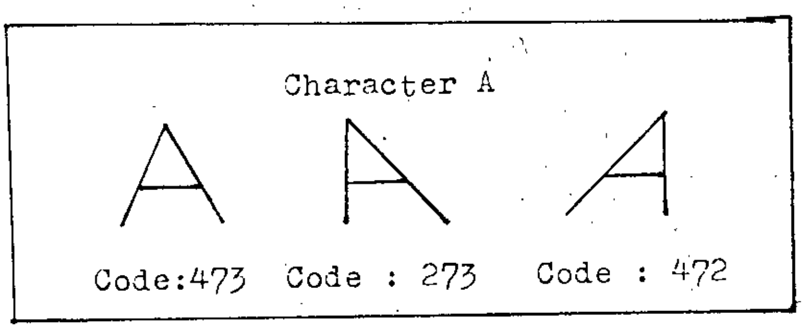


Fig 4.3(a)

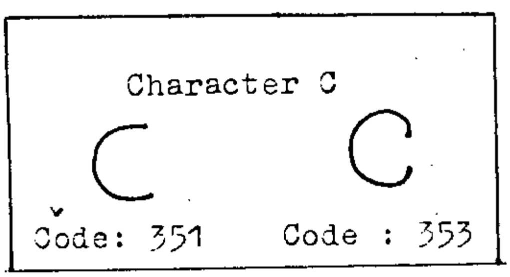


Fig 4.3(b)

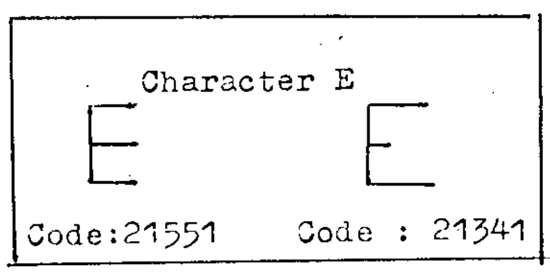


Fig 4.3.(c)

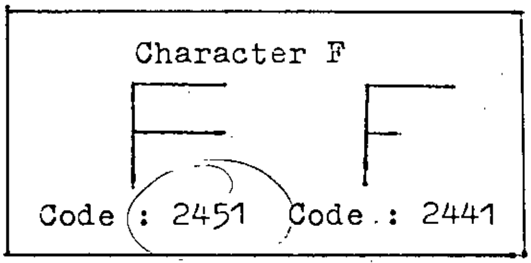


Fig 4.3(d)

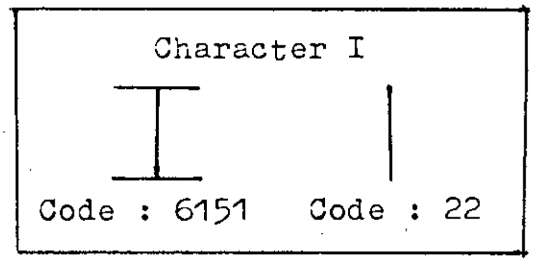


Fig 4.3(e)

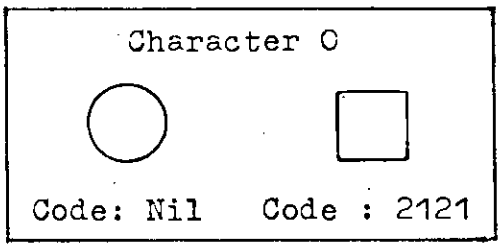


Fig 4.3(f)

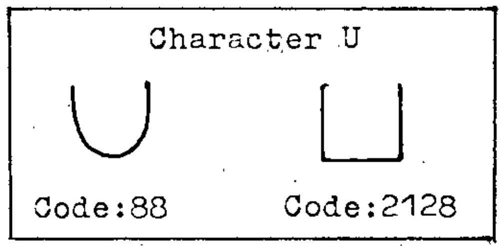


Fig 4.3(g)

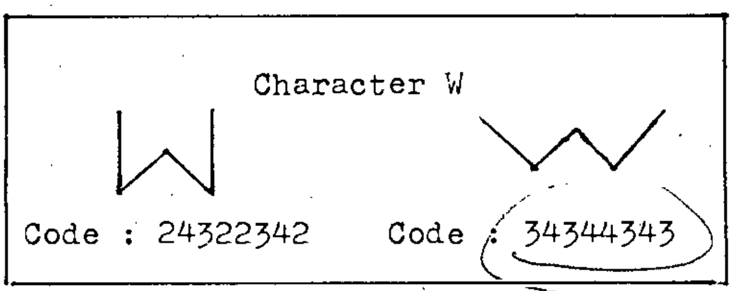


Fig 4.3(h).

Fig 4.3 : Structures of some characters having more than one code.

Fig 4.3(a) shows three different codes for character A due to difference in writing style. The first A resolves into a positive sloping line, a horizontal concave curve and a negative sloping line in sequence, giving rise to a code 473. For the second A, the first two morphs are identical to that of the first one but the third morph is a vertical line and consequently gives a code 273. The third A is similar to the first one except that its first morph is a vertical line thus giving a code 472.

Fig 4.3(b) shows two different structures of character C. The first one with code 351 is a flatter one making the first morph a horizontal line. The second one is curved with its head moved downward making the first morph a positive sloping line.

Fig 4.3(c) shows two structures of character E. When the middle arm of character E is shorter than the others; in some cases it may happen that the third and fourth morphs are sloping lines instead of curvatures. Thus the two codes of E may appear in practice.

Fig 4.3(d) shows two structures of character F. Explanation for two codes is similar to that of character E.

Fig 4.3(e) shows two I's, one is a simple vertical line having code 22 and in the other the ends of the vertical line are flanked by horizontal lines.

Fig 4.3(f) shows two structures of character O, with the first one without any numeric code. It is recognised directly without searching the dictionary. The second O is rather rectangular in shape and gives a code 2121.

Fig 4.3(g) shows two U's. The first one have circular bends having a code 88 and the second one has got prominent bends — giving code 2128.

Fig4.3(h) shows two different writing styles of W. In the first one the first and last strokes are vertical and in the second one they are slant.

(2) <u>Characters with same code:</u>

Characters D and P : Character D and P have got striking structural similarities. Both give rise to numeric code 26 which means each of D and P is made up of two morphs, a vertical convex curve and a vertical line. For distinguishing D and P a second check is employed. The second check is based on the structural difference that is in D : the ends of the vertical convex curve coincide with the ends of the vertical line; but in case of P only one end of the two morphs coincides, thus giving rise to three nodes in P and two nodes in D.

Characters V and Y : Both the characters V and Y give rise to code 3443 due to structural resemblance. For V-, second morph and third morph are always almost equal in length (fig. 4.4). If some one writes a 'Y'

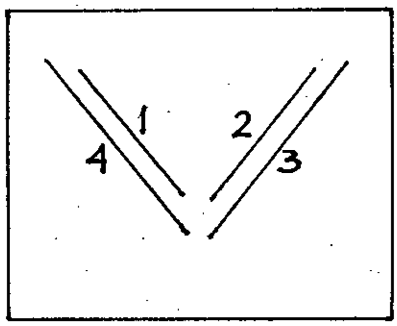


Fig 4.4 : Four morphs of character V.

in such a fashion so as to make second and third morphs almost equal, naturally the character will be closer to V and not Y. To decide whether the code 3443 is for a V or a Y, the ratio of length of the third morph to the second one is taken. If this ratio is greater than 1.3, the character will be recognised as Y, otherwise V.

Characters J and S : Character J gives code 641. If some one writes S in such a way so that node is not obtained at any one of its two bends, S too gives a code 641. The second check for distinguishing the two takes a count for number of nodes of the input character. If the number of nodes is 6, the code 641 is for character J, otherwise the character is S.

4.5 APPLICATIONS

The computer aided character recognition scheme can find its applications in areas, such as,

i) Robotics

ii) Linguistics

iii) To extend visual understanding to Computers

iv) To aid the education for the blind.

CHAPTER 5

CONCLUSION AND SUGGESTIONS FOR FUTURE WORK

5. CONCLUSION AND SUGGESTIONS FOR FUTURE WORK

5.1 CONCLUSION

For the present analysis the basic idea of syntactic method —— to represent each complex pattern in terms of simple subpatterns, is utilised. But the techniques developed for resolving each character into a number of morphs, their recognition and representation of characters by numeric codes and their recognition lead to a new approach and have never been suggested before.

Though the technique is quite efficient in recognising capital English letters, in some cases it fails to give a reasonable idea about the structural composition of the character. This limitation comes into effect in only a few cases when a curve on a slant line is recognised as a slant line, because the ratio of the morph length to deviation in those cases fall below 5.

The technique developed for selecting nodes will no doubt play a significant role in the field of pattern recognition; but still it needs further improvement so as to detect any sharp change of direction.

Regression analysis could have been used for recognising all the morphs. But it will demand more computer processing time. The present technique for recognising morphs is quite simple and efficient.

The character codes of the present analysis are quite informative. The code gives information about the morphs building up the character as well as their relation.

Like every method the present one naturally has got some limitations as mentioned; but still the performance of the present method in recognising capital English characters has found to be quite satisfactory. Also the techniques developed for the present analysis can be successfully implemented to other fields of pattern recognition.

5.2 SUGGESTIONS FOR FUTURE WORK

Further research works on present analysis should concentrate on

i) improvement of node selection technique, so as to detect any desired sharp change of direction

ii) investigation to find if statistical measurement for recognising morphs result in a more efficient technique

iii) modification of the present algorithm for implementing in character generation

iv) development of more efficient computer programs.

REFERENCES

REFERENCES

1. K.S.Fu : 'Introduction', Digital Pattern Recognition, ed. by K.S.Fu, PP1 - 14, springer-verlag New York, 1976.

2. D.Dutta Majumder : 'Pattern Recognition Methods and Applications', Recent Developments in Pattern Recognition and Digital Techniques, PP150-190, Indian Statistical Institute, 1977.

3. K.S.Fu : 'Introduction to Syntactic Pattern Recognition', Syntactic Pattern Recognition, Applications, ed. by K.S.Fu, pp 1-30, Springer-Verlag New York, 1977.

4. Laveen Kanal : 'Patterns in Pattern Recognition 1968-1974', Machine Recognition of Patterns, ed. by Ashok K. Agrawala, pp 1-26, IEEE Press, 1977.

5. Kendall Preston, Jr : 'A comparison of Analog and Digital Techniques for Pattern Recognition', Machine Recognition of Patterns, ed. by Ashok K. Agrawala, pp 92-107, IEEE Press,1977.

6. A.W.Holt : 'Comparative Religion in character Recognition Machines', Machine Recognition of Patterns, ed. by Ashok K. Agrawala, pp 433-441, IEEE Press, 1977.

7. K.S.Fu : 'Syntactic Methods in Pattern Recognition, pp 193-227, Academic Press, 1974.

8. A.C.Chottera and Shridhar : 'Cancellation of Spurious Features Due to Light Reflection', Canadian Journal of Electrical Engineering, 1974.

APPENDICES

APPENDIX-A

REGRESSION ANALYSIS FOR FINDING THE CO-EFFICIENTS OF A SECOND ORDER EQN.

Comparing the magnitudes of the co-efficients of eqn $y = a+bx+cx^2$, it is possible to make an idea about the curve that fits best to an array of N no. of points. With b = c = 0 and a = 0, the best fit would be a straight line parallel to the x - axis; with b dominating the best fit is a sloping line and with c dominating the best fit would be parabolic in nature. An idea about other natures of the best fit can also be obtained by comparing the magnitudes of a, b and c.

For solving for the co-efficients, regression analysis is performed.

From $y = a + bx + cx^2$

The following equations can be obtained for N no. of points,

$$\sum y = aN + b\sum x + c\sum x^2$$
$$\sum xy = a\sum x + b\sum x^2 + c\sum x^3$$
$$\sum x^2 y = a\sum x^2 + b\sum x^3 + c\sum x^4$$

Co-efficients a, b and c can now be obtained by solving the above three simultaneous eqns :

$$a = \frac{\begin{vmatrix} \sum y & \sum x & \sum x^2 \\ \sum xy & \sum x^2 & \sum x^3 \\ \sum x^2 y & \sum x^3 & \sum x^4 \end{vmatrix}}{D}$$

$$b = \frac{\begin{vmatrix} N & \sum y & \sum x^2 \\ \sum x & \sum xy & \sum x^3 \\ \sum x^2 & \sum x^2 y & \sum x^4 \end{vmatrix}}{D}$$

$$C = \frac{\begin{vmatrix} N & x & y \\ x & x^2 & xy \\ x^2 & x^3 & x^2 y \end{vmatrix}}{D}$$

$$\text{where } D = \begin{vmatrix} N & x & x^2 \\ x & x^2 & x^3 \\ x^2 & x^3 & x^4 \end{vmatrix}$$

Code	Character(s)
32153.	G
14114	Z
222243	M
223223	N
223542	K
223545	K
347348	X

Code	Character(s)
643543	X
2233442	K
2282243	M
22342243	M
24322342	W
34344343	W
34433443	X

* After second check, code of this number is changed by adding 1 to the present code.

APPENDIX-B

DICTIONARY SHOWING CHARACTERS CORRESPONDING TO DIFFERENT NUMERIC CODES.

Code	Character(s)
22	I
26	D,P*
33	Q
88	U
213	L
243	B
273	A
333	Q
341	T
351	C
353	C
472	A
473	A
533	Q
641	J,S*
643	S
843	V

Code	Character(s)
2121	O,I*
2128	U
2441	F
2451	F
2728	H
2733	R
3251	G
3253	G
3351	G
3443	Y,V*
6151	I
6758	X
21341	E
21551	E
22333	R
31611	S
32151	G

APPENDIX-C

MAIN PROGRAM FOR CHARACTER RECOGNITION

```
C***********************************************************************
C
C       THIS PROGRAM IS DEVELOPED FOR RECOGNISING CAPITAL ENGLISH
C       LETTERS BY SYNTACTIC METHOD OF PATTERN RECOGNITION. INPUT
C       PATTERN IS IN MATRIX FORM HAVING VALUES '0' FOR WHITE AND
C       '1' FOR BLACK POINTS.
C
C***********************************************************************
C
        DIMENSION IPIXEL(20,20),ISTRX(99),ISTRY(99),NODX(50),NODY(50),
       +IAX(50),IAY(50),X(50),Y(50),IXQ(50),IYQ(50),ISEG(25),ICHRC(75)
       +,ACHRC(75),LEN(30)
C
        DATA ICHRC / 22,26,27,33,88,213,243,273,333,341,351,353,472,473,
       +533,641,642,643,843,2121,2122,2128,2441,2451,2728,2733,3251,3351,
       +3253,3443,
       +3444,6151,6758,21341,21551,22333,31611,32151,32153,141141,222243,
       +223223,223542,223545,347348,643543,2233442,2282243,22342243,
       +24322342,34344343,34433443 /
C
        DATA ACHRC /'I','D','P','Q','U','L','B','A','Q','T','C','C','A',
       +'A','Q','J','S','S','V','D','I','U','F','F','H','R','G','G','G',
       +'Y','V','I','X','E','E','R','S','G','G','Z','M','N','K','K','X',
       +'X','K','M','M','W','W','X' /
        N =20
C*****************************************************************
C
C       SEARCH FOR A TRANSITION FROM WHITE TO BLACK
C
C*****************************************************************
333     READ (1,11,END=444) ((IPIXEL(I,J),J=1,N),I=1,N)
11      FORMAT (20I2)
        WRITE (3,717) ((IPIXEL(I,J),J=1,N),I=1,N)
717     FORMAT (10X,20I2)
        IX1 = 0
1       IY1 = 0
        IX1 = IX1+1
        IF (IX1.GE.N) GO TO 150
3       IY1 = IY1+1
        IF (IY1.GE.N) GO TO 1
        IX = IX1
        IY = IY1+1
        IF (IPIXEL(IX1,IY1).NE.IPIXEL(IX,IY)) GO TO 2
        GO TO 3
2       NBX1 = IX1
        NBY1 = IY1
        IBRX1 = IX
        IBRY1 = IY
C*****************************************************************
C
C       SUBROUTINE BORDER IS CALLED FOR DRAWING THE
C       BORDER LINE OF THE INPUT CHARACTER.
C
C*****************************************************************
        CALL BORDER(IX1,IY1,IX,IY,IPIXEL,NUM,ISTRX,ISTRY)
C       WRITE (3,11) ((IPIXEL(I,J),J=1,N),I=1,N)
C       WRITE (3,101) NUM
C101     FORMAT (1X,'NUM=',I2,/)
C       DO 103 I = 1,NUM
C       WRITE (3,102) ISTRX(I),ISTRY(I)
C102     FORMAT (1X,'ISTRX=',I2,',','ISTRY=',I2,/)
103     CONTINUE
```

```
C***********************************************************************
C
C      SUBROUTINE NODE IS CALLED FOR FINDING THE NODES.
C
C***********************************************************************
       CALL NODE (IBRX1,IBRY1,IPIXEL,NUM2,NODX,NODY,NBX1,NBY1)
       IF (NUM2.EQ.1) GO TO 88
C      DO 501 I = 1,NUM2
C      WRITE (3,502) NODX(I),NODY(I)
C502    FORMAT (1X,'NODX=',I2,',','NODY=',I2,/)
C501    CONTINUE
C***********************************************************************
C
C      FORMATION OF ARRAY WITH THE POINTS INBETWEEN TWO NODES
C
C***********************************************************************
100    NUM3 = NUM2 + 1
       NODX(NUM3) = NODX(1)
       NODY(NUM3) = NODY(1)
       JJ = NUM + 1
       ISTRX(JJ) = ISTRX(1)
       ISTRY(JJ) = ISTRY(1)
       K = 1
       J = 0
       NUMSEG = 0
       IDVAL = 0
       MULT = 1
       ITER = 0
5      JK = 1
       NUMSEG = NUMSEG + 1
       K = K+1
30     J = J+1
       IF (ISTRX(J).EQ.NODX(K).AND.ISTRY(J).EQ.NODY(K)) GO TO 10
       IAX(JK) = ISTRX(J)
       IAY(JK) = ISTRY(J)
       JK = JK+1
       GO TO 30
10     IAX(JK) = ISTRX(J)
       IAY(JK) = ISTRY(J)
C      DO 4 I = 1,JK
C      WRITE (3,33) I,IAX(I),I,IAY(I)
C33     FORMAT (1X,'IAX(',I3,')=',I3,'IAY(',I3,')=',I3,/)
C4      CONTINUE
C***********************************************************************
C
C      CHECKS TO DECIDE WHETHER THE MORPH IS A SLOPING LINE OR EITHER
C      OF A CURVE OR A STRAIGHT LINE (HORIZONTAL OR VERTICAL)
C
C***********************************************************************
       IF (JK.GT.4) GO TO 60
       GO TO 79
60     ITER = ITER + 1
       LEN(ITER) = JK
       IDELY = (IAY(JK)-IAY(1))
       IF (IDELY.EQ.0) GO TO 55
       IDELX = (IAX(JK)-IAX(1))
       IF (IDELX.EQ.0) GO TO 52
       DELX = FLOAT (IDELX)
       DELY = FLOAT (IDELY)
       ARRAYL = FLOAT (JK)
       RATY =ABS (ARRAYL/DELY)
       RATX =ABS (ARRAYL/DELX)
       IF (RATY.GE.5.OR.RATX.GE.5) GO TO 31
       GO TO 22
31     IF (RATY.EQ.RATX) GO TO 32
       IF (RATY.GT.RATX) GO TO 55
       GO TO 52
```

```
C*******************************************************************
C
C      IF RATY = RATX, IT IS TO BE FOUND WHETHER THE CURVE
C      IS ON THE VERTICAL LINE OR ON THE HORIZONTAL LINE
C
C*******************************************************************
32     IBIGY = IAY(1)
       IBIGX = IAX(1)
       ISMALY = IAY(1)
       ISMALX = IAX(1)
       I = 0
12     I = I+1
       IF (I.GE.JK) GO TO 13
       IF (IAY(I).GT.IBIGY) GO TO 14
       GO TO 15
14     IBIGY = IAY(I)
15     IF (IAX(I).GT.IBIGX) GO TO 16
       GO TO 12
16     IBIGX = IAX(I)
       GO TO 12
13     I = 0
17     I = I+1
       IF (I.GE.JK) GO TO 18
       IF (IAY(I).LT.ISMALY) GO TO 19
       GO TO 20
19     ISMALY = IAY(I)
20     IF (IAX(I).LT.ISMALX) GO TO 21
       GO TO 17

21     ISMALX = IAX(I)
       GO TO 17
18     IAVY = (IAY(1) + IAY(JK))/2
       IAVX = (IAX(1) + IAX(JK))/2
       MXDLY = IAVY - IBIGY
       MNDLY = IAVY - ISMALY
       MXDLX = IAVX - IBIGX
       MNDLX = IAVX - ISMALX
       IABMXY = IABS (MXDLY)
       IABMNY = IABS (MNDLY)
       IABMXX = IABS (MXDLX)
       IABMNX = IABS (MNDLX)
       IMAX = MAX0 (IABMXY,IABMNY,IABMXX,IABMNX)
       IF (IMAX.EQ.IABMXY.OR.IMAX.EQ.IABMNY) GO TO 55
       GO TO 52
C**************************************************************
C
C      CHECKS FOR FINDING THE NATURE OF THE SLOPE
C
C**************************************************************
22     SIGN = DELX/DELY
       IF (SIGN.LT.O) GO TO 50
C      WRITE (3,61)
C61     FORMAT (//,10X,'IT IS A LINE WITH POSITIVE SLOPE',//)
       ISEG (NUMSEG) = 3
       GO TO 77
C50     WRITE (3,51)
C51     FORMAT (//,10X,'IT IS A LINE WITH NEGATIVE SLOPE',//)
50     ISEG (NUMSEG) = 4
       GO TO 77
```

```
C*****************************************************************************
C
C       SUBROUTINE TEST IS CALLED FOR SOLVING EQN. X = A + BX + CX ** 2
C       BY REGRESSION ANALYSISTO FIND 'C'. FOR ABS. VALUE OF 'C' GREATER
C       THAN 0.06 THE MORPH WILL BE TREATED AS A CURVE, OTHERWISE A
C       HORIZONTAL LINE
C
C*****************************************************************************
52      CALL TEST (IAY,IAX,A,B,C,JK)
C       WRITE (3,707) C
C707     FORMAT (//,' C = ',F10.6,/)
        AC = ABS(C)
        IF (AC.LE.0.06) GO TO 53
        IF (C.LT.0) GO TO 62
C        WRITE (3,63)
C63      FORMAT (//,10X,'IT IS A HORIZONTAL CONCAVE CURVE',//)
        ISEG (NUMSEG) = 7
        GO TO 77
C62      WRITE (3,64)
C64      FORMAT (//,10X,'IT IS A HORIZONTAL CONVEX CURVE',//)
62      ISEG (NUMSEG) = 8
        GO TO 77
C53      WRITE (3,54)
C54      FORMAT (//,10X,'IT IS A HORIZONTAL LINE',//)
53      ISEG (NUMSEG) = 1
        GO TO 77
C*************************************************************************
C
C       SUBROUTINE TEST IS CALLED TO DECIDE WHETHER THE MORPH IS A
C       CURVE OR A VERTICAL LINE DEPENDING ON THE VALUE OF 'C'
C
C*************************************************************************
55      CALL TEST (IAX,IAY,A,B,C,JK)
C       WRITE (3,708) C
C708     FORMAT (//,' C = ',F10.6,/)
        AC = ABS(C)
        IF (AC.LE.0.06) GO TO 56
        IF (C.LT.0) GO TO 57
C78      WRITE (3,65)
C65      FORMAT (//,10X,'IT IS A VERTICAL CONCAVE CURVE',//)
78      ISEG (NUMSEG) = 5
        GO TO 77
C57      WRITE (3,66)
C66      FORMAT (//,10X,'IT IS A VERTICAL CONVEX CURVE',//)
57      ISEG (NUMSEG) = 6
        GO TO 77
C56      WRITE (3,67)
C67      FORMAT (//,10X,'IT IS A VERTICAL LINE',//)
56      ISEG (NUMSEG) = 2
C********************************************
C
C       FORMATION OF CHARACTER CODE
C
C********************************************
77      IDVAL = IDVAL + ISEG (NUMSEG) * MULT
        MULT = MULT * 10
79      IF (J .LT. NUM) GO TO 5
        GO TO 630
C       WRITE (3,151) IDVAL
C151     FORMAT (//,10X,'IDVAL = ',I20,//)
88      WRITE (3,89)
89      FORMAT (//,10X,' THE RECOGNISED CHARACTER IS --  0  --',//)
        GO TO 150
```

```
C*******************************************************
C
C      CHECKS FOR DISTINGUISHING CHARACTERS
C      HAVING SAME CODE
C
C*******************************************************
630    IF ( IDVAL .EQ. 26) GO TO 601
       IF ( IDVAL .EQ. 641) GO TO 602
       IF ( IDVAL .EQ. 2121) GO TO 603
       IF ( IDVAL .EQ. 3443) GO TO 605
       GO TO 250
601    IF (NUM2 .EQ. 2) GO TO 250
       IDVAL = 27
       GO TO 250
602    IF (NUM2 .EQ. 6) GO TO 250
       IDVAL = 642
       GO TO 250
603    IF (NUM2 .EQ. 4) GO TO 250
       IDVAL = 2122
       GO TO 250
605    TESTA = FLOAT(LEN(3))/FLOAT(LEN(2))
       IF (TESTA .GT. 1.3) GO TO 250
       IDVAL = 3444
C*******************************************************
C
C      SUBGROUPING OF THE DICTIONARY AND
C      TECHNIQUE FOR ITS SEARCHING
C
C*******************************************************
250    IF (IDVAL.LT.100) GO TO 91
       IF (IDVAL.LT.1000) GO TO 92
       IF (IDVAL.LT.10000) GO TO 93
       IF (IDVAL.LT.100000) GO TO 94
       IF (IDVAL.LT.1000000) GO TO 95
       IST = 47
       IEND = 52
       GO TO 313
91     IST = 1
       IEND = 5
       GO TO 313
92     IST = 6
       IEND = 19
       GO TO 313
93     IST = 20
       IEND = 33
       GO TO 313
94     IST = 34
       IEND = 39
       GO TO 313
95     IST = 40
       IEND = 46
313    DO 332 I = IST,IEND
       IF (ICHRC(I) .NE. IDVAL) GO TO 332
       IVAL = I
       GO TO 41
332    CONTINUE
C*****************************************************************
C
C      RECOGNITION OF THE INPUT CHARACTER
C
C*****************************************************************
       WRITE (3,400)
400    FORMAT (/,10X,' THE CHARACTER CAN NOT BE RECOGNISED',/)
       WRITE (3,421) IDVAL
421    FORMAT (//,10X,'IDVAL = ',I9,//)
       GO TO 150
41     WRITE (3,401) ACHRC(IVAL)
401    FORMAT (////,10X,' THE RECOGNISED CHARACTER IS --',1X,A2,'--',///)
C      WRITE (3,420)IDVAL
C420    FORMAT (//,10X,' IDVAL = ',I9,//)
150    GO TO 333
444    STOP
       END
```

APPENDIX - D

SUBROUTINE BORDER

```
C*****************************************************************************
C
C       SUBROUTINE BORDER PERFORMS THE TASK OF TRACING THE BORDER
C       LINE OF THE INPUT CHARACTER BY CHANGING THE LEVEL OF THE
C       BORDER POINTS TO 2. HERE,
C               (IX,IY) = CO-ORDINATES OF THE 1ST BLACK POINT
C               (IX1,IY1) = CO-ORDINATES OF THE NEIGHBOUR POINT
C
C*****************************************************************************
        SUBROUTINE BORDER (IX1,IY1,IX,IY,IPIXEL,NUM,ISTRX,ISTRY)
        DIMENSION IPIXEL(20,20),ISTRX(99),ISTRY(99)
        NUM = 1
        ISTRX (NUM) = IX
        ISTRY (NUM) = IY
        IPIXEL (IX,IY) = 3
10      IREF = IPIXEL (IX1,IY1)
        IP1 = IX1
        IP2 = IY1
        K1 = IY - IY1
        K2 = IX - IX1
        IP1 = IP1 - K1
        IP2 = IP2 + K2
        IF (IPIXEL(IP1,IP2).NE.IREF) GO TO 8
        IX1 = IP1
        IY1 = IP2
        IP1 = IP1 + K2
        IP2 = IP2 + K1
        IF (IPIXEL(IP1,IP2).NE.IREF) GO TO 8
        IX1 = IP1
        IY1 = IP2
        IP1 = IP1 + K2
        IP2 = IP2 + K1
        IF (IPIXEL(IP1,IP2).NE.IREF) GO TO 8
        IX1 = IP1
        IY1 = IP2
        IP1 = IP1 + K1
        IP2 = IP2 - K2
        IF (IPIXEL(IP1,IP2).NE.IREF) GO TO 8
        IX1 = IP1
        IY1 = IP2
        IP1 = IP1 + K1
        IP2 = IP2 - K2
        IF (IPIXEL(IP1,IP2).NE.IREF) GO TO 8
        IX1 = IP1
        IY1 = IP2
        IP1 = IP1 - K2
        IP2 = IP2 - K1
        IF (IPIXEL(IP1,IP2).NE.IREF) GO TO 8
        IX1 = IP1
        IY1 = IP2
        IP1 = IP1 - K2
        IP2 = IP2 - K1
        IF (IPIXEL(IP1,IP2).NE.IREF) GO TO 8
        ISTRX (NUM) = IP1
        ISTRY (NUM) = IP2
        GO TO 9
8       IF (IPIXEL(IP1,IP2).NE.3) GO TO 11
        IPIXEL (IP1,IP2) = 2
        GO TO 9
11      IF(IPIXEL(IP1,IP2).EQ.2) GO TO 9
        NUM = NUM + 1
        IPIXEL (IP1,IP2) = 2
        ISTRX (NUM) = IP1
        ISTRY (NUM) = IP2
        IX = IP1
        IY = IP2
        GO TO 10
9       RETURN
        END
```

APPENDIX-E

SUBROUTINE NODE

```
C*****************************************************************************
C
C       AFTER TRACING THE BORDER LINE OF THE INPUT CHARACTER SUBROUTINE
C       NODE IS CALLED FOR LOCATING THE 'NODES' OF THE INPUT PATTERN.
C       NODES ARE FORMED AT POINTS WHERE THERE IS A SHARP CHANGE OF
C       DIRECTION OF 90 DEGREE OR LESS AND 270 DEGREE OR HIGHER
C
C*****************************************************************************
        SUBROUTINE NODE (IBRX1,IBRY1,IPIXEL,NUM2,NODX,NODY,NBX1,NBY1)
        DIMENSION IPIXEL(20,20),NODX(50),NODY(50)
        NUM2 = 1
        NODX (NUM2) = IBRX1
        NODY (NUM2) = IBRY1
        IPIXEL (IBRX1,IBRY1) = 3
11      IP1 = NBX1
        IP2 = NBY1
        K1 = IBRY1 - NBY1
        K2 = IBRX1 - NBX1
        IP1 = IP1 -K1
        IP2 = IP2 +K2
        IF (IPIXEL(IP1,IP2).EQ.3) GO TO 13
        IF (IPIXEL(IP1,IP2).EQ.2) GO TO 8
        NBX1 = IP1
        NBY1 = IP2
        IP1 = IP1 +K2
        IP2 = IP2 + K1
        IF (IPIXEL(IP1,IP2).EQ.3) GO TO 13
        IF (IPIXEL(IP1,IP2).EQ.2) GO TO 8
        NBX1 = IP1
        NBY1 = IP2
        IP1 = IP1 + K2
        IP2 = IP2 +K1
        IF (IPIXEL(IP1,IP2).EQ.3) GO TO 13
        IF (IPIXEL(IP1,IP2).EQ.2) GO TO 8
        NBX1 = IP1
        NBY1 = IP2
        IP1 = IP1 + K1
        IP2 = IP2 -K2
        IF (IPIXEL(IP1,IP2).EQ.3) GO TO 13
        IF (IPIXEL(IP1,IP2).EQ.2) GO TO 8
        NBX1 = IP1
        NBY1 = IP2
        IP1 = IP1 + K1
        IP2 = IP2 - K2
        IF (IPIXEL(IP1,IP2).EQ.3) GO TO 13
        IF (IPIXEL(IP1,IP2).EQ.2) GO TO 8
        NBX1 = IP1
        NBY1 = IP2
        IP1 = IP1 - K2
        IP2 = IP2 - K1
        IF (IPIXEL(IP1,IP2).EQ.3) GO TO 13
        IF (IPIXEL(IP1,IP2).EQ.2) GO TO 8
        NBX1 = IP1
        NBY1 = IP2
        IP1 = IP1 - K2
        IP2 = IP2 - K1
        IF (IPIXEL(IP1,IP2).EQ.3) GO TO 13
        IF (IPIXEL(IP1,IP2).EQ.2) GO TO 8
        GO TO 13
```

```
8       NX1  = NBX1
        NY1  = NBY1
        IBX  = IP1
        IBY  = IP2
        IK1  = IBX - NX1
        IK2  = IBY - NY1
        NX = IBX + IK1
        NY = IBY + IK2
        IF (IPIXEL(NX,NY).EQ.2) GO TO 9
        NX2 = NX1 + IK2
        NY2 = NY1 + IK1
        NX3 = NX1 - IK2
        NY3 = NY1 - IK1
        IF (IPIXEL(NX2,NY2).EQ.2.AND.IPIXEL(NX3,NY3).EQ.2) GO TO 10
        GO TO 12
9       NX = IBX + IK2
        NY = IBY +IK1
        IF (IPIXEL(NX,NY).EQ.2) GO TO 10
        NX = IBX - IK2
        NY = IBY - IK1
        IF (IPIXEL(NX,NY).EQ.2) GO TO 10
        GO TO 12
10      NUM2 = NUM2 + 1
        NODX(NUM2) = IP1
        NODY(NUM2) = IP2
12      IBRX1 = IP1
        IBRY1 = IP2
        GO TO 11
13      RETURN
        END
```

APPENDIX-F

SUBROUTINE TEST.

```
C*****************************************************************************
C
C       SUBROUTINE TEST PERFORMS REGRESSION ANLYSIS FOR SOLVING THE
C       EQN. Y = A + BX + CX ** 2(OR EQN. X = A + BY +CY ** 2) TO
C       FIND THE CONSTANTS. FOR THE PRESENT ANALYSIS THE VALUE OF 'C'
C       IS USED TO DECIDE WHETHER THE MORPH UNDER CONSIDERATION IS
C       A CURVE OR A STRAIGHT LINE
C
C*****************************************************************************
      SUBROUTINE TEST (IXQ,IYQ,A,B,C,N)
      DIMENSION IXQ(50),IYQ(50),X(50),Y(50)
      DO 15 I = 1,N
      KX = IXQ(I)
      KY = IYQ(I)
      X(I) = FLOAT(KX)
      Y(I) = FLOAT(KY)
15    CONTINUE
      SMY = 0.
      SMX = 0.
      SMXSQ = 0.
      SMXCB = 0.
      SMX4 = 0.
      SMXY = 0.
      SMX2Y = 0.
      DO 3 I = 1,N
      SMY = SMY + Y(I)
      SMX = SMX + X(I)
      SMXSQ = SMXSQ + X(I)**2
      SMXCB = SMXCB + X(I)**3
      SMX4 = SMX4 + X(I)**4
      SMXY = SMXY + X(I)*Y(I)
      SMX2Y = SMX2Y + X(I)**2*Y(I)
3     CONTINUE
      D = (N*SMXSQ*SMX4-N*SMXCB**2-SMX**2*SMX4+2*SMX*SMXSQ*SMXCB-SMXSQ**
     +3)
      A = (SMY*SMXSQ*SMX4-SMY*SMXCB**2-SMX*SMXY*SMX4+SMX*SMX2Y*SMXCB+SMX
     +SQ*SMXY*SMXCB-SMXSQ**2*SMX2Y)/D
      B = (N*SMXY*SMX4-N*SMX2Y*SMXCB-SMY*SMX*SMX4+SMY*SMXSQ*SMXCB+SMX*SM
     +XSQ*SMX2Y-SMXSQ**2*SMXY)/D
      C = (N*SMXSQ*SMX2Y-N*SMXY*SMXCB-SMX**2*SMX2Y+SMX*SMXY*SMXSQ+SMY*SM
     +X*SMXCB-SMY*SMXSQ**2)/D
      RETURN
      END
```

APPENDIX-G.

INPUT CHARACTERS IN DIGITIZED FORM AND THE RESULTS

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

THE RECOGNISED CHARACTER IS -- A --

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

THE RECOGNISED CHARACTER IS -- B --

G2

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

THE RECOGNISED CHARACTER IS -- C --

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

THE RECOGNISED CHARACTER IS -- D --

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

THE RECOGNISED CHARACTER IS -- E --

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

THE RECOGNISED CHARACTER IS -- F --

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

THE RECOGNISED CHARACTER IS -- G --

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

THE RECOGNISED CHARACTER IS -- H --

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

THE RECOGNISED CHARACTER IS -- I --

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

THE RECOGNISED CHARACTER IS -- J --

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

THE RECOGNISED CHARACTER IS -- K --

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

THE RECOGNISED CHARACTER IS -- L --

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

THE RECOGNISED CHARACTER IS -- M --

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

THE RECOGNISED CHARACTER IS -- N --

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

THE RECOGNISED CHARACTER IS -- 0 --

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

THE RECOGNISED CHARACTER IS -- P --

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

THE RECOGNISED CHARACTER IS -- Q --

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

THE RECOGNISED CHARACTER IS -- R --

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

THE RECOGNISED CHARACTER IS -- S --

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

THE RECOGNISED CHARACTER IS -- T --

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

THE RECOGNISED CHARACTER IS -- U --

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

THE RECOGNISED CHARACTER IS -- V --

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

THE RECOGNISED CHARACTER IS -- W --

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

THE RECOGNISED CHARACTER IS -- X --

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

THE RECOGNISED CHARACTER IS -- Y --

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

THE RECOGNISED CHARACTER IS -- Z --