1976-03

Language differentiation based on sound patterns of the spoken word

Cook, Roger Darrell

LANGUAGE DIFFERENTIATION BASED ON SOUND PATTERNS OF THE SPOKEN WORD

Roger Darrell Cook

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

LANGUAGE DIFFERENTIATION BASED ON
SOUND PATTERNS OF THE SPOKEN WORD

by

Roger Darrell Cook

March 1976

Thesis Advisor: S. Jauregui

Approved for public release; distribution unlimited.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Language Differentiation Based On Sound Patterns of the Spoken Word		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; March 1976
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Roger Darrell Cook		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE March 1976
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

phonemics, language, phonetics, voice, speech

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A categorical analysis was made of five languages. The sounds of speech were simulated using written text converted via International Phonetic Alphabet (IPA). The sounds of speech were identified as members of fracative, nasal, stop, or vowel categories. A statistical analysis was performed on categorical content of one (at various positions in the word), two, and three sound combinations.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
(Page 1) S/N 0102-014-6601 |

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Several attempts to achieve a differentiation scheme were made before any success was realized. Two methods for developing conditional expectation are compared; Bayes' Conditional Probability Rule, and Cook's Prognostic Progression.

Statistical analysis and "loop" tests indicated that languages do have unique patterns and can be differentiated on the statistics contained in the first three sounds. 100% correct decisions were achieved for as few as five words in the loop test. Limited base data negated result significance beyond three successive sounds.

Language Differentiation Based On Sound Patterns Of The Spoken Word

by

Roger Darrell Cook
Lieutenant, United States Navy
B.S.E.E., University of New Mexico, 1971

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN MANAGEMENT

from the
NAVAL POSTGRADUATE SCHOOL
March 1976

ABSTRACT

A categorical analysis was made of five languages. The sounds of speech were simulated using written text converted via International Phonetic Alphabet (IPA). The sounds of speech were identified as members of fricative, nasal, stop, or vowel categories. A statistical analysis was performed on categorical content of one (at various positions in the word), two, and three sound combinations.

Several attempts to achieve a differentiation scheme were made before any success was realized. Two methods for developing conditional expectation are compared; Bayes' Conditional Probability Rule, and Cook's Prognostic Progression.

Statistical analysis and "loop" tests indicated that languages do have unique patterns and can be differentiated on the statistics contained in the first three sounds. 100% correct decisions were achieved for as few as five words in the loop test. Limited base data negated result significance beyond three successive sounds.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Several people have made numerous contributions toward the fulfillment of this research study. The most significant contribution was made by Lieutenant Michael C. Thomas, U.S.N., without whose diligent efforts and outstanding ability to interpret the basic (often vague) guidelines into a dynamic computer program the experiments would undoubtedly still be in the natal stages. Superb guidance, efforts, and encouragement were received from the Defense Language Institute staff, in particular Dr. Francis Cartier, head of research, Dr. Tom Arkwright, computer consultant, and especially Micheline Ponder a research assistant who dug through many volumes of material in order to locate the most applicable literature. Dr. Stephen Jauregui and Dr. Alan Washburn both put me back on track many times, and opened key doors which allowed access to the real experts in this field. Several classmates and friends were also instrumental in various areas particularly in forcing the weight, volume, and milestone criteria. Last but not least to my family, who often thought Dad was deployed again during a large portion of the study.

I. INTRODUCTION

A. LANGUAGE PROBLEMS, GENERAL

There has existed for a great many years the desire and requirement to differentiate languages both categorically and specifically. It is relatively easy to accept that a person can mentally detect an unknown foreign language and guess (often accurately) its origin. However, this mental process has not been successfully duplicated using mechanical/electrical devices is spite of research efforts.

Language barriers have historically been major oppcnents of international well-being. Interpreters bridged many gaps, however, the lack of multi-linguists necessitated the need for a large number of bi-lingual specialists. Therefore, a pool of interpreters is required to conduct the routine business of international relations. Linguist selection is simplified for some physical characteristics, however, it would be quite embarrassing to select a Chinese translator for a Korean dignitary. Modern voice communications eliminate all physical characteristics except those representative of the speaker and his language. The question then, was whether the sound content itself contained sufficient information of a unique quality (language dependent) so as to make it distinguishable from all other possibilities. Certainly it is desirable to establish communications expediently and correctly.

Languages are coded utterances, therefore, it seems possible to differentiate the languages by recognizing the type and quality of code being used. A "guess" as noted above has some merit, but lacks the continual precision available in a machine (computer).

B. WRITTEN LANGUAGES

Written communications have been developed based on either the interpretation of the coded sound or a picture representing the item one desires to convey. Not surprisingly there has been a move in the interest of internationalism to convert the pictorial forms of some written languages to the Romanized script. Most apparent was the recent introduction of "PINYIN" to the Chinese language [Ref. 1]. One intent of the conversion is the introduction of phonetics to the school systems to establish proper pronunciation. The accepted international languages are English, French, and Spanish. Since these languages are Romanized, the conversion from pictorial to Roman suggests that future generations will be less burdened in mastering a second language. In the conversion the ideograms are phonetically transcribed using the IPA (discussed in section II).

Phonetics, then, is a basis through which sounds are transcripted and establishes a common base between the spoken and written word. It is the purpose of this paper to establish the feasibility of automatically differentiating languages based on the average phonemic content over various input lengths. The source data used was various phonetic transcriptions of the selected test languages. The reference data is based on transcripted word frequency (for usage) lists, some text selections, and conversation samples for each of the selected test languages. The reference data is used to establish matrices of probability which measure the sound types and sequences in languages.

C. SPEECH

Speech, as stated earlier, is any length of coded utterances. An utterance is herein defined as the forced emission of modulated air from an oral or nasal orifice. Coding implies that some means is applied to modify the air from its zero state. The zero state is defined as unmodulated inhaled or exhaled air, i.e., normal breathing patterns. The modification or modulation of the air is achieved by the mechanisms of speech to produce audible sound waves. The mechanics of speech are discussed in some detail in Appendices A and B.

The coding or modulation of air flow is classified into the major categories: vowels (V), stops (S), fricatives (F), and nasals (N). These categories are further subclassified into individual sounds which are unique in their generation and in the characteristics of the modulated pressure wave output. A more detailed description of these categories can be found in section 2 of Appendix A.

The uniqueness of certain sounds has been a detrimental aspect to experimenters in the area of synthetic voice generation. Conversely, certain categories have striking similiarities which allow simulator redundancy. In the vowel category, Dr. Rosenblum [Ref. 2] demonstrated how one vowel sound could be converted directly to another by means of selective formant filtering. Other works with specific sound categories are discussed in section 3 of Appendix A.

The pause is a special case of coded output, and is an essential element in speech analysis. Potter, Kopp, and Kopp [Ref. 3] interpreted long strings of sounds as recorded on a spectrogram after they had first recorded the series "word--pause--word--pause" spectrograms and were sure where the word boundaries occurred. In speech the pauses provide a starting and ending point for analysis. The time between pauses is random, as is the actual length of the pause.

The data between the pauses is composed of sounds from the major categories, and for this analysis it was assumed that the sounds could be identified at least categorically, and removed for analysis. The actual conversion of spoken speech into a form for analysis was accomplished indirectly via phonetic transcriptions which is the subject of part III.

II. ANALYSIS TECHNIQUES

In the 90th meeting of the Acoustic Society of America, 150 different papers were presented on various research efforts being done in the voice/speech field [Ref. 4]. The speakers varied from language teachers, phoneticians, etc., to medical doctors, psychologists, physicists, and engineers. They represented various universities, and private organizations from the United States and abroad. There were several papers which looked into differences in languages, but were microscopic in nature. They were concerned with the specific functions of the system which made certain vowel or consonant sounds in one language vary from another, i.e., why a palatilized "i" sound varied from the non-palatilized "i." More specifically is it only the palatilization or are other factors involved. For example some looked at sound clusters of vowel-consonant combinations (VCVVCCV), and focused on the fact that certainly a palatalized "i" is forced by the sound preceeding or succeeding the "i" sound. Other studies were slanted toward speaker recognition schemes. Some were done in languages other than their own, while others looked at recognition criteria in their native languages. These latter results in general hinted that language dependent factors do exist. Note that when transcriptions are used speaker dependence is somewhat removed, i.e., the method of speaking (dialect) is retained, however, the properties unique to the speaker (fundamental frequency, muscle impairment, nose size and shape, etc.) are absent.

III. PHONETIC TRANSCRIPTIONS

A. HISTORY

1. General Background

Using symbols as a means of communications goes back through several civilizations. It is almost academic to determine whether hieroglyphic styles came prior to, concurrently or after vocalized descriptions. The fact is that a means to convey thought other than by vocalization was required, and developed. The various alphabets used throughout the world have certainly proven themselves able vehicles inside of their boundaries, but simultaneously represent an additional hurdle to commonality across language boundaries. There are approximately thirty different alphabets currently being used throughout the world. It was not documented until the ninteenth century that the presently used alphabets were deemed to be deficient. Potter, Kopp, and Kopp [Ref. 3, p. 1] point out that Melvin Bell (father of Alexander Graham Bell) developed a set of symbols which specified the pronunciation of words so accurately, that anyone familiar with the system could repeat the word precisely. Bell's contribution was primarily aimed as an aid for the deaf, although it also made a definite impact on the fields of language, phonetics, and the electronic analysis of speech; e.g., the sound spectrograph was certainly a by-product of the search for a visible means to convey what is heard. A brief description of the sound spectrograph is given in Appendix A.

2. Development of the IPA

Language teachers, in the mid 1800's, recognized certain limitations in their ability to teach proper pronunciation using traditional alphabets. Many teachers circumvented these limitations by developing symbols to describe the type of sound required. As communications and transportation technology bridged the gaps between teachers they began to discover that others were having problems similiar to their own, thus informal teacher collectives were formed. The International Phonetic Association [Ref. 5] attribute their origin largely to the efforts of Paul Passy who headed a small group of language teachers in France in 1886. Passy and his colleagues found that the 26 characters in the Roman alphabet were largely inadequate when trying to teach proper pronunciation. Many sounds for the same letter or combination of letters were in fact generated in the vocal system with distinctly different mechanics. Armed with this information, they drew up a new alphabet to fulfill their needs. The new alphabet was composed entirely of symbols each of which represented a unique sound. Their new alphabet was fully defined and used throughout the text of the first publication of the periodical "Dhi Fonetik Titcer [Ref. 5]." Others were drawn by this unique idea and subsequent events resulted in the formation of the International Phonetic Association. The work of Samuel Bell was done independently of the European group, and his symbols actualy resembled hieroglyphic techniques, rather than an expanded version of the Roman alphabet, which was the objective of Passy. Therefore, Bell's alphabet faded with time. The association grew as numerous inputs were made from various other countries and field-phoneticians. Eventually a collection of phonetic transcriptions of languages (English et al) were assembled, and the IPA was expanded to include sounds unique to previously unavailable language transcriptions. This work across language boundaries enabled many languages to be taught much more easily both in native and second languages. Heretofore, languages could be described by the basic parameters as usual using the unique structures and additionally by those sound production techniques unique to a specific language.

In the IPA every symbol has a distinct sound associated with it. There are some 89 different symbols encompassing all but the African languages as noted by Dr. Francis Cartier [Ref 6]. Only 43 of these symbols appear in American English, (depending on the reference used; general range is 42-46), which makes its transcription much easier. In "The Phonetic Alphabet," Dr. Cartier describes in detail the techniques of transcripting the English language, and has a small treatment of other languages and non-sense words [Ref. 6]. In order that the technique for this research be expandable to eventually include all languages a computer version was developed for 101 IPA characters.

B. THE TRANSCRIPTION

A transcription consists of writing (in symbols) the sounds one makes when speaking. This can be accomplished quite effectively (after training) in either a native language or a language for which adequate knowledge of proper pronunciation is known first-hand. If the language is foreign to the transcriptor there is considerable room for error, particularly if sessions cannot be spread so as to keep the speaker, interpreter, and transcriptor in a fresh state of mind. If a transcriptor is transcripting from an unknown language, which appears close to one he is vaguely familiar with he will tend to do the transcription in the symbols and forms associated with the most familiar language until he has a key to the language being transcribed, then he will retrace and make whatever corrections are required to maintain the quality (uniqueness) of that language [Ref. 7]. This could lead to an erroneous transcription if a key group of words were not located, and/or the language had not been the subject of a detailed analysis to determine which if any sounds were definitely unique to that language. Another problem presented along these same lines is dialect. The simple word "the" can be properly pronounced several ways depending upon the geographical area of the world in which the word is being used. Therefore, it is possible for the transcriptor to write θi when he really should have written ðə.

C. SOURCES AND USES

The transcriptions used for this paper were obtained from a variety of sources. Most of the test text portions were taken from "The Principles of the International Phonetic Association, 1949." There is a second group of text particularly done in a conversational mode type obtained from Micholin Ponder of the Research Department of the Defense Language Institute, Monterey, California. Additional samples of conversational text was made available by Dr. R. Wohlford, and particularly useful English samples were taken from "Informal Speech" [Ref. 8]. A sample of Spanish was contributed by Dr. Maria Baird, Chairman, Spanish Department, DLI, Monterey, Ca.

There are also several dictionaries which use the IPA symbols for their pronunciation guides. One such source for American English is "A Pronouncing Dictionary of American English," by Kenyon and Kott [Ref. 9]. Several foreign dictionaries were scanned as well as language textbooks, in most cases the IPA was used sparingly, making it extremely difficult to collect a satisfactory representation.

IV. DIFFERENTIATION DEVELOPMENT

The actual classification of specific sounds has historically been achieved first by aural perception, and later by electronics devices such as the oscilloscope, the spectrograph, and the computer. All of these devices required manual intervention. That is, the sounds are processed and recorded so that they can be tangibly observed and interpreted for content, classification, etc. Presently no real time conversion exists for the interpretation/classification domain. Developing a routine or algorithm to directly convert spoken speech into some form of usable data does not appear infeasible. There are currently research projects being conducted in this area. Hess [Ref. 10], although not specifically oriented to language differentiation developed a recognition model whose techniques and results may in fact be very applicable to a real time classification converter. For the purposes of this research it was assumed that such a device could be produced.

Although it is not clearly understood how the mind can differentiate languages, it seemed that a model could be developed to produce this differentiation function, not on a "guess" basis but with some reasonable assurance as to which language was being used. Here, it was felt that the sound content might be the key. The research is based on these assumptions and the belief that the IPA represents an appropriate sound classification capturing vehicle.

A. DATA BASE SELECTION

The first problem was to determine a satisfactory data base from which one could establish the categorical sound statistics. Considerable research has been done on frequency factors for use of words under such conditions as:

ordinary (informal) conversation, formal speaking (addresses etc.), informal writing (letters), and formal writing (textbooks, newspapers, periodicals, etc.), and these lists are available in a number of languages. After obtaining such lists it became necessary to convert first into the Roman alphabet, and then to transcribe from the Romanized script into the IPA symbology. As discussed above in the transcription section, this did not propose an easy task for a non-native novice transcriptor. Therefore, solicitations for assistance particularly from DLI, were abundant [refs. 11, 12]. The question arose concerning the possibility of error introduction when using word lists. The major error would seem to appear when the factor of frequency is not utilized within the list itself, i.e., if the frequency of the word were 1/10 then one would expect every tenth word in the list to be the same. A further example is drawn from the Markov process. One such use of the Markov process is to establish the frequency of occurrence of letters (singularly and in combination) for the given language. Then a string is developed using those frequencies (note that in English one considers 27 characters to include the pause). A typical string would appear as:

EAMI HEIBRAGEL----------etc.

The frequency of use is a must in this case, and is generally determined by: (for the single letter case)

$$\text{Frequency(desired letter)} = \frac{\text{(desired letter)}}{\sum \text{(letters)}} \qquad (1)$$

Obviously the string of characters does not make sense. However, when one looks at the strings generated by the two letter combinations [of which there are $(27)^2$ or $(n)^2$], they may find some words (simple) begin to appear. Using the three letter combination level $(n)^3$ more words begin to appear. The process could of course continue ad infinitum and at each step would generate more complex words and perhaps even paragraphs. An analysis very similiar to the Markov Chain type, just described, was used by Rau [Ref. 13], to develop a process by which short samples of written text could be automatically differentiated. Rau looked at two languages, and achieved results, ranging from 51% to 100% accuracy factors which were proportional to the length of the "short samples" he used [Ref, 13, p. 21, ff]. In the case of word lists, it would appear that by not at least using a Markcv process something would be lost. However, in the early testing, the single listings of words generated data very close to that of various random text and conversation inputs. Therefore, it was concluded that a significant error was not introduced by using the most common words independent of their frequency factor.

B. COMPUTER LANGUAGE DECISION

It was apparent that data processing should be accomplished by a computer but the choice of a language (compiler) was not immediately obvious. FORTRAN IV was attempted, but rejected because of the conversions required in the input/output (I/O) for the raw data. The language SNOBOL IV (used in conjuction with SPITBOL) is probably the best language [Ref. 14], however, arrangements for its installation could not be achieved in time to be effective for this research. ALGOL W contains qualities of both FORTRAN and COBOL, and was ultimately selected because of its inherent ability to process both text and equations. The initial program was developed by M. C. Thomas[Ref 15], using strictly English data, and a special alphabet conversicn code (ROMI) consisting of the 44 IPA sounds associated with the English languages. The program developed the statistics for the first sound, the first two sounds, first three sounds, and the first four sounds of the data input. The statistics were developed in the program by reading in the data word by word, then converting the data into the sound categories (F,N,S,V), via ROMI, letter by letter. The converted words were processed to extract the appropriate statistics to produce the data shown in computer output 1.

As languages outside of English were drawn into the data base, ROMI had to be increased, and was ultimately expanded to its present state of 101 characters. The number of symbols (101) decided upon was not entirely the result of the different languages introduced or tested, rather the number was selected as being the universal set recognized by the International Phonetic Association [Ref. 5], plus some diphthongs uniquely identified by Cartier and Todaro [Ref. 6]. Thus ROMI is capable of converting any language subjected to the models. The ROMI to IPA conversion is listed by categories in the appendix C tables.

Five languages were selected at random from among those for which frequency tables and translators or previously transcripted materials were available. Theoretically the languages are independent variables to any algorithym and thus were designated A, B, C, D, and E.

C. FIRST SOUND ANALYSIS

Results similiar to the first data run are shown in computer output 1. The sound pattern of the first letter per word (or phrase, in the conversational cases) showed some promise of a pattern, even though they were bunched together at the lower end of the scale (see figure 1). The words were separated from the phrases to prove (or disprove) similarity, and similarity did occur, thus disproving that an error factor was introduced because of using word lists without frequency considerations.

A decision table was drawn up using ranges established by figures 1-a and 1-b. A new set of data for a single language was then analyzed and its results compared with the table. The test data is indicated by the X's in figure 2.

The results of the single test indicated that the sample size was inadequate. Additionally, it appeared that the ranges would increase proportionately with the sample size, unless a convergence could be determined. If the ranges were to increase, there would be an even greater overlapping of regions, and decisions would be made more difficult. Therefore it was concluded that differentiation was not feasible on the basis of a single sound.

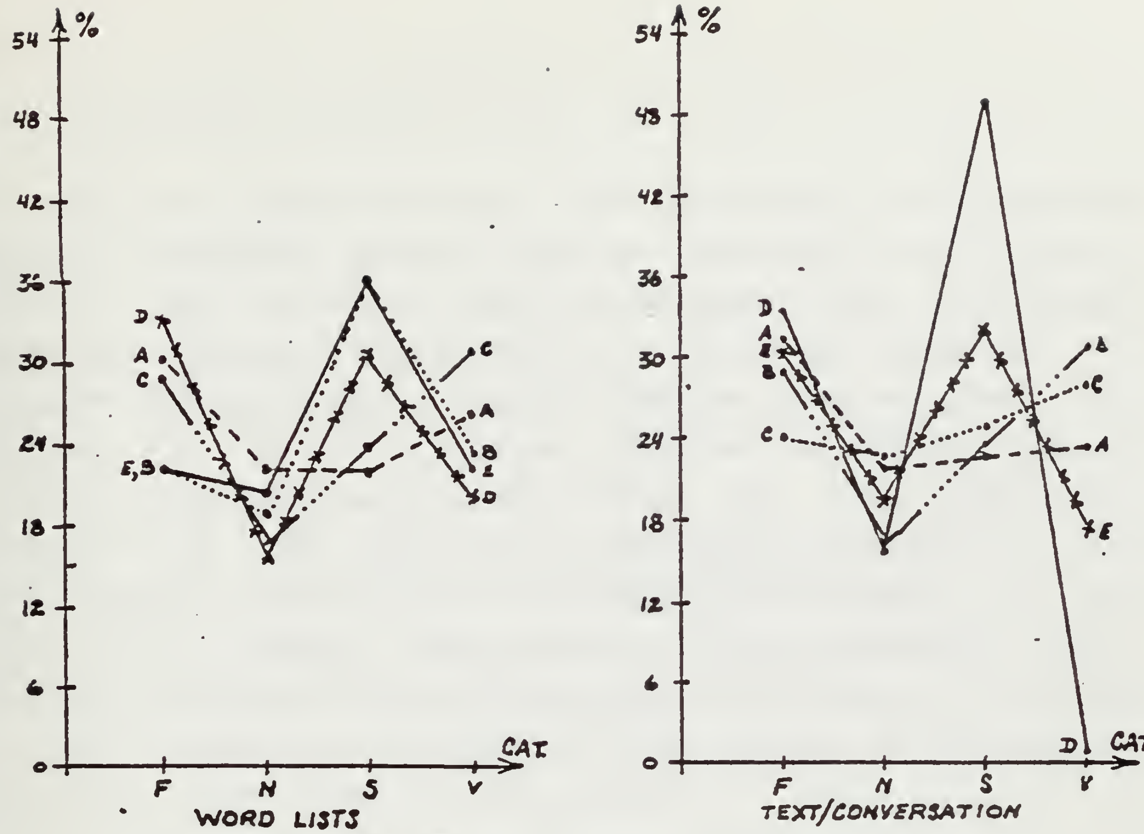


Figure 1. First Sound Frequencies

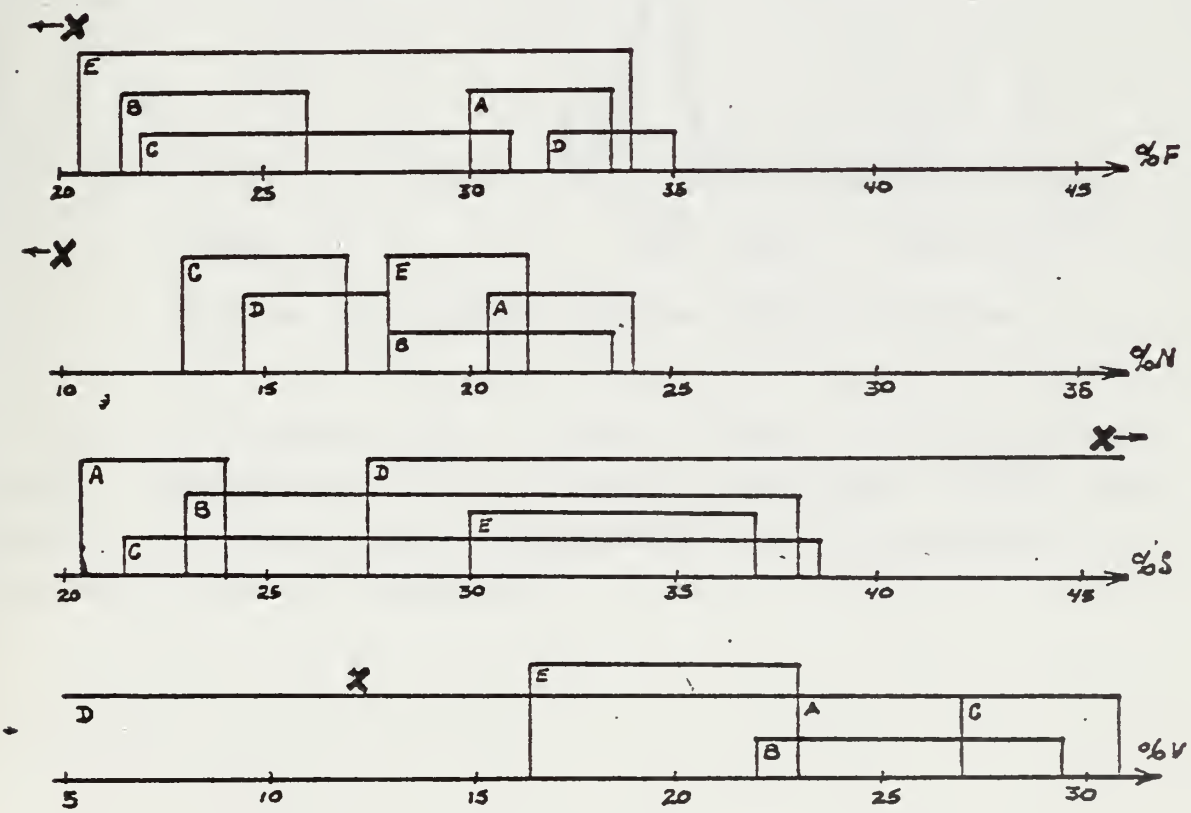


Figure 2. First Sound Decision Table

D. NORMALIZATION CROSS-SECTION

The second approach dealt with observing a cross-section of the languages collectively on the basis of categorical content. This method of analysis presumed that the sounds would be randomly distributed in a "Gaussian" fashion. The decision factor selected was the Shannon Theorem [Ref. 16, chapter 3]. In the Shannon theorem the data is assumed Gaussian, and three situations occur: (1) the curves are disjoint and the decision is immediately obvious; (2) the curves totally overlap and no decision is possible; or (3) there is a slight overlapping and some criterion can be developed which will give reasonable predictions of the case at hand, providing the error factor (amount of overlap) is not severe. The three situations are:

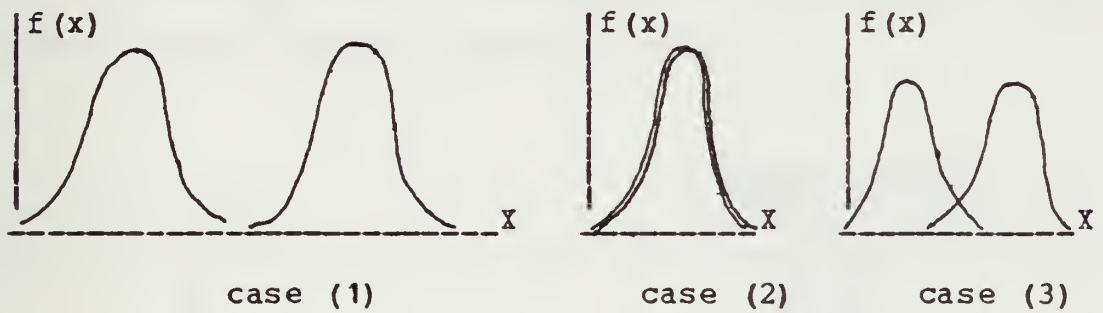


Figure 3. Shannon Theorem Decision Curves

It was anticipated that either case (1) or (3) would emerge. Therefore, the statistics obtained from the computer printout were "normalized" or "Gaussianized" in accordance with the following:

X = the % of occurrence of a category
in the first three and last two
sound positions of a word. % => frequency.

$$E(x) = \frac{\sum (x)}{N} \tag{2}$$

$$\sigma(x) = \left[\frac{\sum [X - E(x)]^2}{N} \right]^{1/2} \tag{3}$$

$$f(x) = \frac{1}{\sqrt{2\pi}\,\sigma(x)} \exp \left\{ - \left[\frac{X - E(x)}{\sqrt{2}\,\sigma(x)} \right]^2 \right\} \tag{4}$$

i = 1, (1,2), (1,2,3), (n-1), (n).

In the case of the fricatives for language "A" the content (from computer output 1) was:

sound location	X = Random Variable = frequency of occurrence of F
1	33.48
2	00.98
3	17.46
4	-----
n-1	15.46
n	01.35

The computed mean is:

$$E(x) = \frac{33.48 + 0.98 + 17.46 + 15.46 + 1.35}{5}$$

= 13.75

Similarily the standard deviation was calculated to be:

$\sigma(x)$ = 10.22

Then using table 6.1 [Ref. 17] the Gaussian density (pdf) for fricatives was established for language A. Using identical procedures the fricative densities were created for the remaining four languages and superimposed on the graph for language A. The same procedure was followed to establish graphs in each category. The tables and graphs for this procedure are contained in appendix D. The results were clearly not what was expected. In fact, Shannon's case (2) is approached in each graph. Even in the graphs where case (3) was there, in spirit, the error factor is clearly greater than the Shannon theorem could accept. It was speculated that increasing the reference data would not be sufficient to obtain more dispersion in the graphs. Therefore, this technique was abandoned.

E. TWO-SOUND PROBABILITIES

The computer program which develops the statistics creates them such that the frequency information follows the tree branch technique, as shown in figure 4.

The frequencies registered at each point in the tree is an exact probability. Therefore at every sound level for each language the probability per language is available. These probabilities are grouped by language in computer output 1, and are summarized in table D-I, in appendix D.

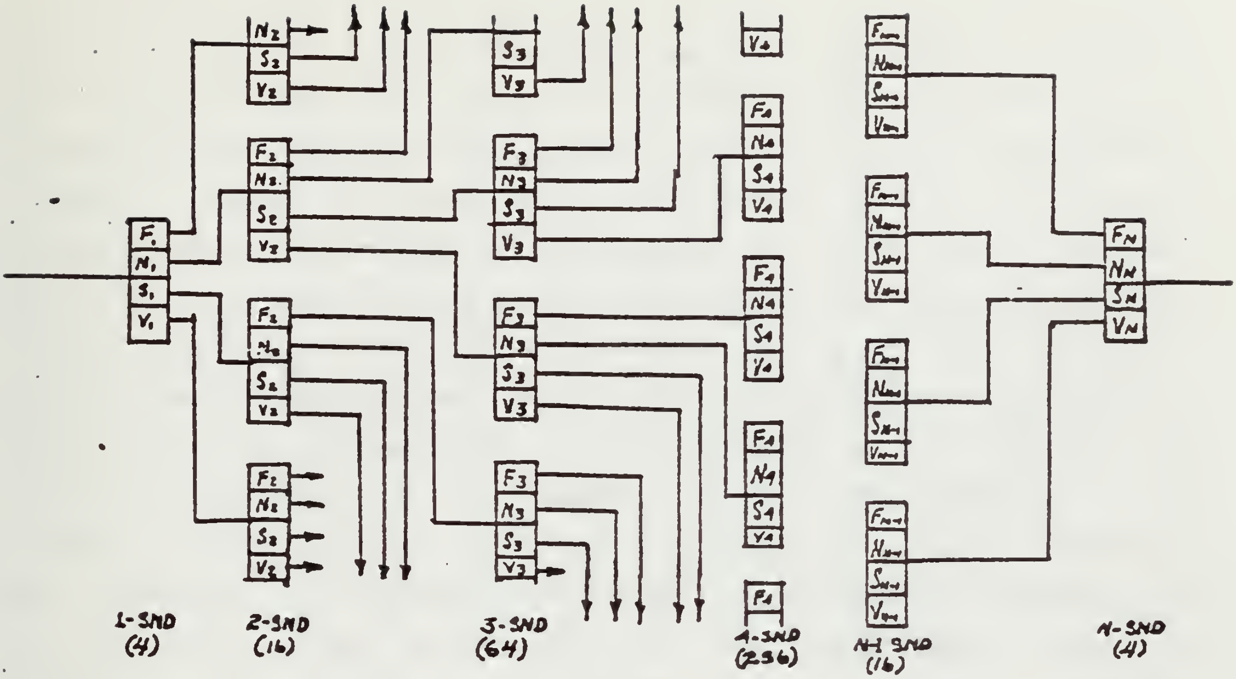


Figure 4. The Frequency Tree.

Assuming that all languages are equiprobable [P(A)=P(B)=P(C)=P(D)=P(E)=1/5], liklihood ratios can be developed using Bayes rule; which computes the probability of a given language based on the categorical sound occurring at a specified position in the word. For example the probability of a test word being from language "X" given that a sound category was identified in position two (second sound) would be computed as follows:

$$P(X|Y_2) = \frac{P(Y_2|X)\ P(X)}{P(Y_2|A)P(A) + P(Y_2|B)P(B) + P(Y_2|C)P(C) + ..} \quad (5)$$

where: X = language = A, B, C, D, E

Y_2 = sound category = F, N, S, V

i = position indicator = 1, 2, 3, 4, n-1, n

j = position indicator = 1, 2, 3, 4, n-1, n

These probabilities are tabulated in table IV-I. A decision can be achieved using the maximum likelihood. For example, from table IV-I, for an "S" in the 2nd position one would choose language E because it clearly has the the greatest probability of being the correct choice. In the case where more than one word from the same language is being analyzed the decision criteria requires an additional iteration; this presumes the words have different sound content arrangements (order). When the position is selected (say at the second sound position) the rows are removed in tact. The removed rows are then isolated and a product is calculated for the column elements. The correct choice is then the columnar product which is largest, i.e., has the greatest likelihood. Multiplication corresponds to the assumption that words are chosen independently at random from the language. For example, given a first word has a "V" in the last position the maximum likelihood would elect language B, however, if the second word had an "S" in the last position and the product is taken, language D will have the maximum likelihood.

V	.074	.313	.229	.126	.258
S	.220	.000	.137	.491	.153
product	.016	.000	.031	.062	.039
sum	.294	.313	.466	.617	.411

Using the product method, clearly eliminates any language which is devoid of a unique sound. The advantage of the summing method is that languages which have high percentage of occurrence are brought out for consideration in spite of a categorical void. Arguments can be established on which is the better method (sum or product), however, it was decided to use the product method in subsequent decisions. The product of one or more column elements is referred to as "validity factors" in computer program 2; a product of validity factors are called "validity products."

In spite of the closing statement in section IV-C, that differentiation was not feasible on the basis of a single sound, testing was performed on some 200 random words/phrases. The results (not included) were ambiguous and supported the earlier claim.

Bayes Rule still offered more potential than previous efforts, therefore, the two-sound combinations [of which there are 16; F, N, S, V => $(4)^2$], were computed in the following manner:

$$P(X|Y_2Y_1) = \frac{P(Y_2Y_1|X)P(X)}{\sum_{X=A,E}[P(Y_2Y_1|X)_i P(X)_i]} \quad (6)$$

where: X = language = A,B,C,D,E

Y = sound category = F,N,S,V

P(X) = P(A)=P(B)=P(C)=P(D)=P(E), by assumption

These values are given in Table IV-II.

The decision rule is identical to that for one-sound, i.e., maximum likelihood. Given any word(s) greater than one sound, a decision can be reached. Again two or more words dictate the product (or sum) rule to obtain the maximum likelihood.

The results of limited testing (30 words) using this method were below the desirable limit. In the case of one random word, decisions were less than 20% correct; for two words less than 50% correct; and three word decisions ranged from 45-55% correct. Though the correct decisions were proportional to the number of words, the decision procedure appeared to be weak. Presuming this was a result of the earlier assumption that all languages were equiprobable, a procedure similiar to Bayes was constructed and defined as follows:

$$L(X|Y_2 Y_1) = \frac{[P(Y_2 Y_1|X)P(X)][P(Y_1|X)P(X)]}{\sum [P(Y_2 Y_1|X)P(X)][P(Y_1|X)P(X)]}$$

$$= \frac{P(Y_2 Y_1|X)P(Y_1|X)[P(X)]^2}{\sum_{X=A,E}\{[P(Y_2 Y_1|X)_i P(Y_1|X)_i [P(X)_i]^2\}} \quad (7)$$

where: X = A,B,C,D,E

Y = F,N,S,V

Again the assumption of equiprobable languages was inserted thus removing the $[P(X)]^2$ terms. The remaining equation is hereby defined as "Prognostic Progression." The equation calculates the "Logical Prognosis" of a language given the frequency of occurrence at a position and the probability [or Logical Prognosis (L)] brought forward from the previous position in the tree (fig. 4). Table IV-III was calculated using this procedure, and is compared with the Bayes solutions in Table IV-IV. The results are interesting. They show that the resultants are shifted, from one table to the next, and tend to negate the assumption of language equiprobability. Thus the probability of language arrivals is introduced, but indirectly, at any selected level except the first (single sound). This enables the use of variable probability for reference languages, i.e., unknown reference sample sizes are inherently accounted for. Three comparative tests were run for Tables IV-III versus IV-II using 100 randomly selected words (20 per language). The results showed for the two-sound level that the formulas had relatively the same effect. The overall results achieved were: (note, 100 word samples increased minimum correct for this test to 40% compared to previous 20%)

	% CORRECT		
Procedure	test 1	test 2	test 3
BAYES	39	40	38
COOKS	41	39	40

Further test at the three-sound level were conducted; results very similiar to the two-sound results were obtained. The number of correct choices were within 1--4% of each other, however, it was noted that in 4 of 7 tests (avg. words = 35) the magnitude of the correct over the next competitor was much greater with Cook's method. Additionally, the order of magnitude from the most likely (maximum product) to the second most likely, second to third, etc., was much more even, and smaller in Bayes than in Cooks. It was thus assumed that the idea of "Prognostic Progression" was at least equivalent to Bayes Rule. This assumption encouraged the writing of computer program number 2.

The basic operation of computer program 2 is a continuation of program 1, except that some extra features were added. The primary extra feature is variable partitioning, wherein words of the same sound length are analyzed collectively by language. The partitioning is variable in that the statistics can be ended at any point, i.e., all words ≥1, create the same data as in computer output 1; words ≥2, creates data for one sounded words, apart from the data for words of two sounds and greater; this process is allowed through words ≥6. The decision portion forces a word to follow a flow path based on the number of sounds the word contains. By doing this, for example, three sounded words are only compared with all other three sounded words in the data collection, unless the the partitioning is set at ≥2 or ≥3. The standard test (comparisons) in the flow path for a partitioning of ≥6, is provided in table IV-VII, and the program flow diagram is provided in figure 5. Secondary features include selectable graphic representations of either the raw statistics, or the Prognostic Progression values for any or all reference data.

The model was tested using randomly selected words from each language within the data base. This "loop" test was used to determine if the model could correctly segregate and identify what it already held in storage.

The first-two position results related a noticeable similariity between the languages, and required at least a two word sample size to achieve correct decisions in 50% of the tests. The results of this test are tabulated in table IV-V. Tests involving the first-three sounds showed even greater improvement in the results. This indicated that at the three-sound level unique patterns were emerging. The test of three-sounds duplicated the procedure used for the two-sounds and their results are tabulated in table IV-VI.

Testing beyond this point was considered insignificant due to the limited reference data. Beyond three-sounds, the tests were approaching 100% for as few as five word/phrase test samples. The consensus was that even though patterns were uniquely developing, their percentage relative to the whole of a language negated the significance of the patterns.

A final group of tests was run with the program changed to use Bayes' Rule, rather than the Prognostic Progression. The results (not included) were for all practical purposes identical. The Bayes' Rule method also forced 100% correct decisions as early as five words. On the basis of this test it is concluded that Bayes' Rule and Cook's Prognostic Progression perform equally on this type of task.

Table IV-I.

Bayes' Theorem Calculations for one-sound

Y_i	$P(A\|Y_i)$	$P(B\|Y_i)$	$P(C\|Y_i)$	$P(D\|Y_i)$	$P(E\|Y_i)$
F_1	.237	.204	.142	.179	.235
N_1	.179	.181	.223	.260	.157
S_1	.281	.167	.185	.140	.227
V_1	.084	.250	.266	.252	.148
F_2	.035	.161	.374	.169	.261
N_2	.070	.350	.234	.148	.200
S_2	.111	.029	.185	.269	.405
V_2	.270	.173	.169	.236	.153
F_3	.230	.129	.224	.254	.162
N_3	.284	.160	.197	.202	.157
S_3	.078	.300	.208	.130	.383
V_3	.174	.215	.187	.212	.212
F_4	.041	.186	.217	.302	.253
N_4	.146	.263	.213	.096	.283
S_4	.499	.083	.131	.140	.148
V_4	.130	.231	.223	.252	.163
F_{n-1}	.022	.396	.151	.214	.216
N_{n-1}	.144	.169	.296	.154	.237
S_{n-1}	.051	.328	.162	.179	.280
V_{n-1}	.358	.098	.178	.233	.134
F_n	.293	.138	.067	.297	.205
N_n	.396	.102	.226	.194	.096
S_n	.220	.000	.137	.491	.153
V_n	.074	.313	.229	.126	.258

Table IV-II.

Bayes' Theorem Calculations
for two-sounds

Y_i Y_j	$P(A\|Y_i\ Y_j)$	$P(B\|Y_i\ Y_j)$	$P(C\|Y_i\ Y_j)$	$P(D\|Y_i\ Y_j)$	$P(E\|Y_i\ Y_j)$
F_1 F_2	.000	.000	.000	.000	1.0
F_1 N_2	.037	.388	.154	.068	.353
F_1 S_2	.282	.000	.000	.137	.581
F_1 V_2	.303	.199	.188	.190	.120
N_1 F_2	.000	.000	.000	.000	.000
N_1 N_2	.000	.161	.400	.000	.438
N_1 S_2	.000	.000	1.0	.000	.000
N_1 V_2	.200	.163	.179	.328	.130
S_1 F_2	.192	.000	.000	.351	.457
S_1 N_2	.062	.324	.243	.151	.221
S_1 S_2	.000	.000	.000	.000	1.0
S_1 V_2	.298	.192	.154	.141	.216
V_1 F_2	.000	.334	.443	.125	.098
V_1 N_2	.119	.276	.285	.246	.075
V_1 S_2	.044	.052	.403	.081	.420
V_1 V_2	.188	.109	.097	.570	.037

Table IV-III.

Cook's Prognostic Progression Calculations for two-sounds

Y_i Y_j	$L(A\|Y_i\ Y_j)$	$L(B\|Y_i\ Y_j)$	$L(C\|Y_i\ Y_j)$	$L(D\|Y_i\ Y_j)$	$L(E\|Y_i\ Y_j)$
F_1 F_2	.000	.000	.000	.000	1.0
F_1 N_2	.045	.359	.114	.062	.421
F_1 S_2	.293	.000	.000	.108	.599
F_1 V_2	.363	.184	.138	.172	.142
N_1 F_2	.000	.000	.000	.000	.000
N_1 N_2	.000	.144	.483	.000	.372
N_1 S_2	.000	.000	1.0	.000	.000
N_1 V_2	.171	.130	.192	.409	.098
S_1 F_2	.253	.000	.000	.231	.516
S_1 N_2	.085	.334	.219	.103	.260
S_1 S_2	.000	.000	.000	.000	1.0
S_1 V_2	.373	.181	.126	.088	.232
V_1 F_2	.000	.325	.486	.130	.060
V_1 N_2	.045	.290	.339	.276	.050
V_1 S_2	.018	.059	.521	.099	.302
V_1 V_2	.073	.118	.119	.664	.026

Table IV-IV.

Shifting Effects Denoted
by the Difference in (V-III) - (V-II)

	A	B	C	D	E
FF	.000	.000	.000	.000	.000
FN	.008	.029(-)	.040(-)	.006(-)	.068
FS	.011	.000	.000	.029(-)	.018
FV	.060	.015(-)	.050(-)	.018(-)	.022
NF	.000	.000	.000	.000	.000
NN	.000	.017(-)	.083	.000	.066(-)
NS	.000	.000	.000	.000	.000
NV	.029(-)	.033(-)	.013	.081	.032(-)
SF	.061	.000	.000	.120(-)	.059
SN	.023	.010	.024(-)	.048(-)	.039
SS	.000	.000	.000	.000	.000
SV	.075	.011(-)	.028(-)	.053(-)	.016
VF	.000	.009(-)	.043	.005	.038(-)
VN	.074(-)	.014	.054	.030	.025(-)
VS	.026(-)	.007	.118	.018	.118(-)
VV	.115(-)	.009	.022	.094	.011

Table IV-V.

Two-Sound Results
using Prognostic Progression

ONE-WORD TEST:

Lng	smpl size	A	B	C	D	E
A	33	.637	.121	.000	.182	.061
B	25	.240	.400	.08	.120	.016
C	49	.367	.204	.163	.265	.000
D	42	.286	.143	.000	.571	.000
E	60	.317	.050	.150	.150	.333

TWO-WORD TEST:

Lng	smpl size	A	B	C	D	E
A	16	.563	.125	.000	.313	.000
B	12	.083	.583	.083	.083	.167
C	24	.167	.292	.208	.330	.000
D	21	.238	.095	.000	.667	.000
E	30	.133	.067	.133	.167	.500

Average correct single word = 83/209 = .397

Average correct two words = 50/90 = .556

Sample sizes of 15 and greater gave results of 100% correct selection in 4 of 5 tests.

Table IV-VI.

Three-Sound Results
using Prognostic Progression

ONE-WORD TEST:

Lng	smpl size	A	B	C	D	E
A	33	.667	.033	.100	.233	.067
B	25	.160	.480	.160	.160	.040
C	49	.306	.041	.327	.265	.061
D	42	.214	.071	.048	.595	.071
E	60	.217	.083	.133	.167	.400

TWO-WORD TEST:

Lng	smpl size	A	B	C	D	E
A	16	.688	.000	.000	.313	.000
B	12	.083	.580	.167	.167	.000
C	24	.167	.042	.500	.292	.000
D	21	.095	.048	.048	.810	.000
E	30	.100	.067	.100	.167	.567

Average correct single word = 97/209 = .464

Average correct two words = 64/103 = .621

Sample sizes of 11 and greater gave results of 100% correct selection in all tests.

Table IV-VII.

Model Test Path
for
A Given Word Length

word length	Test Criterion
1.	$P(x)$
2.	$P(x)$, $L(y_1 y_2 \| x)$
3.	$P(x)$, $L(y_1 y_2 \| x)$, $L(y_1 y_2 y_3 \| x)$
4.	$P(x)$, $L(y_1 y_2 \| x)$, $L(y_1 y_2 y_3 \| x)$, $L(y_1 y_2 y_3 y_4 \| x)$
5.	$P(x)$, $L(y_1 y_2 \| x)$, $L(y_1 y_2 y_3 \| x)$, $L(y_1 y_2 y_3 y_4 \| x)$, $L(y_1 y_2 y_3 y_4 y_5 \| x)$
6.	$P(x)$, $L(y_1 y_2 \| x)$, $L(y_1 y_2 y_3 \| x)$, $L(y_1 y_2 y_3 y_4 \| x)$, $L(y_1 y_2 y_3 y_4 y_5 \| x)$, $L(y_5 y_6 \| x)$, $L(y_1 y_2 y_5 y_6 \| x)$, $L(y_1 y_5 \| x)$, $L(y_1 y_2 y_3 y_4 y_5 y_6 \| x)$

where: x = language = A,B,C,D,E
y = sound category = F,N,S,V
1 = first sound
2 = second sound
3 = third sound
4 = fourth sound
5 = last sound = n
6 = next to last sound = n-1

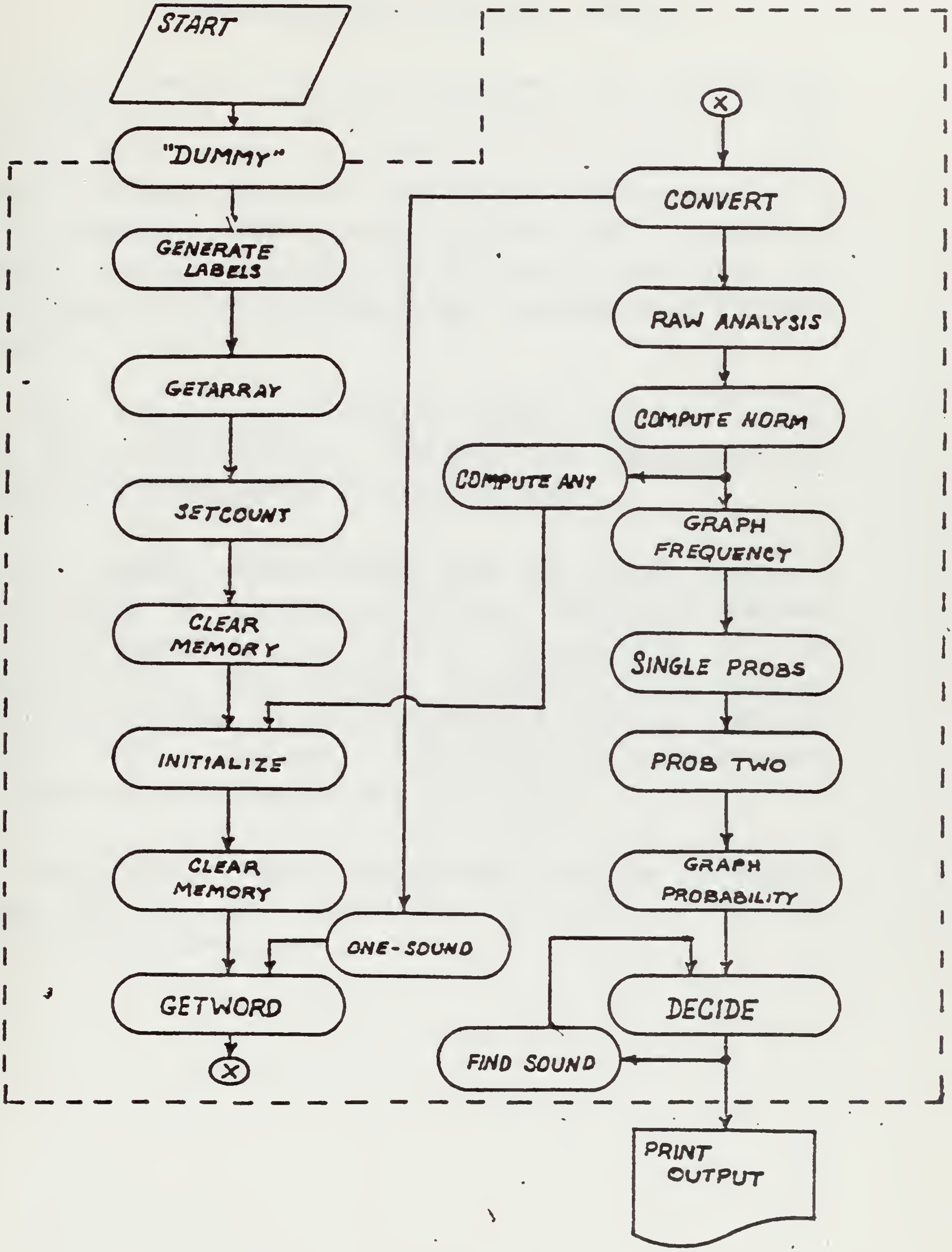


Figure 5. Final Model Flow Diagram

V. RECOMMENDATIONS FOR FUTURE STUDIES

1. The model finally developed has considerable potential which was not exhaustively exploited herein. The reference data should be vastly expanded, or at least altered to reflect the most current language/vocabularies in use. 1000-1500 words and/or phrases per language is suggested, because only 20% of that range was used here, and it appeared that word uniqueness was overshadowing language uniqueness.

2. The Prognostic Progression formula (no. 7) should be tested at greater lengths to identify its uniqueness, or qualify it as an extension of Bayes Theorem.

3. The largest problem encountered was in not having a spoken speech to phoneme converter. Assuming a similiar lack of availability for future researchers, it is recommended that high quality transcriptions be obtained earlyon, and the quantity of transcripted words be used as a criterion for continuance. Further phrase transcriptions will provide better results than word lists.

4. Diacritical marks and emphasis were not considered herein, but may prove beneficial in further language differentiation developments.

VI. CONCLUSION

The following are presented in relation to the findings herein:

1. The IPA does appear to be an able vehicle to support a language differentiation scheme. However, the ROMI conversion alphabet should have been expanded to reflect diacritical marks across the languages used.

2. Single sounds are definitely an inadequate differentiation approach. Two-sound combinations present only borderline results. At the three-sound point definite uniqueness appears, and is improved with each additional sound brought into the combinations. The ending sounds (backdoor approach) begin to present patterns of interest at the two-sound point, and should have been explored at greater length. For the languages used herein (all European) it is generally concluded that patterns emerged which were unique in spite of the limited sample size.

3. The final analysis and decision model (computer program 2) is considered a successful tool for language differentiation. It appears to have considerable potential for expanded analysis.

4. Perhaps the most important, though unexpected, result was obtained via the "loopback" testing of the model. This test pointed out the model was capable of pattern matching decisions of high reliability with small test samples(<5). This certainly presents an improvement to the technique of word matching, in that storage requirements and accessing times are greatly reduced. This potential is limited only by the ability to update the data base relative to current desired operability.

APPENDIX A

THE SPEECH SYSTEM

When discussing the speech system, it is quite possible to delve into many scientific fields of endeavour in order to fully exploit the interaction and functions of the components. In an overview the players in a system diagram analysis (figure 1) are the transmitting section and the receiving section.

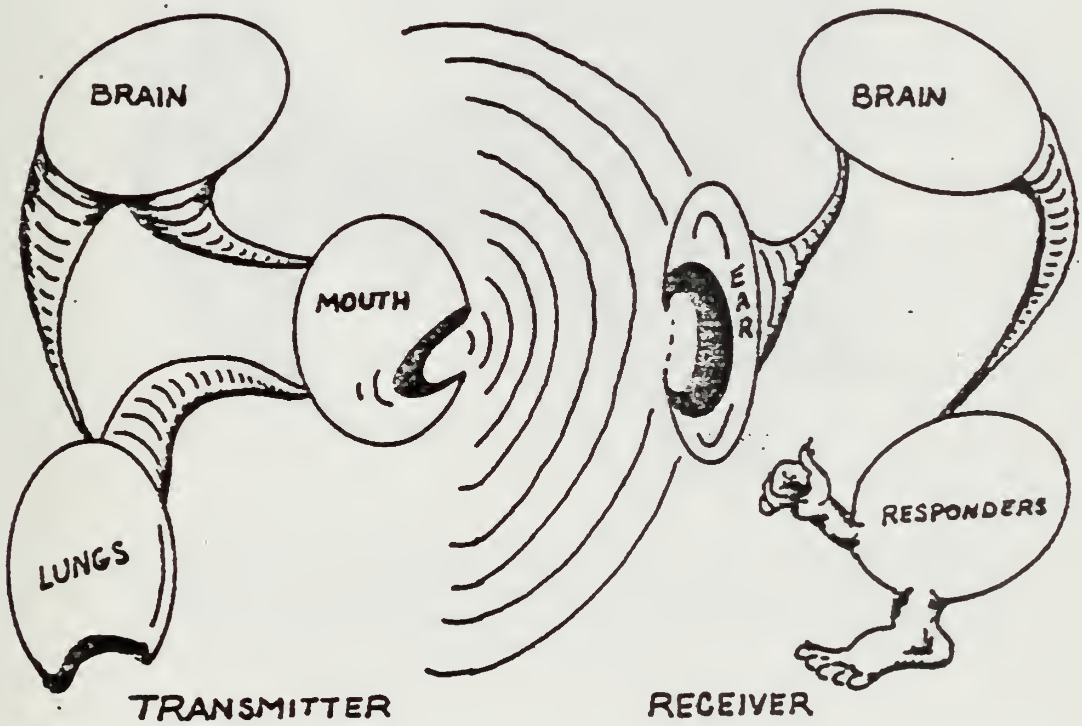


Figure A-1. Block Diagram of the Speech System

The receiving section is comprised of the hearing organ, the brain, and the responders. A schematic of the hearing system (figure 2) shows a subdivision into (1) the outer ear, (2) the middle ear, and (3) the inner ear.

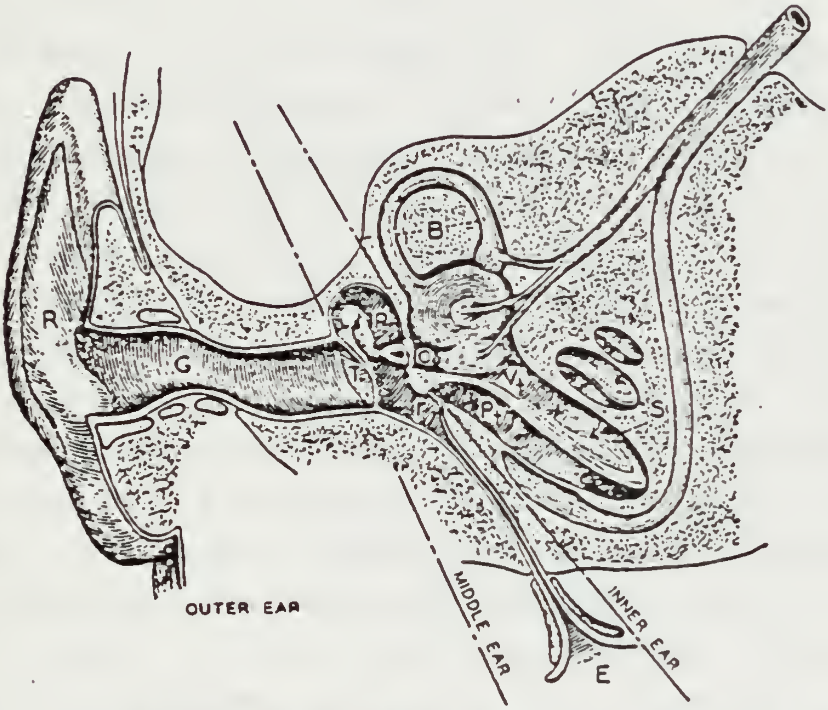


FIG. 75.—SEMIDIAGRAMMATIC SECTION THROUGH THE RIGHT EAR (CZERMAK): G, EXTERNAL AUDITORY MEATUS; T, MEMBRANA TYMPANI; P, TYMPANIC CAVITY; O, FENESTRA OVALIS; R, FENESTRA ROTUNDA; B, SEMICIRCULAR CANAL; S, COCHLEA; VT, SCALA VESTIBULI; PT, SCALA TYMPANI; E, EUSTACHIAN TUBE; R, PINNA.

Figure A-2. The Hearing System

The outer ear functions as an antenna, and both ears used together comprise a directional antenna system i.e., capable of determining the direction of the sound source relative to the listener. The outer ear also contains the feed system to the receiver. The middle and inner ear comprise the receiver itself. The middle ear senses the presence of sound and passes the mechanical energy to the inner ear, where it is converted to electrical impulses modulated in both amplitude and frequency. These impulses are then sent to the brain which detects the intelligence, and activates decision circuits to provide physical response as necessary. The Eustachiam tube (E in figure 2) insures that there is equal pressure on both sides of the membrana tympani (eardrum). The Eustachian tubes also provide some side-tonal pressures whenever speech or sound is generated at its input (upper part of the throat behind the nasal cavity). This side-tonal effect appears to provide a negative feedback to "pad" or reduce the effects of the amplitude of sound waves emitted when they arrive at the receiver input of the "transceiver" system. An analogy to the "TR/ATR" action in a radar is certainly suggested by the side-tone effect. The functions of the receiving mechanism are those for which a mechanized language differentiation model is desired. The available technology does not dictate a direct approach to the solution, however, the basics are clear in the "blackbox" approach. The output is a response resulting from a decision based on the input received. The type of decision device, must then be determined from what information can be gleaned from the input. The input is assumed to be an exact duplication of the output from the transmitter section. Therefore, if it can be determined how the transmitter functions based on observable output, it seems feasible that some model could be developed to interpret the information the transmitter would send subsequently.

The transmitting section is of much greater interest and value to this research. Basically, as seen in figure 1, the brain sends responder messages to the lungs and vocal shaping components. The lungs provide the mechanical energy which is shaped, amplitude and frequency modulated, and emitted as speech or vocalization. Thus the transmitter is the human vocal system. A block diagram of the system (figure 3) gives the breakdown of the vocal system into four subdivisions. The schematic diagram (figure 4) gives the anatomical layout of the components.

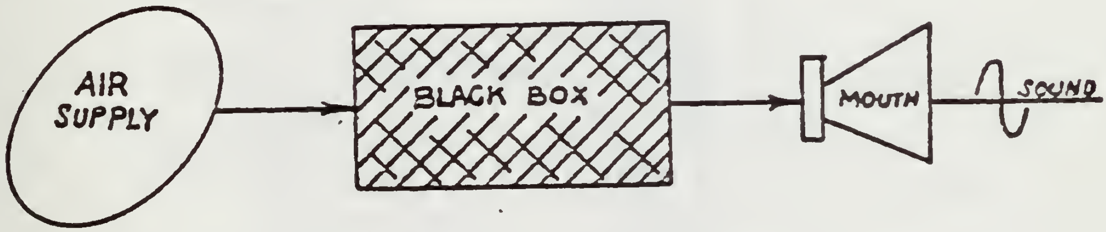


fig. A-3.a. The "Black-box" Approach

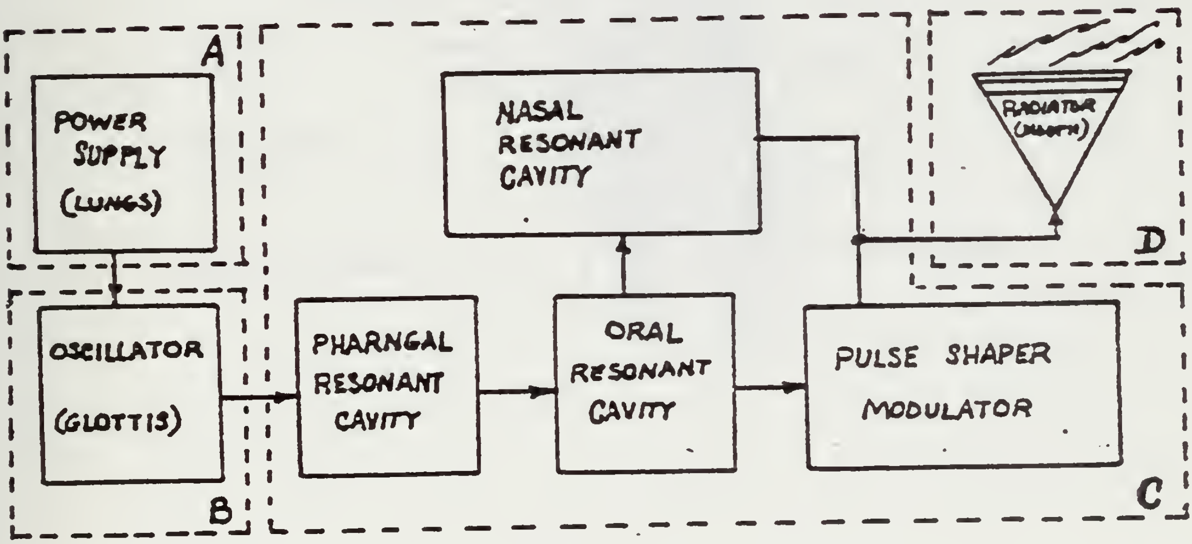


fig. A-3.b. The "Black-box" Expansion

Figure A-3. Human Vocal System
(block diagram)

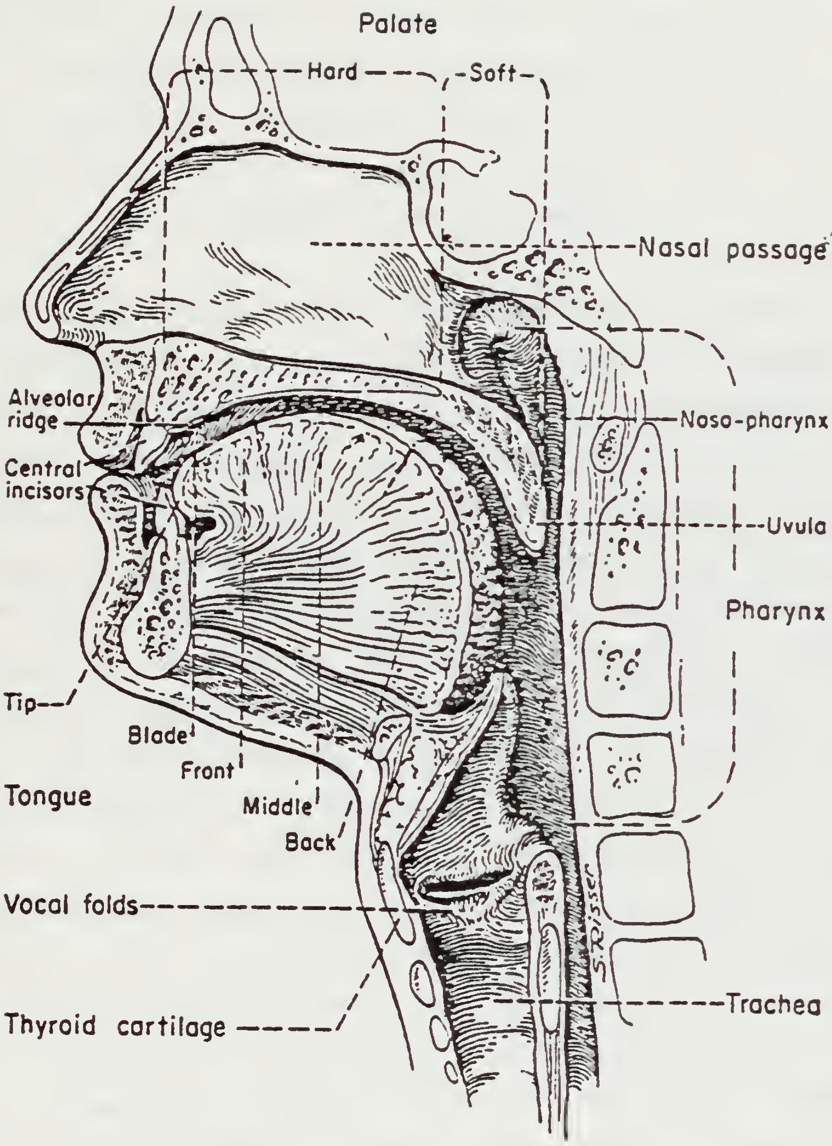


Figure A-4. Human Vocal System
(schematic diagram)

The lungs act as a power supply, collecting air such that it can be expelled with the required volume and velocity whenever dictated by the brain. Both the voluntary and involuntary actions by the lungs are used in the speech process. Voluntary action is very apparent in the "raised" voice or "yelling" process, where there is an obvious increase of pressure on the lungs forcing the volume and velocity out of the lungs to be increased.

The oscillator and resonant cavities consist of the glottis (vocal chords), the pharyngeal passage (upper throat), the oral cavity (the mouth), and the nasal passage. The glottis, when excited by an expulsion of air, vibrates or oscillates at a fundamental frequency (male average is 130Hz, female average is 205Hz) [Ref. 18], and generates many harmonics. The frequency actually emitted may be the fundamental, a harmonic, or some combination. In general, when sounds are emitted there is an amplification of some parts of the spectrum and suppression of others, much like the characteristics of a bandpass filter, i.e., selective transmission. The fundamental frequency is a result of a number of factors, namely the size, shape, and elasticity of the glottis, the circumference, length, and shape of the trachea, and the size shape, and depth of the oral and nasal cavities; specifics for typical dimensions and frequencies relative to these paramenters can be found in appendix B. In the aggregate these factors constitute a tuned waveguide and/or cavities which tend to alter the basic frequency of the glottis vibrations. In support of this, consider what occurs when the velum is opened (by its trapdoor action) to allow air to flow through the nasal cavity; the volume of the system is increased which is consistant with a lower frequency of resonance normally attributed to the nasalized tones.

The pulse shaper provides the ultimate control of power during constant output. The shapers consist of the tongue, teeth, lips, jaw, and palate. The shapers affect both the frequency and amplitude of the transmitted waves ànd thus act very similiar to a modulator section in a radio transmitter. Frequency variability occurs in the range available from the fully open to the fully closed mouth, and has numerous descrete frequencies in between these extremes.

The antenna or radiating system consists of the actual oral and nasal orifices. The lips and jaw both work independently and together in the shaping, frequency, amplitude, and emission operations. The parameters of importance in the output are again frequency and amplitude. The sound wave emitted will be of higher frequency when the mouth is more in the closed position than open. The affects of the nasal chamber is controlled by the velum, which determines the amount of air (by volume) allowed to pass in the cavity. The nasal-oral combinations regulate the output frequency by combining the orifice controls of the velum and the tongue position. These combination sounds occur more frequently in languages other than English.

RECOGNITION OF SOUNDS

Every emission via the vocal system falls into categories which have been previously developed and quantized in varying degrees. Numerous experiments have been conducted particularly in the speaker recognition area where the uniqueness within these categories are further quantized to search out factors which are undeniably speaker dependent, and thus provide keys to recognition algorithms. Doherty [Ref. 19] used 25 significant factors in his Long Term Spectra (LTS) analysis to achieve 100% recognition (a rarity in recognition results) for a sample size of 50. Doherty did not provide a specific breakdown of the 25 factors he used, however, he did indicate that various combinations of frequency, times, nazalization, pauses, stops, and fricatives were some of the factors used. Others such as Young [Ref. 20] have used a single category of sound to promote recognition schemes. Young looked specifically at nazalization and achieved results from 50% to 97% with a sample size of 13. The categories are relative then to recognition of many entities in speech. Word/phrase boundaries as well as the individual components and sound patterns can be identified and removed for analysis. Experimenters make every effort to eliminate odds whenever possible, therefore, it is not uncommon to find that when a project is dedicated to a specific area of sound to research that test segments are carefully screened to be representative of the specific area. Sounds can be classified by a number of parameters (time, frequency, form, etc.) One such classification technique categorizes sounds by form, i.e., by the way they are generated.

The categories of concern are: the vowels (v), the stops (s), the nasals (n), the fricatives (f), and the pauses (p).

The vowel sounds are that group which flow smoothly (uninterupted) from the lungs through the oral orifice utilizing the whole of the vocal cavity, with minor adjustments in resonance promoted by the specific location of the tongue. Vowels are further defined according to the tongue position with respect to the oral cavity (mouth) in general. The front vowels are generated whenever the tongue position toward the front of the mouth is the major factor; the sounds specific to this location are: i(HF), I(LHF), e(HMF), Ɛ(LMF), æ(LF). The back vowels are: u(HB), U(LHB), O(MB), ɔ(HLB), a(LB). The central vowels are: ɝ(MC), ɚ(MC), 3(MC), ə(MC), ʌ(LMC). In addition to the distinctive vowels noted above are a group of connected vowels whose separation are said to alter the meaning, therefore they are taken collectively and have the special sub-categorization name "diphthongs." The diphthongs are aI, aU, ɔI, and ju. Diphthongs are also referred to as the vowel-glides since there is an alteration in the resonance which resemble the glide consonants.

The stop category consists of those sounds which are emitted as a 'puff-of-air, i.e., no energy is followed by a large spike of energy. Stops are also known as plosives due to the energy spike, but stop is more commonly used since the speaker must actually establish a stop in the smooth expulsion of sounds to generate the plosive burst. The stop category has the components: p, b, t, k, and g. A sub-category of the stops is the affricates tʃ and dʒ which are again combinations which cannot be reduced without changing the meaning, therefore, they are classified as unique and essential.

The third group is known as the fricatives. These sounds require a velocity change which is achieved by creating a narrow slit between the upper teeth and the lower lip, the upper teeth and the tongue, or the upper teeth with the lower teeth. The objective is to create a higher frequency sound at a higher velocity. The components of this category are: f, v, θ, ð, s, z, ʃ, ʒ, and h. At present there are no apparent sub-categories which are considered unique.

The nasal category actually consists of glides, nasals, and laterals. The glide is so called because of articulation organ movement required in its generation. For example, note that when the single sound of "r" is generated the oral cavity enlarges, specifically note that the tongue drops in position. Enlarging the vocal cavity tends to lower the frequency of resonance, such that the sound is readily identifiable as a smooth transition from a high to a lower frequency. The nasal sounds are those sounds which are emitted via the nasal cavity only, that is the mouth is blocked by a tight closing of the lips for the sound of "m," and by the tongue pressed firmly against the hard palate for the generation of "n." The final component of this sub-category is generated when the mouth is blocked by the center and back of the tongue to form the sound associated with ŋ. The lateral has a single component "l." The lateral utilizes both the nasal and oral cavities for the generation of this glide-like sound.

The final category is that of pauses. A pause is an obvious lack of sound of sufficient duration so as to provide a well-defined boundary for a word or phrase as appropriate. Pauses really represent a special situation that requires a more indepth discussion.

The pause is that period in speaking which is denoted by the total absence of sound. In writing one provides word separations or boundaries by the absence of script or by leaving an intentional space. Unfortunately this does not follow in speaking. In speech, words are run together such that a combination of words (hereforth defined as a phrase) would appear as a single utterance which is devoid of pauses. For example, the phrase "the brown cow" is spoken as "thebrowncow." In a transcription from the text the phrase becomes "ðə braun kəu," but the spoken appears as "ðəbraunk u." This draws forth the problem of sound pattern significance for the two cases described as (1) with and (2) without pauses.

It appears as though a pause may or may not be present with equal probability. In the case of English it is observed that three distinct areas enter into the study of pause insertion. Speaking generally for the U.S.A., the Northeast coast population speak very rapidly and pauses are difficult to detect; Southerners speak very slowly, so that pauses are distinct and occur more often; Westerners/mid-westerners fall somewhere in between, representing a median for the extremes. The actual detection of word boundaries via the spectrograph or similiar process has not been achieved with any degree of accuracy, and is particularly poor with an unknown sample [Ref. 21]. Therefore the problem must be looked at from two points of view. The first looks at the sound patterns based on individual word list inputs. The premise here is that the probability of any word beginning a phrase could be high. From the words of the phrase "the brown cow" one would certainly expect the article "the" to follow a pause with a high degree of certainty. The word "brown" could be given a probability assignment of .5 (equiprobable that it would or would not begin a phrase with some certainty). The word "cow" would have a much lower degree of certainty associated with it, but greater than zero, since words such as "cowboy," etc. exist. A second argument would be that only phrases (devoid of pauses) exist and they therefore, should be the only sound strings used in the analysis.

The solution to these points as far as this thesis was concerned was to look at both words and phrases, individually and collectively. The rationale being that since the analysis only looks at the first four and last two of the sound string, and since there is sufficient probability that any given word may be the first word in a given string (phrase), that the characteristics of that individual word are very germane to the sound string analysis. The use of phrase analysis, recognizes that pause detection may be very applicable in the conversion of spoken sounds to a computer-acceptable input.

A series of sound emissions can be observed either in whole or in part wherein individual sounds can be sensed categorically or uniquely, dependent upon the type of analysis which is desired. In the analysis by Young [Ref. 20] he was able to concentrate on the nasal phonetation by first carefully selecting words (including nonsense words) which clearly contained significant nasal sounds, then via a masking or filtering technique he removed those quantities which were not of concern to him. Others have used a variety of masking techniques to eliminate all but the desired elements of the test speech.

The concern herein was not to extract any sounds or rely on the uniqueness of sounds for the purpose of recognition. Rather the intent was to establish whether or not sound patterns are unique to languages, on either an individual or a categorical basis. In order to achieve this end it is necessary that sounds can be uniquely defined categorically.

ELECTRONIC PERCEPTION OF SOUND

Many attempts to fully comprehend the operation of the vocal system have been made through the ages. A large number of models were constructed across time, many were unsuccessful, while others proved major contributions. In the late 1700's three mechanical models were independently produced by Kratzenstein, VonKempelen, and Wheatstone, each of which used bellows, leather resonators, and reeds to create fricative, plosive, and vowel sounds [Ref. 18]. The first real electrical model of any value was the "VODER," (VOice Operation DEmonstratoR), developed by H. Dudley in the 1930's, (Dudley also developed the first VOCODER: VOice CODER; also in the 1930's). The VODER consisted of oscillators, resonance chambers and a keyboard to control the frequency, amplitude, and duration of each sound in the formulation of words at its speaker output. The VODER had a vocabulary of 2500 words [Ref. 22, pp. 16, 32, 49, 191].

The spectrograph is an analysis tool which came into being in the 1940's as a result of studies by Koenig, Dunn, and Lacy [Refs. 3 and 23]. There are various types now being used, however they have common operations in that all take in speech sounds, filter separate the sound into bands (bandpass filtering) and display the results on a frequency (vertical) versus time (horizontal) basis. The spectrograph uses basically a Fourier analysis processing [Ref. 3]. One motivating factor behind the development of the spectrograph was the desire to have a visible speech mechanism to aid the learning process for deaf persons. Many side uses were subsequently developed for the spectrograph. Peterson and Barney used the spectrograph technique to confirm that most English sounds occur in the standard telephone bandwidth of 300-3300Hz [Ref. 18, p. 154]. Spectrograms have been used as a means to develop speaker identification through "voiceprints." Voiceprints are unfortunately not at a state where 100% accuracy is consistant with experimental results. Therefore, voiceprints promote (presently) a shadow of doubt in their reliability and are not accepted as a legal tool. The voiceprint area is one of considerable interest to the speaker recognition subgroup of the Acoustic Society of America, and was the subject of attention during a special open discussion during the 90th meeting in San Francisco [Ref. 9].

APPENDIX B

Some Characteristics of Vocal Tract Transmission

FORMANT = manifestation of the normal modes as spectral peaks in the output sound, which are derived from the transmission function relating the mouth and glottal volume currents. The transmission function is:

$$H(jw) = \frac{1}{\alpha l} = \frac{1}{j \sinh \alpha l} = \frac{1}{\cosh(\alpha + j\beta) l} = \frac{1}{\cos\left(\frac{wl}{c}\right)}$$

where = w/c and w = $(2n + 1)\frac{\pi}{2}\frac{c}{l}$

Formant Parameters:

$L_a = \frac{\rho}{A}$ = acoustic inertance/unit length

$C_a = \frac{A}{\rho C^2}$ = accoustic compliance/unit length

$R_a = \frac{S}{A^2}\sqrt{\frac{\omega \rho \mu}{2}}$ = acoustic resistance/unit length

$G_a = \frac{S\ \eta - 1}{\rho C^2}\sqrt{\frac{\lambda \omega}{2 C_p \rho}}$ = acoustic conductance/unit length

A = tube area = $5cm^2$ = $0.775\ in^2$

r = radius of tube = 1.26cm = 0.497in

S = tube circumference = 7.9265cm = 3.12in

ρ = air density = $1.14 \times 10^{-3} gm/cm^2$,
(moist air at body temperature = 37 C)

C = sound velocity = $8.5 \times 10^4 cm/sec$,
(moist air at body temperature = 37 C)

μ = viscosity coefficient = 1.86×10^{-4} dyne-sec/cm^2,
(20 C, 0.76 in.Hg.)

λ = coefficient of heat conduction
= 0.055×10^{-3} cal/cm-sec-deg (0 C)

η = adiabatic constant = 1.4

c_p = specific heat of air at constant pressure
= 0.24 cal/gm-deg (0 C, 1 atmos.)

l = tube length = 17cm = 6.693in (for avg. male)

$$f = \frac{\sigma}{\pi} = \frac{\alpha C}{\pi} = \alpha(11141)$$

$$\alpha = \frac{R_a}{2}\sqrt{\frac{C_a}{L_a}} + \frac{G_a}{2}\sqrt{\frac{L_a}{C_a}}$$

The formant frequencies as computed are:

$$w = (2n + 1)\left(\frac{\pi}{2}\right)\left(\frac{C}{l}\right) \qquad f = (2n + 1)\left(\frac{C}{4l}\right)$$

$$= (2n + 1)(3230/\text{sec}), \qquad = (2n + 1)(514.4),$$

$$n=0,1,2,... \qquad n=0,1,2,...$$

w = 3230 radians — f = 514.4 Hz

w = 9690 radians — f = 1543.3 Hz

w = 16160 radians — f = 2571.9 Hz

w = 29070 radians — f = 4630 Hz

APPENDIX C

TRANSCRIPTION SYMBOLS

ROMI to IPA Conversion Tables

Table C-I. Fricative Conversion

ROMI	IPA
F	f
V	v
Q	θ
T-H	ð
S	s
Z	z
S-S	ʃ
Z-Z	ʒ
H	h
Q-A	ɸ
B-B	β
C-H	C
S-H	S
Z-H	Z
X	χ
G-F	g
X-X	x
R-B	ʁ
H-H	ħ
U-R	ɹ
Z-C	ʑ
J-A	j
G-B	γ

Table C-II. Nasal Conversion

ROMI	IPA
M	m
N-N	n
N-J	ŋ
L	l
W	w
H-W	hw
J	j
R-R	r
R	R
M-J	ɱ
J-N	[illegible]
N-Q	[illegible]
N	N
A-N	[illegible]
K-Z	[illegible]
Y-U	[illegible]
L-L	[illegible]
R-T	ɽ

Table C-III. Stop Conversions

ROMI	IPA
P	p
T	t
B	b
D	d
K	k
G	g
C-C	c
F-F	ɟ
T-T	ʈ
D-D	d
Q-Q	q
G-G	G
N-U	?
AFFRICATIVES	
T-S	tʃ
D-Z	dʒ
P-F	pf
B-V	bv
T-L	ts
T-M	dz
T-C	tc
K-X	kx

Table C-IV. Vowel Conversion

ROMI	IPA
I-I	i
I	ɨ,I
E-E	e
E	ɛ
A-E	æ
A	a
C	ɔ
O	o
U	U
U-U	u
V-V	ʌ
U-T	ɜ
U-H	ə, (ɛ)
E-R	ɚ
A-A	a
A-B	ɒ
3-R	ɜ
Y-Y	y
Y	Y
O-O	ɸ
O-E	œ
G-A	ɤ
W-W	ɯ
I-T	i
DIPHTHONGS	
A-I	aI
C-I	I
J-U	ju
A-U	aU
O-U	oU

Table C-IV. (cont.):

ROMI	DIPHTHONGS	IPA
C-E		ɔɚ
E-Q		ɛɚ
A-R		aɚ
I-R		I
U-S		U
E-S		ɛə
I-U		I
U-Q		U
C-Q		ɔə
E-I		eI

APPENDIX D

The data and graphs contained in this appendix were used in an attempt to achieve a decision model to differentiate languages. In each language the data reading across the sound rows are normalized to 100; the columns were not normalized, however, the columnar data were used to compute the mean E(x), and the standard deviation $\sigma(x)$. The graphs were generated in accordance with formula (4).

Table D-I. Combined Vector Data

Lng	Sounds	F	N	S	V	#wds
A	1	33.48	16.29	39.82	10.41	221
A	1-2	0.98	7.84	5.88	85.29	204
A	1-2-3	17.46	45.24	7.14	30.16	126
A	1-2-3-4	-----	-----	-----	-----	
A	n	15.46	54.63	11.34	18.56	97
A	n-1	1.35	17.57	5.41	75.68	74
A	E(x)	13.75	28.31	13.92	44.02	
A	$\sigma(x)$	10.22	18.21	13.12	30.58	
B	1	28.87	16.49	23.71	30.93	97
B	1-2	4.55	39.39	1.52	54.55	66
B	1-2-3	9.80	25.49	27.45	37.25	51
B	1-2-3-4	-----	-----	-----	-----	
B	n	7.31	14.63	0.00	78.05	41
B	n-1	24.14	20.69	34.48	20.69	29
B	E(x)	14.93	23.34	17.43	44.29	
B	$\sigma(x)$	9.71	8.85	14.05	20.15	

Table D-I (cont.):

C	1	20.61	20.33	26.18	32.87	359
C	1-2	10.53	26.32	9.72	53.44	247
C	1-2-3	17.01	31.44	19.07	32.47	194
C	1-2-3-4	-----	-----	-----	-----	
C	n	3.53	32.35	7.06	57.06	170
C	n-1	9.22	36.17	17.02	37.59	141
C	E(x)	12.18	29.32	15.81	42.67	
C	σ(x)	6.01	5.48	6.83	10.48	
D	1	25.31	23.65	19.92	31.12	241
D	1-2	4.76	16.67	14.17	74.40	168
D	1-2-3	19.27	32.11	11.93	36.70	109
D	1-2-3-4	-----	-----	-----	-----	
D	n	15.66	27.71	25.30	31.33	83
D	n-1	13.04	18.84	18.84	49.28	69
D	E(x)	15.61	23.80	18.03	44.57	
D	σ(x)	6.81	5.65	4.67	16.31	
E	1	33.22	14.29	32.22	18.27	301
E	1-2	7.36	22.48	21.32	48.24	258
E	1-2-3	12.27	25.00	25.90	36.82	220
E	1-2-3-4	-----	-----	-----	-----	
E	n	10.84	13.79	7.88	64.49	203
E	n-1	13.16	28.95	29.47	28.42	203
E	E(x)	15.37	20.90	23.36	39.85	
E	σ(x)	9.14	5.97	8.56	16.97	

Using the data from table D-I and the following information from table 6.1 of Lipschutz[Ref. 17], points were calculated for the magnitude coordinates at 0, ±1, ±2, and ±3 standard deviations, for each category:

$0 = \frac{0.4}{\sigma(x)}$ $\pm 1 = \frac{0.242}{\sigma(x)}$ $\pm 2 = \frac{0.054}{\sigma(x)}$ $\pm 3 = \frac{0.0044}{\sigma(x)}$

Table D-II. Fricative Magnitudes

	-3	-2	-1	0	1	2	3
A,X	-16.91	-6.69	3.53	13.75	23.97	34.19	44.41
A,F(x)	.0004	.005	.024	.039	.024	.005	.0004
A,σ(x)				10.22			
B,X	-14.2	-4.49	5.22	14.93	24.64	34.35	44.06
B,f(x)	.0005	.006	.025	.041	.025	.006	.0005
B,σ(x)				9.71			
C,X	-5.85	0.16	6.17	12.18	18.19	24.2	30.21
C,f(x)	.0007	.009	.040	.067	.040	.009	.0007
C,σ(x)				6.01			
D,X	-4.82	1.99	8.8	15.61	22.42	29.23	36.04
D,f(x)	.0006	.008	.036	.059	.036	.008	.0006
D,σ(x)				6.81			
E,X	-12.05	-2.91	6.23	15.37	24.51	33.65	42.79
E,f(x)	.0005	.006	.026	.044	.026	.006	.0005
E,σ(x)				9.8			

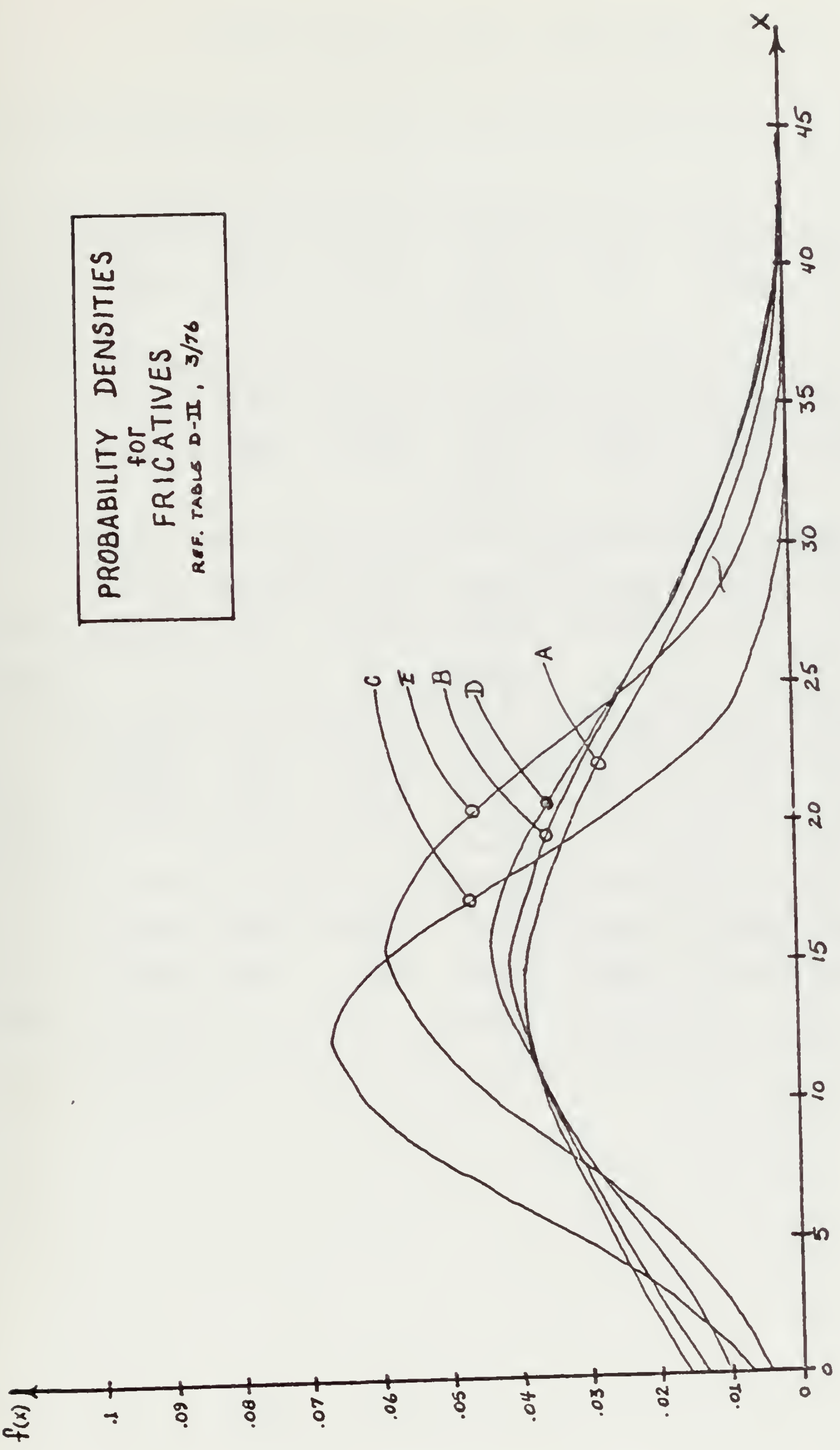

PROBABILITY DENSITIES
for
FRICATIVES
REF. TABLE D-II, 3/76
f(x)
.1
.09
.08
.07
.06
.05
.04
.03
.02
.01
0
0
5
10
15
20
25
30
35
40
45
X
A
B
C
D
E

Table D-III. Nasal Magnitudes

	-3	-2	-1	0	1	2	3
A,X	-26.32	-8.11	10.1	28.31	46.52	64.73	82.94
A,f(x)	.0002	.003	.013	.022	.013	.003	.0002
A, σ(x)				18.21			
B,X	-3.21	5.64	14.49	23.34	32.19	41.04	49.89
B,f(x)	.0005	.006	.027	.045	.027	.006	.0005
B, σ(x)				8.85			
C,X	12.88	18.36	23.84	29.32	34.80	40.28	45.76
C,f(x)	.0008	.01	.044	.072	.044	.01	.0008
C, σ(x)				5.48			
D,X	6.85	12.5	18.15	23.80	29.45	35.10	40.75
D,f(x)	.0007	.010	.043	.071	.043	.010	.0007
D, σ(x)				5.65			
E,X	2.99	8.96	14.93	20.90	26.87	32.84	38.81
E,f(x)	.0008	.009	.041	.067	.041	.009	.0008
E, σ(x)				5.97			

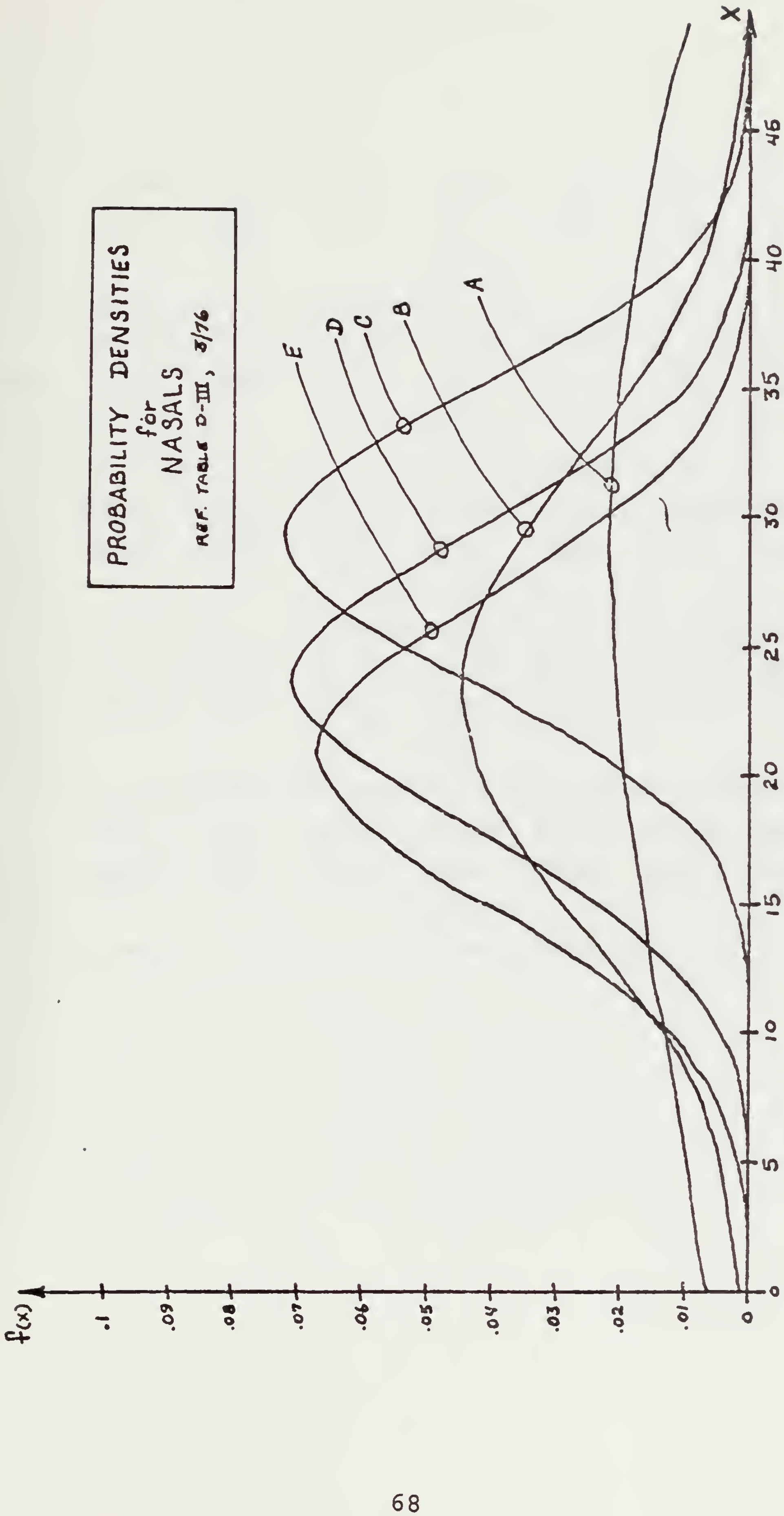

PROBABILITY DENSITIES
for
NASALS
REF. TABLE D-III, 3/76
A
B
C
D
E
f(x)
X
.1
.09
.08
.07
.06
.05
.04
.03
.02
.01
0
0
5
10
15
20
25
30
35
40
45

Table D-IV. Stop Magnitudes

	-3	-2	-1	0	1	2	3
A,X	-25.44	-12.32	0.80	13.92	27.04	40.16	53.28
A,f(x)	.0003	.004	.018	.031	.018	.004	.0003
A, σ(x)				13.12			
B,X	-24.72	-10.67	3.38	17.43	31.48	45.53	59.58
B,f(x)	.0003	.004	.017	.028	.017	.004	.0003
C, σ(x)				14.05			
C,X	-4.68	2.15	8.98	15.81	22.64	29.47	36.30
C,f(x)	.0006	.008	.035	.059	.035	.008	.0006
C, σ(x)				6.83			
D,X	4.02	8.69	13.36	18.03	22.70	27.37	32.04
D,f(x)	.0009	.012	.052	.086	.052	.012	.0009
D, σ(x)				4.67			
E,X	-2.32	6.24	14.8	23.36	31.92	40.48	49.04
E,f(x)	.0005	.006	.028	.047	.028	.006	.0005
E, σ(x)				8.56			

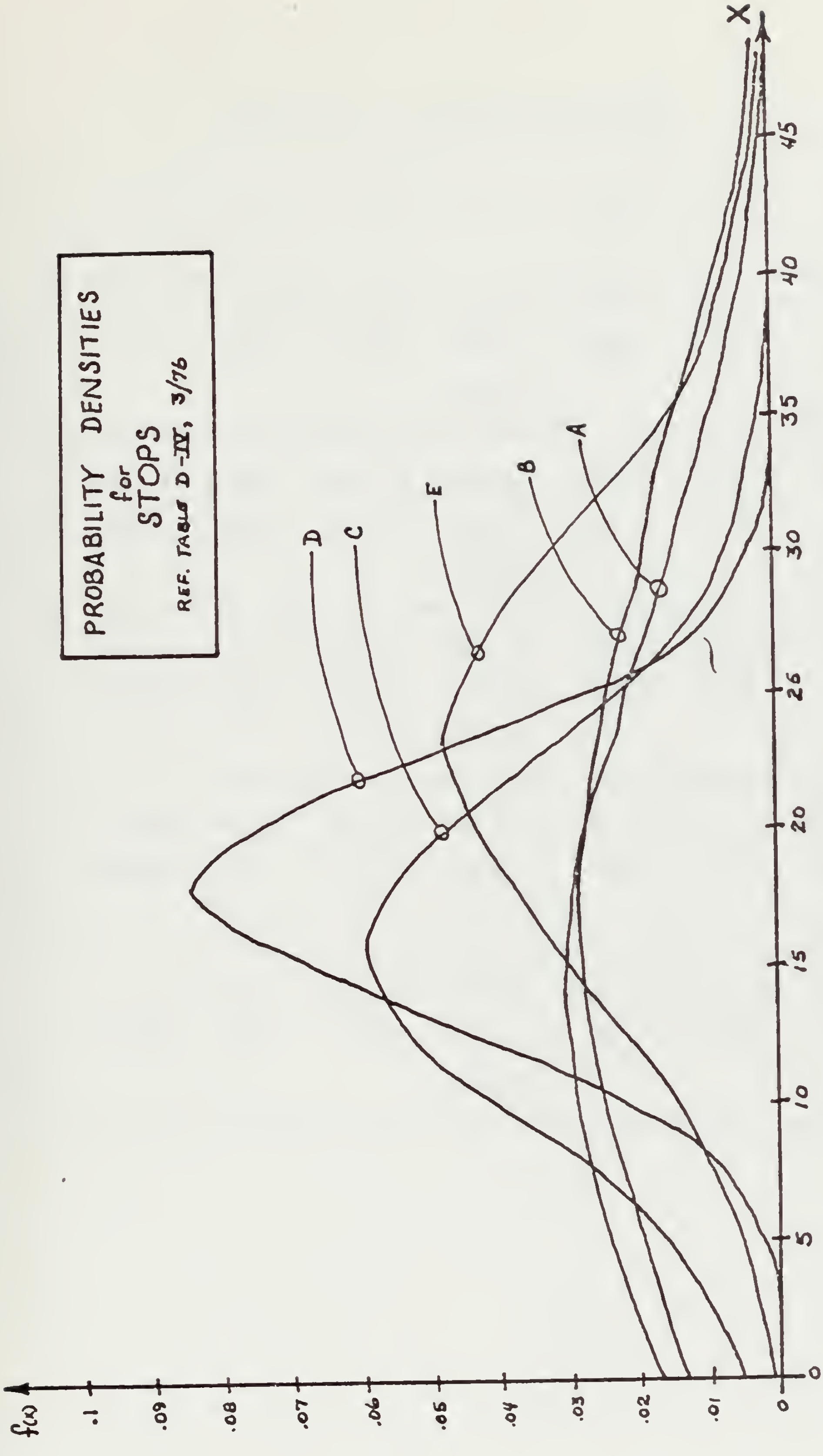
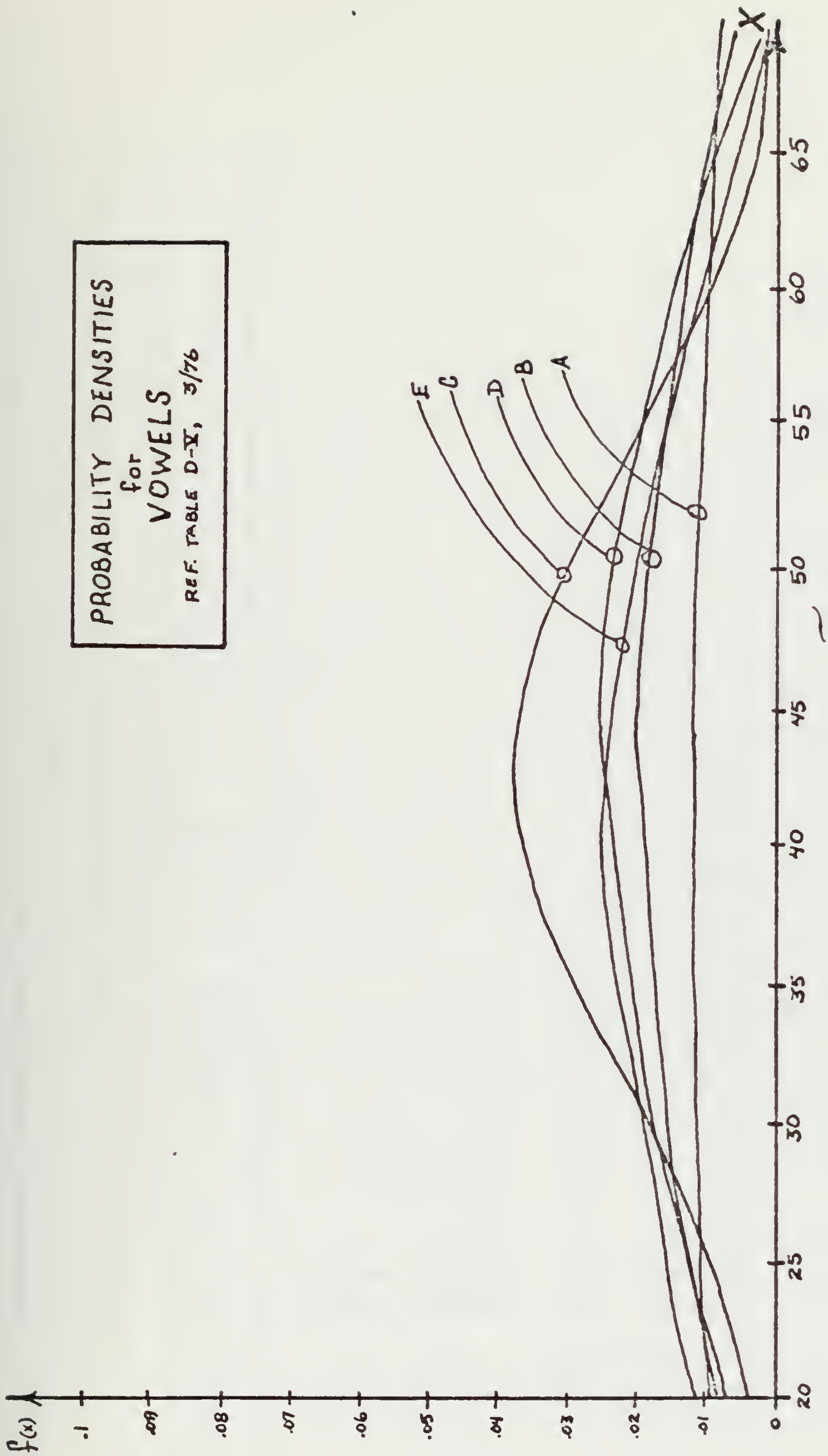

PROBABILITY DENSITIES
for
STOPS
REF. TABLE D-IV, 3/76
A
B
C
D
E
f(x)
.1
.09
.08
.07
.06
.05
.04
.03
.02
.01
0
X
0
5
10
15
20
25
30
35
40
45

Table D-V. Vowel Magnitudes

	-3	-2	-1	0	1	2	3
A,X	-47.72	-17.14	13.44	44.02	74.60	105.18	135.76
A,f(x)	.0001	.002	.008	.013	.008	.002	.0001
A,σ(x)				30.58			
B,X	-16.16	3.99	24.14	44.29	64.44	84.59	104.74
B,f(x)	.0002	.003	.012	.020	.012	.003	.0002
B,σ(x)				20.15			
C,X	11.23	21.71	32.19	42.67	53.15	63.63	74.11
C,f(x)	.0004	.005	.023	.038	.023	.005	.0004
C,σ(x)				10.48			
D,X	-4.36	11.95	28.26	44.57	60.88	77.19	93.5
D,f(x)	.0002	.003	.015	.025	.015	.003	.0002
D,σ(x)				16.31			
E,X	-11.06	5.91	22.88	39.85	56.82	73.79	90.76
E,f(x)	.0003	.003	.014	.024	.014	.003	.0003
E,σ(x)				16.97			


PROBABILITY DENSITIES
for
VOWELS
REF. TABLE D-V, 3/76
f(x)
.1
.09
.08
.07
.06
.05
.04
.03
.02
.01
0
20
25
30
35
40
45
50
55
60
65
X
E
C
D
B
A

```
****************************************************************************************

          C O M P U T E R   O U T P U T   NUMBER 1

****************************************************************************************


****************************************************************************************
'ROMI' IS THE PHONETIC CHARACTER REPRESENTATION USED IN THE
IMPLEMENTATION OF THE CONVERSION FROM THE IPA INPUT INTO THE FIVE CLASSES.

****************************************************************************************

A V   A-A V   A-B V   A-E V   A-I V   A-N F   A-R V   A-U V
B S   B-B F   B-V S
C V   C-C S   C-E V   C-H F   C-I V   C-Q V
D S   D-D S   D-Z S
E V   E-E V   E-I V   E-Q V   E-R V   E-S V
F F   F-F S
G S   G-A V   G-B F   G-F F   G-G S
H F   H-H F   H-W N
I V   I-I V   I-R V   I-T V   I-U V
J N   J-A F   J-N N   J-U V
K S   K-X S   K-Z F
L N   L-L N
M N   M-J N
N N   N-J N   N-N N   N-Q N   N-U S
O V   O-E V   O-O V   O-U V
P S   P-F S
Q F   Q-A F   Q-Q S
R N   R-B F   R-R N   R-T N
S F   S-H F   S-S F
T S   T-C S   T-H F   T-L S   T-M S   T-S S   T-T S
U V   U-H V   U-Q V   U-R F   U-S V   U-T V   U-U V
V F   V-V V
W N   W-W V
X F   X-X F
Y V   Y-U N   Y-Y V
Z F   Z-C F   Z-H F   Z-Z F
3-R  V
```

```
*****************************************************************************

                         LANGUAGE = "A"

*****************************************************************************

 VVNFS  FSNVSVN  FVF  NVNSFVNS  VNS  FVNV  FVN  FVN  VNVN  SVVSVN  FVN
SVN   FSVNSVNV  FVNV  VNF  VVN  FVNSVNVN  SVN  VN  VVNVN  FVNNVN
NVNSVN   SVFVNS  FVN  SVF  FVSVF  SV  FVN  SVN  FV  FVNSVN  VVNVF  SVF
SVN   NVNVSV  FVN  SVN  FSVNSVNVN  SVNSVN  FVNSV  SVN  SVN  FVNSVNVN
SFVNVN   FVNSV  FVVNVN  NVNSVN  VSSVNVNVN  SVN  NVNSFVNS  SNVF
NVS   VNVN  NVFS  VSVN  NV  NVN  VN  SNVF  SVFSV  FVFSVN  FVNSV  FVF  SVN
FVNSVNVN   VN  FVVNVN  NVNSVN  VVN  VNSNVF  SVS  SVN  NVNSFVNS  SVN  SVNSF
VVF   NVN  VNFVNNSV  SV  FVNV  SV  NVFS  NVS  VNVN  FNVVNSNVFVN  FSNVNVN
VNS   FVN  NVF  FVNVSVN  VVSVNSNVSVN  SVS  SVN  FVNSVNVN  FVVNVN  NVNSVN
VVF   SV  NVFSV  SVN  NVNSFVNS  SVSVSVN  SVF  SV  FVNV  FVN  VNVN  SVVSVN
SVN   FSVSVNV  FVN
SVVNFS   FSNVSVN  FVF  NVFSFVNS  NVNS  FVNV
FVF   FVN  SVNVN  SVVSVN  FVN  SVF  FSVFSVNV
FVNV   SVNF  SVVN  FVNSVNVF  SVF  SVN
SVVNVN   FVFNVN  NVNSVN  SV  FVNS  FVF  SVF
FVSVF   SVFVF  SVN  FV  FVFSVN  SVVNVF  SVF
SVFNVNVSV  FVF  SVN  FSVFSVNVN  SVNSVN
FVNSV   SVF  SVN  FVNSVNVN  SFVNVN
FVFSV   FVVNVN  NVNSVN  SVSSVNVNVN
SVF   NVFSFVNS  SNVF  NVS  SVNVF  NVFS
SVSVF   NV  NVF  SVF  SNVF  SVFSV  FVFSVF
FVNSV   FVF  SVF  FVNSVNVF  SVN  FVVNVN
NVNSVN   SVVN  SVNSNVF  SVS  SVF  NVFSFVNS
SVN   SVNSF  SVVF  NVN  VNFVFNSV  SV  FVNV  SV  NVFS
NVS   SVNVN  FNVVNSNVFVN  FSVVNVN  SVNS
FVN   NVF  FVNVSVN  SVVSVNSNVSVN  SVS  SVF
FVNSVNVF   FVVNVN  NVNSVN  SVVF  SV  NVFSV
SVF   NVFSFVNS  SVSVNSVN  SVF  SV  FVNV  FVN  SVNVN
SVVSVN   SVF  FSVFSVNV  FVF

PERCENTAGE OF WORDS WITH FIRST SOUND AS NOTED.

NUMBER OF WORDS IN SAMPLE OF LENGTH 1 OR GREATER IS:
     221.0000
```

F	N	P	S	V
33.48415	16.28958	0	39.81900	10.40724

LANGUAGE = "A"

PERCENTAGE OF WORDS LONGER THAN ONE SOUND WITH BEGINING TWO-SOUND COMBINATIONS:

NUMBER OF WORCS IN SAMPLE OF LENGTH 3 OR GREATER IS:
204.0000

FF	FN	FP	FS	FV
0	0.9803921	0	4.901960	29.41176
NF	NN	NP	NS	NV
0	0	0	0	16.66666
PF	PN	PP	PS	PV
0	0	0	0	0
SF	SN	SP	SS	SV
0.9803921	1.960784	0	0	35.29411
VF	VN	VP	VS	VV
0	4.901960	0	0.9803921	3.921568

ANALYSIS OF SECOND SOUND AS THE PERCENTAGE OF OCCURANCE IN THE SECOND POSITION. (VECTORS = COLUMN SUM).

0.9803921	7.843136	0	5.882352	85.29408

LANGUAGE = "A"

PERCENTAGE OF WORDS LONGER THAN 2 SOUNDS WITH BEGINING 3-SOUND COMBINATIONS:

NUMBER OF WORDS IN SAMPLE OF LENGTH 4 OR GREATER IS:
126.0000

FFF	FFN	FFP	FFS	FFV
0	0	0	0	0
FNF	FNN	FNP	FNS	FNV
0	0	0	0	1.587301
FPF	FPN	FPP	FPS	FPV
0	0	0	0	0
FSF	FSN	FSP	FSS	FSV
0	2.380952	0	0	5.555554
FVF	FVN	FVP	FVS	FVV
3.968253	20.63492	0	1.587301	4.761904
NFF	NFN	NFP	NFS	NFV
0	0	0	0	0
NNF	NNN	NNP	NNS	NNV
0	0	0	0	0
NPF	NPN	NPP	NPS	NPV
0	0	0	0	0
NSF	NSN	NSP	NSS	NSV
0	0	0	0	0
NVF	NVN	NVP	NVS	NVV

7.936507	11.11111	0	0	0
PFF	PFN	PFP	PFS	PFV
0	0	0	0	0
PNF	PNN	PNP	PNS	PNV
0	0	0	0	0
PPF	PPN	PPP	PPS	PPV
0	0	0	0	0
PSF	PSN	PSP	PSS	PSV
0	0	0	0	0
PVF	PVN	PVP	PVS	PVV
0	0	0	0	0
SFF	SFN	SFP	SFS	SFV
0	0	0	0	1.587301
SNF	SNN	SNP	SNS	SNV
0	0	0	0	3.174603
SPF	SPN	SPP	SPS	SPV
0	0	0	0	0
SSF	SSN	SSP	SSS	SSV
0	0	0	0	0
SVF	SVN	SVP	SVS	SVV
3.968253	8.730155	0	3.174603	9.523808
VFF	VFN	VFP	VFS	VFV
0	0	0	0	0
VNF	VNN	VNP	VNS	VNV

1.587301	0	0	0.7936507	3.174603
VPF	VPN	VPP	VPS	VPV
0	0	0	0	0
VSF	VSN	VSP	VSS	VSV
0	0	0	0.7936507	0.7936507
VVF	VVN	VVP	VVS	VVV
0	2.380952	0	0.7936507	0

ANALYSIS OF THIRD SOUND AS TO THE PERCENTAGE OF
OCCURANCE IN THE THIRD POSITION. (VECTOR = COLUMN SUM).

17.46031	45.23807	0	7.142856	30.15869

LANGUAGE = "A"

PERCENTAGE OF WORDS LONGER THAN 3-SOUNDS WITH BEGINING 4-SOUND COMBINATIONS:

NUMBER OF WORDS IN SAMPLE OF LENGTH 5 OR GREATER IS:
97.00000

FNVV	2.061855
FSNV	3.092783
FSVF	3.092783
FSVN	2.061855
FSVS	1.030927
FSVV	1.030927
FVFN	1.030927
FVFS	4.123711
FVNN	1.030927
FVNS	14.43298
FVNV	2.061855
FVSV	2.061855
FVVN	6.185567
NVFS	6.185567
NVNS	12.37113
NVNV	1.030927
SFVN	2.061855
SVFN	1.030927
SVFS	2.061855

SVFV	2.061855
SVNS	5.154638
SVNV	4.123711
SVSS	1.030927
SVSV	3.092783
SVVN	3.092783
SVVS	5.154638
VNFV	2.061855
VNSN	1.030927
VSSV	1.030927
VVNF	1.030927
VVNV	2.061855
VVSV	1.030927

LANGUAGE = "A"

PERCENTAGE OF WORDS WITH LAST SOUND AS NOTED.

NUMBER OF WORDS IN SAMPLE OF LENGTH 1 OR GREATER IS:
221.0000

F	N	P	S	V
6.787330	23.98189	0	4.977375	8.144795

LANGUAGE = "A"

PERCENTAGE OF WORDS LONGER THAN 1-SOUND WITH ENDING 2-SOUND COMBINATIONS:

NUMBER OF WORDS IN SAMPLE OF LENGTH 3 OR GREATER IS:
204.0000

FF	FN	FP	FS	FV
0	0	0	0	3.431372
NF	NN	NP	NS	NV
0	0	0	0	24.01959
PF	PN	PP	PS	PV
0	0	0	0	0
SF	SN	SP	SS	SV
0.4901960	4.411764	0	0	0
VF	VN	VP	VS	VV
0	1.960784	0	1.960784	0

ANALYSIS OF NEXT TO LAST SOUND AS TO THE PERCENTAGE
OF OCCURANCE IN THAT POSITION.

0.4901960	6.372548	0	1.960784	27.45096

```
******************************************************************************

                                LANGUAGE = "B"

******************************************************************************

 VN  FNVNSV  NVNSV  N  VN  FVN  SVNFNVFVN  FVFNV  SNVN  FV  VNVF  VNV  VN
NVF   FNVNSV  SNVNSV  VFVNSV  V  SVFVN  VN  FNVFVNV  VNSNVNSV  VN  VNSV  SVSV
SVNSVNNVNVN   VN  SV  SNVN  VNSVF  NVFNVNV  VFNVFVN  VN  FNVFVNV  V  SVSVNFV
NV   SVSV  FVNVV  SVNFVFVNVFV  NVF  SVSVNVFV  VN  SNVNSV  NVNSV  FVSNV
SVN   SNVN  FVNNV  SVNV  SNVNSV  NVF  FVSNVFV  NVF  FV  VNNVFVFVFV
VN   FV  SVSV  VN  FNVFVNV  SVN  FVN  VN  FNVNSV  NVNSV  VSVNSVNV  NV
VNSNVFFV   VNSVNFVF  FNVNV  VN  FVN  SVN  VNFVN  V  VNNVFNVSVNVNSV
FV   FVFSVFV  FV  FV  SVSV  VN  FNVFVNV  SVN  NV  SV  N  FNVNSV  NVNSV
VFV   FV  NNVSVNVFVN  NV  FVSVNNVNVFVF  FVN  FVN
SNVNVFSVVF   SVSVSVN  SNVNVFSVVF  FVNVNV
VFVFSVSVNFNNVSVFSVNVN   SVSVSVN  FVFVNVN  FVVVNVNNVSVFSVNVN
VVFVV   NVSNVFVFVNVSVNSVFVV  VVFVVVNSVSVSVNSNVNSV
NVSVFVFSV   SVSNVVNSNVNSV  NVSVFVFSV  FVNVNVSVNSVFVV
SVNFVSVNNVFV   SVSVSVN  VFSVSNVSVVNV  SNVFVFVNV
VFSVNVFVFS   FVNVNV  VFSVNVFVFSVSVSVSVN

PERCENTAGE OF WORDS WITH FIRST SOUND AS NOTED.

NUMBER OF WORDS IN SAMPLE OF LENGTH 1 OR GREATER IS:
    119.0000

        F            N            P            S            V

    27.73108     15.96638         0        26.05042     30.25209
```

LANGUAGE = "B"

PERCENTAGE OF WORDS LONGER THAN ONE SOUND WITH BEGINING TWO-SOUND COMBINATIONS:

NUMBER OF WORDS IN SAMPLE OF LENGTH 3 OR GREATER IS:
88.00000

FF	FN	FP	FS	FV
0	10.22727	0	0	19.31818
NF	NN	NP	NS	NV
0	1.136363	0	0	13.63636
PF	PN	PP	PS	PV
0	0	0	0	0
SF	SN	SP	SS	SV
0	10.22727	0	0	22.72726
VF	VN	VP	VS	VV
7.954543	11.36363	0	1.136363	2.272726

ANALYSIS OF SECOND SOUND AS THE PERCENTAGE OF OCCURANCE IN THE SECOND POSITION. (VECTORS = COLUMN SUM).

7.954543	32.95451	0	1.136363	57.95451

LANGUAGE = "B"

PERCENTAGE OF WORDS LONGER THAN 2 SOUNDS WITH BEGINING 3-SOUND COMBINATIONS:

NUMBER OF WORDS IN SAMPLE OF LENGTH 4 OR GREATER IS:
73.00000

FFF	FFN	FFP	FFS	FFV
0	0	0	0	0
FNF	FNN	FNP	FNS	FNV
0	0	0	0	12.32876
FPF	FPN	FPP	FPS	FPV
0	0	0	0	0
FSF	FSN	FSP	FSS	FSV
0	0	0	0	0
FVF	FVN	FVP	FVS	FVV
4.109589	6.849313	0	4.109589	1.369863
NFF	NFN	NFP	NFS	NFV
0	0	0	0	0
NNF	NNN	NNP	NNS	NNV
0	0	0	0	1.369863
NPF	NPN	NPP	NPS	NPV
0	0	0	0	0
NSF	NSN	NSP	NSS	NSV
0	0	0	0	0
NVF	NVN	NVP	NVS	NVV

1.369863	5.479451	0	4.109589	0
PFF	PFN	PFP	PFS	PFV
0	0	0	0	0
PNF	PNN	PNP	PNS	PNV
0	0	0	0	0
PPF	PPN	PPP	PPS	PPV
0	0	0	0	0
PSF	PSN	PSP	PSS	PSV
0	0	0	0	0
PVF	PVN	PVP	PVS	PVV
0	0	0	0	0
SFF	SFN	SFP	SFS	SFV
0	0	0	0	0
SNF	SNN	SNP	SNS	SNV
0	0	0	0	12.32876
SPF	SPN	SPP	SPS	SPV
0	0	0	0	0
SSF	SSN	SSP	SSS	SSV
0	0	0	0	0
SVF	SVN	SVP	SVS	SVV
1.369863	6.849313	0	13.69863	0
VFF	VFN	VFP	VFS	VFV
0	1.369863	0	4.109589	2.739725
VNF	VNN	VNP	VNS	VNV

1.369863	2.739725	0	6.849313	1.369863
VPF	VPN	VPP	VPS	VPV
0	0	0	0	0
VSF	VSN	VSP	VSS	VSV
0	0	0	0	1.369863
VVF	VVN	VVP	VVS	VVV
2.739725	0	0	0	0

ANALYSIS OF THIRD SOUND AS TO THE PERCENTAGE OF OCCURANCE IN THE THIRD POSITION. (VECTOR = COLUMN SUM).

10.95890	23.28764	0	32.87669	32.87668

LANGUAGE = "B"

PERCENTAGE OF WORDS LONGER THAN 3-SOUNDS WITH BEGINING 4-SOUND COMBINATIONS:

NUMBER OF WORDS IN SAMPLE OF LENGTH 5 OR GREATER IS:
63.00000

FNVF	6.349206
FNVN	7.936507
FVFN	1.587301
FVFS	1.587301
FVFV	1.587301
FVNN	1.587301
FVNV	6.349206
FVSN	3.174603
FVSV	1.587301
FVVV	1.587301
NNVS	1.587301
NVFN	1.587301
NVNS	6.349206
NVSN	1.587301
NVSV	3.174603
SNVF	1.587301
SNVN	7.936507
SVFV	1.587301
SVNF	4.761904

SVNS	1.587301
SVSN	1.587301
SVSV	7.936507
VFNV	1.587301
VFSV	4.761904
VFVF	1.587301
VFVN	1.587301
VNFV	1.587301
VNNV	3.174603
VNSN	3.174603
VNSV	3.174603
VSVN	1.587301
VVFV	3.174603

LANGUAGE = "B"

PERCENTAGE OF WORDS WITH LAST SOUND AS NOTED.

NUMBER OF WORDS IN SAMPLE OF LENGTH 5 OR GREATER IS:
63.00000

F	N	P	S	V
7.936507	20.63492	0	1.587301	69.84126

LANGUAGE = "B"

PERCENTAGE OF WORDS LONGER THAN 1-SOUND WITH ENDING 2-SOUND COMBINATIONS:

NUMBER OF WORDS IN SAMPLE OF LENGTH 6 OR GREATER IS:

50.00000

FF	FN	FP	FS	FV
0	0	0	0	7.999998
NF	NN	NP	NS	NV
0	0	0	0	21.99998
PF	PN	PP	PS	PV
0	0	0	0	0
SF	SN	SP	SS	SV
1.999999	0	0	0	0
VF	VN	VP	VS	VV
16.00000	19.99998	0	27.99998	3.999999

ANALYSIS OF NEXT TO LAST SOUND AS TO THE PERCENTAGE
OF OCCURANCE IN THAT POSITION.

17.99998	19.99998	0	27.99998	33.99995

```
*******************************************************************************

                              LANGUAGE = "C"

*******************************************************************************

 VNSVVN   SVFVNV   SVNVSV   NVSVNV   VNSNFV   VFSVVN   NVNFVS
NVSV   NVFVNV   SVFVNV   SVNVVN   FVNSNN   SVFVSV   NVNNV
VFVSVN   SVFSFV   FVNSNV   VFVNNV   VNVNSV   VNNNNV   SVSNNV
VNVSVN   VNNVNV   VSVVSN   FVNNVN   NVNVSV   NVSVNV   NNVFVN
VFVSVN   VNFVVN   NVNFVN   VNNVVS   SVNNVN   VNFVNV   SVSVNV
VNVSVN   NVSVVN   FVFVVN   VVNFNV   VFVSFV   SVNSNV   SVSNSV
SVNVNV   NVSVVN   NVFNVS   VSVFVF   NVSFVS   FVNVSN   FVNVNV
VNNVSV   FVSVVV   SVFVF   VFSFNV   FVFVSN   SVFVVN   VNNVVN
VVFVVF   FVNVNV   VVNNSV   VNFVVN   SVNVVS   NVFNV   FVNSNV
VFVFFV   FVSVVN   VFSVNV   NNV   VNVNNV   VNVSFV   VFVFVN
FNVFS   VSVFVN   VSVNVS   VFVNNV   NNVVSV   NVSNVN   SVFVNV
FVSVNV   SNVSVN   FVSNFV   VVSVNV   VFVFNV   SNV   SNVFVF
NVFVNV   VSVFNV   FNVFS   FVFVVS   SVFVVN   VNNVSV   NVFNV
NVNVNV   VFVSSV   FNVNV   NVSVNV   FVSNNV   VSVNFV   FNVFVN
FNVFS   VSVVFV   VNNVVN   FVNFNV   VNVSNV   VFVSSN   SVSNSV
VSNVVF   SVNNVN   NVSVSV   VNFVSN   VNSVVN   VFVNFV   FVNVNV
VSNVNV   VFVSNV   NVSVNV   VSNVNV   VFVSNV   NVSVNV   SNVFSV
VVSVFV   VNVSNV   SVNFFV   VSVNNV   SVNVNV   VSVSSV   VFVSNV
NVNVNV   VNVFSN   VNFVVF   VNNVSN   VFVFVN   NVFFVN   SVNVVS
VNNVVN   SVFVNV
NV   SV   VN   VSNV   V   V   VN   NV   SV   SV   VFNV   FV   SV
FV   V   SV   V   FV   SV   FV   FV   SVN   VFNV   SNV   V   SV   NV
VSN   VNV   SVN   FVN   NV   SV   SVFNV   VFVS   NV   FVN   SVN   FVF   NV
SVN   V   V   SNV   NVV   V   SV   FV   NVN   FVNF
V   NF   FNVN   SV   NV   NVSN   VSVN   SNV   FVFNVN   SVNSV
SV   FVNVN   VFV   SVNV   SNV   SVFNVN   NVFNV   FVNNVN
SVF   VN   SV   SVSVS   FVN   VSN   SF   FVNNVN   SNVFV
NV   FVNV   NV   NVN   SNVNNV   SNVSN   NNV   FVN   NV   SV   SV
FVNVV   FV   V   NNV   SVFVN   SNNVN   VN   SV   FVN
SV   NNV   FV
NV   SVF   V   N   FVNVN   FV   SVFSVSV   FVSV   VFVNV   S   VN   VSV
N   SNV   FVN   SVS   VF   V   FV   V   FNVNVFVN   SV   F   VFVFV   VFNVSV
SV   FV   NVSV   V   FV   SVSV   SVSVN   SV   FVNVV   SV   VNVFNV   N
SNVNNV   V   FVN   VSV   FV   NVSV   V   FNVNVFVN   FVNV   NSVNSV
SVN   NV   SNV   FVN   VNVN   NV   SVF   F   V   NVF   V   FVFNV   S   SVS   FV
FVNF   NV   SNV   VN   FVFNV   SNV   N   FNVNVFVN   FVNV   FV   NVSV   VSVN   SV
NVV   V   V   NV   FV   NV   SVF   V   NNVFV   V   NV   NVV   FVN   VSV
VNVN   NV   FVNVN   V   SVNVFV   V   SNVNV   V   V   SV   S   V   NVNV   N
FNVFVFVN   NVFVFV   V   VSV   FV   NVSV   VFV   NV   SVF   V
SV   NSVNVSNV   SV   N   FVNVN   VSV   N   SNV   FVN   SV   SV
```

PERCENTAGE OF WORDS WITH FIRST SOUND AS NOTED.

NUMBER OF WORDS IN SAMPLE OF LENGTH 1 OR GREATER IS:
359.0000

F	N	P	S	V
20.61281	20.33426	0	26.18384	32.86906

LANGUAGE = "C"

PERCENTAGE OF WORDS LONGER THAN ONE SOUND WITH BEGINING TWO-SOUND COMBINATIONS:

NUMBER OF WORDS IN SAMPLE OF LENGTH 3 OR GREATER IS:
247.0000

FF	FN	FP	FS	FV
0	4.048582	0	0	18.21861
NF	NN	NP	NS	NV
0	2.834007	0	0.8097164	14.97976
PF	PN	PP	PS	PV
0	0	0	0	0
SF	SN	SP	SS	SV
0	7.692307	0	0	18.21861
VF	VN	VP	VS	VV
10.52631	11.74089	0	8.906877	2.024291

ANALYSIS OF SECOND SOUND AS THE PERCENTAGE OF
OCCURANCE IN THE SECOND POSITION. (VECTORS = COLUMN SUM).

10.52631	26.31578	0	9.716593	53.44125

LANGUAGE = "C"

PERCENTAGE OF WORDS LONGER THAN 2 SOUNDS WITH BEGINING 3-SOUND COMBINATIONS:

NUMBER OF WORDS IN SAMPLE OF LENGTH 4 OR GREATER IS:
194.0000

FFF	FFN	FFP	FFS	FFV
0	0	0	0	0
FNF	FNN	FNP	FNS	FNV
0	0	0	0	5.154638
FPF	FPN	FPP	FPS	FPV
0	0	0	0	0
FSF	FSN	FSP	FSS	FSV
0	0	0	0	0
FVF	FVN	FVP	FVS	FVV
3.092783	11.34020	0	3.092783	0
NFF	NFN	NFP	NFS	NFV
0	0	0	0	0
NNF	NNN	NNP	NNS	NNV
0	0	0	0	1.546391
NPF	NPN	NPP	NPS	NPV
0	0	0	0	0
NSF	NSN	NSP	NSS	NSV
0	0	0	0	1.030927

NVF	NVN	NVP	NVS	NVV
4.123711	3.608247	0	8.247416	0
PFF	PFN	PFP	PFS	PFV
0	0	0	0	0
PNF	PNN	PNP	PNS	PNV
0	0	0	0	0
PPF	PPN	PPP	PPS	PPV
0	0	0	0	0
PSF	PSN	PSP	PSS	PSV
0	0	0	0	0
PVF	PVN	PVP	PVS	PVV
0	0	0	0	0
SFF	SFN	SFP	SFS	SFV
0	0	0	0	0
SNF	SNN	SNP	SNS	SNV
0	0.5154635	0	0	4.123711
SPF	SPN	SPP	SPS	SPV
0	0	0	0	0
SSF	SSN	SSP	SSS	SSV
0	0	0	0	0
SVF	SVN	SVP	SVS	SVV
6.701028	6.701028	0	3.608247	0
VFF	VFN	VFP	VFS	VFV
0	1.546391	0	1.546391	9.278345

VNF	VNN	VNP	VNS	VNV
2.577319	4.639174	0	1.546391	5.670102
VPF	VPN	VPP	VPS	VPV
0	0	0	0	0
VSF	VSN	VSP	VSS	VSV
0	2.061855	0	0	5.670102
VVF	VVN	VVP	VVS	VVV
0.5154635	1.030927	0	1.030927	0

ANALYSIS OF THIRD SOUND AS TO THE PERCENTAGE OF
OCCURANCE IN THE THIRD POSITION. (VECTOR = COLUMN SUM).

17.01028	31.44324	0	19.07213	32.47418

LANGUAGE = "C"

PERCENTAGE OF WORDS LONGER THAN 3-SOUNDS WITH BEGINING 4-SOUND COMBINATIONS:

NUMBER OF WORDS IN SAMPLE OF LENGTH 5 OR GREATER IS:
170.0000

FNVF	2.941175
FNVN	2.352941
FVFN	1.764706
FVFV	1.764706
FVNF	0.5882353
FVNN	1.764706
FVNS	1.764706
FVNV	5.882352
FVSN	1.176470
FVSV	1.764706
NNVF	1.176470
NNVV	0.5882353
NSVN	1.176470
NVFF	0.5882353
NVFN	2.352941
NVFV	1.764706
NVNF	1.176470

NVNN	0.5882353
NVNV	1.764706
NVSF	0.5882353
NVSN	0.5882353
NVSV	4.705882
SNNV	0.5882353
SNVF	1.764706
SNVN	1.764706
SNVS	1.176470
SVFN	1.176470
SVFS	1.176470
SVFV	5.294117
SVNF	0.5882353
SVNN	1.176470
SVNS	1.176470
SVNV	4.117646
SVSN	1.764706
SVSV	1.764706
VFNV	0.5882353
VFSF	0.5882353
VFSV	1.176470
VFVF	2.941175
VFVN	2.352941
VFVS	4.705882

VNFV	2.941175
VNNN	0.5882353
VNNV	4.705882
VNSN	0.5882353
VNSV	1.176470
VNVF	1.176470
VNVN	1.176470
VNVS	2.941175
VSNV	1.764706
VSVF	1.764706
VSVN	1.764706
VSVS	0.5882353
VSVV	1.176470
VVFV	0.5882353
VVNF	0.5882353
VVNN	0.5882353
VVSV	1.176470

LANGUAGE = "C"

PERCENTAGE OF WORDS WITH LAST SOUND AS NOTED.

NUMBER OF WORDS IN SAMPLE OF LENGTH 1 OR GREATER IS:
359.0000

F	N	P	S	V
1.671309	15.32033	0	3.342618	27.01949

LANGUAGE = "C"

PERCENTAGE OF WORDS LONGER THAN 1-SOUND WITH ENDING 2-SOUND COMBINATIONS:

NUMBER OF WORDS IN SAMPLE OF LENGTH 3 OR GREATER IS:
247.0000

FF	FN	FP	FS	FV
0	0	0	0	2.024291
NF	NN	NP	NS	NV
0	0.4048582	0	2.834007	15.78947
PF	PN	PP	PS	PV
0	0	0	0	0
SF	SN	SP	SS	SV
0	0	0	0	3.238866
VF	VN	VP	VS	VV
5.263157	20.24290	0	6.882589	0.4048582

ANALYSIS OF NEXT TO LAST SOUND AS TO THE PERCENTAGE
OF OCCURANCE IN THAT POSITION.

5.263157	20.64775	0	9.716597	21.45746

```
*****************************************************************************

                              LANGUAGE = "D"

*****************************************************************************

 NVNNNVVN  NVNVVNVN  VN  VNFVNFV  VNVNVVSV  VVVVSF
VVNSVNNV   VVSVNVSN  VVFVFVFN  SVNFVVFV
SVSFVNVS   NVVNS  NVFVNFVS  NVNNVVFV
FVFSVVNF   VVNVNVN  VNNVNFVV  NVN  VVNVN  FVNVSV
VN   NVV  SVF  NVVSVV  VVVSV  FSVVFVVN  VVNVFVNF
VVNFVNVV   SVSNVVFS  SVFVVSVS  VVFVNNVVN  FVNFVNVS
FVNN   FVNFVVSV  NVNSVVSV  NVVSVNVV  NVVNVNNV
SVSFVNVV   NVNVVN  NVNVVNSVN  VNVVN  FVNFVNFV
NV   NVVFVNSVN  NVVSVS  VVNVFSVN  SVVSVFVS  VVV
VNVVFVNS   VVSVNVN  VVNVVSVN  SNVNVS  FVN
VVFVNSVS   VVFVFFVN  VNNVVNVV  VNVVFVFV
VNVSVVSV   NVN  VNNVS  NVS  NVSVVSVN  VVNVNVS
VNVNVNSN   VNVVV  VNVFVNNV  VVFVNSF
FVFFVNVN   VFVVSVSV  VVFVNSNV
SVFFVNFV   SVSFVNFV  NVSFSVS  NVN  FVVFVVNS
SVSVSNVS   VVFVNSVS
V   NV  FV  V  VN  SV  FVS  VS  VF  VNS  SVS  NVN  VF  VN
FV   NV  FV  FV  FVF  FVN  NV  SVNS  SV  VN  NVNS  SV  SVN
NVF   NV  FVN  VSVS  VS  FVNS  FVF  SV  FVN  FV  SVN
SVN   NVS  FVN  NVV  NV  FVVN  FNVS  NVVNVN  VN
SFFS   VFV  SV
FV   NVF  NVNS  N  FV  FVN  NF  VFVVN  NVN  SV  VSVVS
FNVSF   VF  VN  NVF  FSNVNSF  FNVN  V  SNVFVNF  SVN
VNVN   NVSS  VS  VN  VF  VFFSVS  FV  VSNVS  SFVS
FV   NVN  FV  SVS  NVS  FV  SNVFVNF  SVS  VF  SVS
VF   NVS  SV  SVNFV  SFS  FSNVNSF  FVN  FV  VFF  NVN
FVN   FV  NVFF  NVNS  SNV  VF  FVFS  VF  V  SVS  SVS  FV
FVFSF   V  SNV  FV  SVSF  FV  SNVFVNF  NVSS  VF
SVS   VNVVNS  VN  N  VS  NVFS  FV  NVFF  NVNS  SVF  VS
SNVVN   FVN  FV  FVN  SVSVN  SV  FVVN  FVS  N  NVVS
VNV   FV  SNVFVNF  SVS  VF  SVS  VF  N  FV  FV  NVFF  NVNS
FVS   SV  VNVS  FVS  FV  FVN  NVF  FSNVNSF  FVN  FV  NVF
```

PERCENTAGE OF WORDS WITH FIRST SOUND AS NOTED.

NUMBER OF WORDS IN SAMPLE OF LENGTH 1 OR GREATER IS:
241.0000

F	N	P	S	V
25.31119	23.65144	0	19.91701	31.12032

LANGUAGE = "D"

PERCENTAGE OF WORDS LONGER THAN ONE SOUND WITH BEGINING TWO-SOUND COMBINATIONS:

NUMBER OF WORDS IN SAMPLE OF LENGTH 3 OR GREATER IS:
168.0000

FF	FN	FP	FS	FV
0	1.785714	0	2.380952	18.45236
NF	NN	NP	NS	NV
0	0	0	0	27.38094
PF	PN	PP	PS	PV
0	0	0	0	0
SF	SN	SP	SS	SV
1.785714	4.761904	0	0	16.66666
VF	VN	VP	VS	VV
2.976190	10.11904	0	1.785714	11.90476

ANALYSIS OF SECOND SOUND AS THE PERCENTAGE OF
OCCURANCE IN THE SECOND POSITION. (VECTORS = COLUMN SUM).

4.761904	16.66666	0	4.166666	74.40471

LANGUAGE = "D"

PERCENTAGE OF WORDS LONGER THAN 2 SOUNDS WITH BEGINING 3-SOUND COMBINATIONS:

NUMBER OF WORDS IN SAMPLE OF LENGTH 4 OR GREATER IS:
109.0000

FFF	FFN	FFP	FFS	FFV
0	0	0	0	0
FNF	FNN	FNP	FNS	FNV
0	0	0	0	2.752293
FPF	FPN	FPP	FPS	FPV
0	0	0	0	0
FSF	FSN	FSP	FSS	FSV
0	2.752293	0	0	0.9174309
FVF	FVN	FVP	FVS	FVV
3.669724	5.504586	0	0	2.752293
NFF	NFN	NFP	NFS	NFV
0	0	0	0	0
NNF	NNN	NNP	NNS	NNV
0	0	0	0	0
NPF	NPN	NPP	NPS	NPV
0	0	0	0	0
NSF	NSN	NSP	NSS	NSV
0	0	0	0	0

NVF	NVN	NVP	NVS	NVV
4.587155	10.09174	0	3.669724	7.339447
PFF	PFN	PFP	PFS	PFV
0	0	0	0	0
PNF	PNN	PNP	PNS	PNV
0	0	0	0	0
PPF	PPN	PPP	PPS	PPV
0	0	0	0	0
PSF	PSN	PSP	PSS	PSV
0	0	0	0	0
PVF	PVN	PVP	PVS	PVV
0	0	0	0	0
SFF	SFN	SFP	SFS	SFV
0.9174309	0	0	0	0.9174309
SNF	SNN	SNP	SNS	SNV
0	0	0	0	5.504586
SPF	SPN	SPP	SPS	SPV
0	0	0	0	0
SSF	SSN	SSP	SSS	SSV
0	0	0	0	0
SVF	SVN	SVP	SVS	SVV
1.834862	2.752293	0	6.422012	0.9174309
VFF	VFN	VFP	VFS	VFV
0.9174309	0	0	0	1.834862

VNF	VNN	VNP	VNS	VNV
0.9174309	2.752293	0	0	10.09174
VPF	VPN	VPP	VPS	VPV
0	0	0	0	0
VSF	VSN	VSP	VSS	VSV
0	0.9174309	0	0	1.834862
VVF	VVN	VVP	VVS	VVV
6.422012	7.339447	0	1.834862	1.834862

ANALYSIS OF THIRD SOUND AS TO THE PERCENTAGE OF
OCCURANCE IN THE THIRD POSITION. (VECTOR = COLUMN SUM).

19.26604	32.11006	0	11.92660	36.69719

LANGUAGE = "D"

PERCENTAGE OF WORDS LONGER THAN 3-SOUNDS WITH BEGINING 4-SOUND COMBINATIONS:

NUMBER OF WORDS IN SAMPLE OF LENGTH 5 OR GREATER IS:
83.00000

FNVS	1.204819
FSNV	3.614457
FSVV	1.204819
FVFF	1.204819
FVFS	2.409638
FVNF	3.614457
FVNV	1.204819
FVVF	1.204819
NVFV	1.204819
NVNN	2.409638
NVNS	1.204819
NVNV	3.614457
NVSF	1.204819
NVSV	1.204819
NVVF	1.204819
NVVN	3.614457
NVVS	3.614457
SNVF	4.819277

SNVN	1.204819
SNVV	1.204819
SVFF	1.204819
SVFV	1.204819
SVNF	2.409638
SVSF	3.614457
SVSN	1.204819
SVSV	2.409638
SVVS	1.204819
VFFS	1.204819
VFVV	2.409638
VNFV	1.204819
VNNV	3.614457
VNVF	1.204819
VNVN	2.409638
VNVS	1.204819
VNVV	6.024096
VSNV	1.204819
VSVV	1.204819
VVFV	8.433729
VVNF	1.204819
VVNS	1.204819
VVNV	7.228910
VVSV	2.409638

VVVS	1.204819
VVVV	1.204819

LANGUAGE = "D"

PERCENTAGE OF WORDS WITH LAST SOUND AS NOTED.

NUMBER OF WORDS IN SAMPLE OF LENGTH 1 OR GREATER IS:
241.0000

F	N	P	S	V
5.394190	9.543568	0	8.713692	10.78838

LANGUAGE = "D"

PERCENTAGE OF WORDS LONGER THAN 1-SOUND WITH ENDING 2-SOUND COMBINATIONS:

NUMBER OF WORDS IN SAMPLE OF LENGTH 3 OR GREATER IS:
168.0000

FF	FN	FP	FS	FV
0	3.571428	0	2.976190	0
NF	NN	NP	NS	NV
0.5952380	0	0	1.190475	8.928567
PF	PN	PP	PS	PV
0	0	0	0	0
SF	SN	SP	SS	SV
0.5952380	1.785714	0	0	7.738090
VF	VN	VP	VS	VV
4.166666	2.380952	0	3.571428	3.571428

ANALYSIS OF NEXT TO LAST SOUND AS TO THE PERCENTAGE
OF OCCURANCE IN THAT POSITION.

5.357141	7.738094	0	7.738093	20.23808

```
**************************************************************************

                          LANGUAGE = "E"

**************************************************************************

 VV  NV  V  FSV  SV  NV  F  FV  V  FVF  F  SVS  SV  NVS
FVS   FFV  S  VNV  NV  SVS  FV  SV  NVVF  FNVS  VNV
V   VSVS  SVS  VN  NV  VVFFV  SSV  SVS  SVSVV  FSVFVS
NVV   SV  NVVFVFV  SVN  SVNSV  VFFV  SV  SVSVV  SSV  NV
VSVN   SVFVNVS  SFV  NVSV  FVVFVF
FSVSVNNV   FSVSVSSNV  FSVFVNNV  SFVNNVSN  NVFVNVNV
SVNVFNVS   SVSNVSNNV  FFNNNVNV  NVSNVFFV  NNVS
SNNVNVSV   SVFFVNNNV  SVNVNVVS  NNNVNVVS  SVSVNVNV
SFVNNVNV   VFSVSVNNV  SVSVNVNV  SVFFVNSV  FNVNFSVV
SVSVFVNV   FVSVSVNSN  SNVNSFSV  FSVNNSVS  SFSVNVNV
VSFVNSNV   NVSVSVSNV  FVSVFFSV  FSVSNVVS  VFSNNNNV
NVVSVVNV   NVSVSVFVF  VNVSNNVF  VNVFVNVF  FVSNVSV
SNVSSVNV   VFFVNNFSN  FVSVSVFV  FSVNNSV  SNNNVNVNV
SNVFVFNV   SFVFVNNSV  VNVFSFFV  FVNVFV  FSVSVSV
SNVSVSVF   VFSFVSFNV  NVFVSSVF  NVFVFVVF  NVFSVF
FVNVSNVF   NNVS  FSVFNVSV  SVNNFVVF  SVFNFNNV
NVFNSNNV   SVSVFVNV  SVSNVSNV  NNNVNVSV  FVSNVFNV
FSVSVNNV   NVSFVNNV  VSV  NVFNVNFNV  NNVSVSNFV
NVNNVFNV   FVSNVNSV  VNVFNVSV  FSNNNVSV  FVNVNVFV
NVNNSFSN   SVSVFVVS  FSNNNVNV  FNVFVNFV  SVFVNVVF
NVSVSNSV   FVSFNFSV  VSNVFSNV  VSNVNNSV  VSNNNVNV
FNVNNVFV   VSFSVSFV  FNVFVNVF  SVFFVNSV  FVFSVFNS
FVSFVNNS   SNNNVSSV  FVNVFV  SVNVSFNV  FSVSNNSV
SNNVNNSV   SVNNVSSV  SVFVSNNV  VSFVFSFV  SVSVNSNV
VSFVFVSN   FSVSVN  SFVVSFSV  SVSVNSVN  VSVNVSVS
VSSVVNNV   VSSVSNSV  FSVFVSNV  FSVNVNSN  SNNNVSNV
NNVNVNFV   FVNVFV  NVSNNVVS  VFVNVFSV  FSVSVSFV
SVFNVNNV   NVFNNVFN  SVNNFVVN
VSNVSV   FNVFNV  SSVFVV  SVSNSV  SVSNNV  VSVNNV  FSVSVN
SVNVSF   SVSFVN  NVSFVN  SNVNSF  SVSFFV  SVSSSV  SVSNSF
SVSNSN   FSVSSV
VSNVFSV   FNVFNVNNVV  FNVSNVN  V  FVNSFV  FSVNNVNNV  SSV  VF
NNVF   FNVNNNNVV  SVS  NVF  F  VSV  FNNVNNV  VNNV  FVNNVSNVNNV
FVSVSVNVFV   F  SNVFSF  SVSNNVSV  SVSFNVSVFFVFVFV  SV
SVNVSNV   V  SVNNVFVNNV  FSV  SVS  VF  NNVF  SVSNVS
FSFVSVSV   FVNVN  FNVNNNVN  SF  NV  NVNNFV  VSVFSV
FVFSVFNVSN   SVSNNVSV  FNNVSN  SNVFSF  SVS  FNVFNVNNVV
FNVSNVN   SNNVNNVNFV  SVSN  V  FV  FFNVF  FNVN  NV  SVN  FNVNNNNVV
VN   SVN  SNVN  FNVNNNNVV  SVSVNFV  SVSNNVS  F  FFVV  SNVFSF  SVS
FSV   F  SVNSV  SVNSVF  FNVFNVNNVV  FNVSNVN  SVNFVN  SVN
VSSVFVSV   VS  FFVNVV  FVSVSV  SVSSV  FVFNVNVNV  FVNNVFSV
SVSNNVS   SVNNVNNVSV  VSVSNNVNFV  V  FFSVNNV  FNNVN  FFVV
```

```
SNVFSF   SVSNVN  VSNVFVN  FNVFNVNNVV  FNVSNVN  FVNVFVNVN
SVN   SNNVFNVS  FSV  FVNSV  FNVNNNNVV  NVFV
VSNVFSV   FVVVNNV  VVSVN  V  FVNFV  FSVNVNV  SSV
VF   NVF  FVNNVV  SVS  NVF  F  VSV  FNVNV
VNV   FV  NVNVNV  FV  SVSVNVFV  F  SNVFSF
SVSNVSV   VSS
```

PERCENTAGE OF WORDS WITH FIRST SOUND AS NOTED.

NUMBER OF WORDS IN SAMPLE OF LENGTH 1 OR GREATER IS:
301.0000

F	N	P	S	V
33.22258	14.28571	0	34.21925	18.27242

LANGUAGE = "E"

PERCENTAGE OF WORDS LONGER THAN ONE SOUND WITH BEGINING TWO-SOUND COMBINATIONS:

NUMBER OF WORDS IN SAMPLE OF LENGTH 3 OR GREATER IS:
258.0000

FF	FN	FP	FS	FV
2.713178	9.302323	0	10.07752	11.62791
NF	NN	NP	NS	NV
0	3.100775	0	0	10.85271
PF	PN	PP	PS	PV
0	0	0	0	0
SF	SN	SP	SS	SV
2.325581	6.976741	0	1.937984	25.58138
VF	VN	VP	VS	VV
2.325581	3.100775	0	9.302323	0.7751938

ANALYSIS OF SECOND SOUND AS THE PERCENTAGE OF
OCCURANCE IN THE SECOND POSITION. (VECTORS = COLUMN SUM).

7.364339	22.48061	0	21.31781	48.83717

LANGUAGE = "E"

PERCENTAGE OF WORDS LONGER THAN 2 SOUNDS WITH BEGINING 3-SOUND COMBINATIONS:

NUMBER OF WORDS IN SAMPLE OF LENGTH 4 OR GREATER IS:
220.0000

FFF	FFN	FFP	FFS	FFV
0	0.9090908	0	0.4545454	1.363636
FNF	FNN	FNP	FNS	FNV
0	1.363636	0	0	9.545451
FPF	FPN	FPP	FPS	FPV
0	0	0	0	0
FSF	FSN	FSP	FSS	FSV
0.4545454	0.9090908	0	0	8.636361
FVF	FVN	FVP	FVS	FVV
1.363636	5.909090	0	4.545454	0.9090908
NFF	NFN	NFP	NFS	NFV
0	0	0	0	0
NNF	NNN	NNP	NNS	NNV
0	0.9090908	0	0	2.727272
NPF	NPN	NPP	NPS	NPV
0	0	0	0	0
NSF	NSN	NSP	NSS	NSV
0	0	0	0	0

NVF	NVN	NVP	NVS	NVV
3.636363	1.818181	0	3.636363	1.363636
PFF	PFN	PFP	PFS	PFV
0	0	0	0	0
PNF	PNN	PNP	PNS	PNV
0	0	0	0	0
PPF	PPN	PPP	PPS	PPV
0	0	0	0	0
PSF	PSN	PSP	PSS	PSV
0	0	0	0	0
PVF	PVN	PVP	PVS	PVV
0	0	0	0	0
SFF	SFN	SFP	SFS	SFV
0	0	0	0.4545454	1.818181
SNF	SNN	SNP	SNS	SNV
0	3.181818	0	0	4.999999
SPF	SPN	SPP	SPS	SPV
0	0	0	0	0
SSF	SSN	SSP	SSS	SSV
0	0	0	0	0.4545454
SVF	SVN	SVP	SVS	SVV
3.636363	6.363636	0	13.63636	0
VFF	VFN	VFP	VFS	VFV
0.9090908	0	0	1.363636	0.4545454

VNF	VNN	VNP	VNS	VNV
0	0.4545454	0	0	1.818181
VPF	VPN	VPP	VPS	VPV
0	0	0	0	0
VSF	VSN	VSP	VSS	VSV
1.818181	3.181818	0	1.363636	2.727272
VVF	VVN	VVP	VVS	VVV
0.4545454	0	0	0.4545454	0

ANALYSIS OF THIRD SOUND AS TO THE PERCENTAGE OF
OCCURANCE IN THE THIRD POSITION. (VECTOR = COLUMN SUM).

12.27272	24.99997	0	25.90907	36.81812

LANGUAGE = "E"

PERCENTAGE OF WORDS LONGER THAN 3-SOUNDS WITH BEGINING 4-SOUND COMBINATIONS:

NUMBER OF WORDS IN SAMPLE OF LENGTH 5 OR GREATER IS:
203.0000

FFNN	0.4926108
FFNV	0.4926108
FFSV	0.4926108
FFVN	0.4926108
FNNV	1.477832
FNVF	3.448276
FNVN	3.940886
FNVS	1.970443
FSFV	0.4926108
FSNN	0.9852216
FSVF	1.970443
FSVN	2.463054
FSVS	4.926107
FVFN	0.4926108
FVFS	0.9852216
FVNF	0.4926108
FVNN	1.477832
FVNS	0.9852216

FVNV	3.448276
FVSF	0.9852216
FVSN	1.477832
FVSV	2.463054
FVVF	0.4926108
FVVV	0.4926108
NNNV	0.9852216
NNVN	0.4926108
NNVS	0.4926108
NVFN	1.477832
NVFS	0.4926108
NVFV	1.477832
NVNN	1.477832
NVNV	0.4926108
NVSF	0.9852216
NVSN	0.9852216
NVSV	1.477832
NVVF	0.4926108
NVVS	0.4926108
SFSV	0.4926108
SFVF	0.4926108
SFVN	0.9852216
SFVV	0.4926108
SNNN	1.477832

SNNV	1.970443
SNVF	2.955665
SNVN	0.9852216
SNVS	0.9852216
SSVF	0.4926108
SVFF	1.477832
SVFN	0.9852216
SVFV	1.477832
SVNF	0.4926108
SVNN	2.463054
SVNS	1.477832
SVNV	2.463054
SVSF	1.477832
SVSN	6.403934
SVSS	0.9852216
SVSV	5.418718
VFFV	0.4926108
VFSF	0.4926108
VFSN	0.4926108
VFSV	0.4926108
VFVN	0.4926108
VNVF	1.477832
VNVS	0.4926108
VSFS	0.4926108

VSFV	1.477832
VSNN	0.4926108
VSNV	2.955665
VSSV	1.477832
VSVF	0.4926108
VSVN	0.9852216
VSVS	0.4926108
VVFF	0.4926108
VVSV	0.4926108

LANGUAGE = "E"

PERCENTAGE OF WORDS WITH LAST SOUND AS NOTED.

NUMBER OF WORDS IN SAMPLE OF LENGTH 1 OR GREATER IS:
301.0000

F	N	P	S	V
7.308966	9.302323	0	5.315614	45.51494

LANGUAGE = "E"

PERCENTAGE OF WORDS LONGER THAN 1-SOUND WITH ENDING 2-SOUND COMBINATIONS:

NUMBER OF WORDS IN SAMPLE OF LENGTH 3 OR GREATER IS:
258.0000

FF	FN	FP	FS	FV
0	0	0	3.100775	5.038759
NF	NN	NP	NS	NV
0.3875968	0	0	3.488372	5.813952
PF	PN	PP	PS	PV
0	0	0	0	0
SF	SN	SP	SS	SV
0	0.7751938	0	0	5.426356
VF	VN	VP	VS	VV
9.302323	20.54263	0	15.11627	4.651162

ANALYSIS OF NEXT TO LAST SOUND AS TO THE PERCENTAGE
OF OCCURANCE IN THAT POSITION.

9.689919	21.31783	0	21.70541	20.93022

PROBABILITY MATRIX FOR FIRST SOUNDS:

F	N	P	S	V
0.2366426	0.1788966	0.2000000	0.2768083	0.08420306
0.2040045	0.1811507	0.2000000	0.1648332	0.2502315
0.1456769	0.2233164	0.2000000	0.1820213	0.2659377
0.1788818	0.2597466	0.2000000	0.1384563	0.2517889
0.2347940	0.1568896	0.2000000	0.2378808	0.1478388

017.12 SECONDS IN EXECUTION

```
********************************************************************************

          C O M P U T E R   O U T P U T   NUMBER 2

********************************************************************************
```

THIS OUTPUT WAS THE RESULT OF TWO TEST SAMPLES FED INTO COMPUTER PROGRAM NUMBER TWO. THERE WERE TWO SAMPLE SETS, THE FIRST CONSISTING OF EIGHT WORDS FROM LANGUAGE "E," AND THE SECOND SEVEN WORDS FROM LANGUAGE TYPE "A." THE RESULTS CLEARLY INDICATE THE CORRECT RESPONSE IN BOTH CASES. THE WORDS USED IN THE TEST SAMPLES WERE SELECTED ENTIRELY AT RANDOM.

```
SAMPLE NUMBER  1

SAMPLE WOPD NUMBER  1  IN CODE FORM IS:V
VALIDITY FACTORS :
            0       0.2571289       0.3031194       0.2380823       0.2016697

SAMPLE WORD NUMBER  2  IN CODE FORM IS:SV
VALIDITY FACTORS :
    0.4042035       0.05285739      0.2613090       0.09662992      0.1850008
     0.6037687       0.01032479      0.2429902       0.03450592      0.1084104

SAMPLE WORD NUMBER  3  IN CODE FORM IS:VSV
VALIDITY FACTORS :
    0.1318377       0.1958732       0.2771791       0.1244956       0.2706143
            0               0        0.5623930              0        0.4376070
            0               0        0.6056350              0        0.3943655

SAMPLE WORD NUMBER  4  IN CODE FORM IS:SVS
VALIDITY FACTORS :
    0.2564532       0.1333556       0.2075819       0.1525679       0.2500418
     0.3812183       0.1030814       0.1362370       0.1124352       0.2670283
     0.1860361              0       0.04892233       0.1632116       0.6018304

SAMPLE WORD NUMBER  5  IN CODE FORM IS:NNVS
VALIDITY FACTORS :
    0.1289187               0       0.2180875       0.3451055       0.3078883
            0               0               0               0        1.000000
            0               0               0               0        1.000000
            0               0               0               0        1.000000
```

SAMPLE WORD NUMBER 6 IN CODE FORM IS:SVNSV

VALIDITY FACTORS :

0.2987307	0.05285237	0.2369244	0.1472315	0.2642618
0.4249547	0.01330184	0.1603813	0.06881660	0.3325458
0.6427862	0	0.03206684	0.02850135	0.2966461
0.5445783	0	0.01077332	0	0.4446487
0	0	0.005401272	0	0.9945992

SAMPLE WORD NUMBER 7 IN CODE FORM IS:FSVSVN

VALIDITY FACTORS :

0.3094826	0.2155455	0.1194290	0.1073822	0.2481610
0.5446888	0	0	0.08107507	0.3742368
0.5717126	0	0	0.01303771	0.4152498
0.2611750	0	0	0	0.7388250
0.7346960	0.05825441	0.08012134	0.02103019	0.1058987
0.6785653	0.07490450	0.1426878	0.08776635	0.01607632
0	0	0	0	1.000000
0.8962153	0	0	0.02384420	0.07994050
0	0	0	0	1.000000

SAMPLE WORD NUMBER 8 IN CODE FORM IS:FSVNNSVS

VALIDITY FACTORS :

0.3094826	0.2155455	0.1194290	0.1073822	0.2481610
0.5446888	0	0	0.08107507	0.3742368
0.5717126	0	0	0.01303771	0.4152498
0.5857502	0	0	0	0.4142498
0	0	0.08074415	0.2967113	0.6225446
0	0	0.05764659	0.8312389	0.1111149
0	0	0	0	1.000000
0	0	0	0	1.000000
0	0	0	0	1.000000

****NOTE*** THE VALIDITY FACTOR PRODUCTS ARE THE RESULTANT PRODUCT FOR EACH WORD MEASURED ABOVE.

VALIDITY FACTOR PRODUCTS:

0	0.2571289	0.3031194	0.2380823	0.2016697
0.2440454	0.0005457415	0.06349552	0.003334304	0.02005602
0	0	0.09440851	0	0.04670182
0.01818775	0	0.001383539	0.002799731	0.04018315
0	0	0	0	0.3078883
0	0	7.090307'-08	0	0.01152894
0	0	0	0	3.877704'-06
0	0	0	0	0.001105081

LANGUAGE	PROD	MEAN
A	0	0.03277914
B	0	0.03220933
C	0	0.05780087
D	0	0.03052703
E	1.154554'-16	0.07864207

SAMPLE NUMBER 2

SAMPLE WORD NUMBER 1 IN CODE FORM IS:FV

VALIDITY FACTORS :

0.09987044	0.2285496	0.2032574	0.3050730	0.1632497
0.04467834	0.2339829	0.1850614	0.4168988	0.1193790

SAMPLE WORD NUMBER 2 IN CODE FORM IS:SVF

VALIDITY FACTORS :

0.2564532	0.1333556	0.2075819	0.1525679	0.2500418
0.3812183	0.1030814	0.1362370	0.1124352	0.2670283
0.8243367	0	0.1354860	0.04017778	0

SAMPLE WORD NUMBER 3 IN CODE FORM IS:VNVN

VALIDITY FACTORS :

0.1632049	0.1893177	0.3155295	0.1092217	0.2227267
0.2081042	0.3500314	0.2430773	0.07766962	0.1211181
0.3191531	0.3891916	0.2252265	0.06642997	0
0.6736734	0	0.2872279	0.03910022	0

SAMPLE WORD NUMBER 4 IN CODE FORM IS:SNVF

VALIDITY FACTORS :

0.2299383	0.5334570	0.05556843	0.1025879	0.07844955
0.1615117	0.8149884	0	0	0.02350023
0.08307642	0.9117687	0	0	0.005155075
1.000000	0	0	0	0

SAMPLE WORD NUMBER 5 IN CODE FORM IS:SVNSF

VALIDITY FACTORS :

0.2987307	0.05285237	0.2369244	0.1472315	0.2642618
0.4249547	0.01330184	0.1603813	0.06881660	0.3325458
0.6427862	0	0.03206684	0.02850135	0.2966461
0.5445783	0	0.01077332	0	0.4446487
1.000000	0	0	0	0

SAMPLE WORD NUMBER 6 IN CODE FORM IS:FVNSVNVN

VALIDITY FACTORS :

0.3094826	0.2155455	0.1194290	0.1073822	0.2481610
0.4920694	0.1844413	0.09059823	0.05826135	0.1746301
0.7221408	0.1001511	0.06396449	0.03056593	0.08317858
0.9799320	0	0.01518463	0	0.004884511
0.7346960	0.05825441	0.08012134	0.02103019	0.1058987
0.6785653	0.07490450	0.1426878	0.08776635	0.01607632
1.000000	0	0	0	0
0.9020541	0.05560115	0.02905477	0.006363511	0.006926771
0.9986470	0	0.001353574	0	0

SAMPLE WORD NUMBER 7 IN CODE FORM IS:FVN

VALIDITY FACTORS :

0.2071953	0.2565276	0.1597247	0.2347879	0.1417652
0.2200228	0.3372685	0.1307530	0.2825265	0.02942924
0.1585909	0.4862019	0.1066935	0.2485138	0

VALIDITY FACTOR PRODUCTS:

0.004462045	0.05347669	0.03761509	0.1271846	0.01948858
0.08059090	0	0.003831588	0.0006892094	0
0.007302348	0	0.004961699	2.203448'-05	0
0.003085270	0	0	0	0
0.04443743	0	0	0	0
0.04839757	0	0	0	0
0.007229794	0.04206553	0.002228240	0.01648486	0

LANGUAGE	PROD	MEAN
A	1.259718'-13	0.02792931
B	0	0.01364888
C	0	0.006948087
D	0	0.02062580
E	0	0.002784083

****

C O M P U T E R   P R O G R A M   NUMBER 1

****

```
//COOK1 JOB (0405,1404,MA52),'SMC 2737',TIME=(00,40)
// EXEC ALGOLW
//SYSPRINT DD SYSOUT=A,DCB=BLKSIZE=665,SPACE=(CYL,(3,1))
//ALGOL.SYSIN DD *
%ALGOL T=035,P=80

BEGIN
COMMENT INPUT DATA WRITTEN IN "ROMI" WITH AT LEAST ONE BLANK
BETWEEN WORDS.  WORD MUST START AND END ON THE SAME CARD.  ;
STRING (2) ARRAY ROMI (1::101,1::2);
STRING(80) TEXT;
INTEGER INDEX ,POINTER,Z,TP.WP,N;
REAL TWOCOUNT,THREECOUNT,FOURCOUNT,Q;
STRING(36) WORD,CODE;
STRING (1) ARRAY CLASS(1::5);
REAL ARRAY SOUNDONE(1::5);
STRING (2) ARRAY SOUNDTWO(1::25);
REAL ARRAY SOUNDTWOSTAT(1::25);
REAL ARRAY SOUNDTHREESTAT(1::125);
STRING(3) ARRAY SOUNDTHREE(1::125);
REAL ARRAY SOUNDFOURSTAT (1::625);
STRING(4) ARRAY SOUNDFOUR(1::625);
REAL ARRAY VECTWO (1::5);
REAL ARRAY VECTHREE (1::5);
INTEGER EPOINT;
INTEGER HEADERCOUNT;
STRING(80) ARRAY HEADER (1::10);
REAL ARRAY ENDTWOSTAT (1::25);
REAL ARRAY ENDONE (1::5);
REAL ARRAY ENDVEC2 (1::5);
LOGICAL FOUND,BAD;
REAL ARRAY STATMUX1 (1::5,1::5);
REAL ARRAY PMUX1 (1::5,1::5);
REAL ARRAY VECFOUR (1::5);
REAL B;
REAL ECOUNT, E_1COUNT;
```

```
COMMENT  ************************************************
PROCEDURE INITIALIZE RESETS THE VALUES OF THE GLOBAL
COUNTERS USED THROUGHOUT THE PROGRAM.  INITIALIZE ALSO READS
AND PRINTS THE DATA SET IDENTIFIER "HEADER," SETTING
"TEXT POINTER," "TP," TO 80 CAUSES THE PROCEDURE GETWORD
TO READ THE FIRST DATA CARD.
*************************************************************;

COMMENT**********************
HEADERCOUNT IS THE NUMBER OF CARDS PRECEEDING THE DATA
WHICH DESCRIBE THE TYPE/QUALITY/ETC. OF THE DATA BUT ARE
NOT TO BE READ AS PART OF THE DATA.
*****************************;

PROCEDURE INITIALIZE;
BEGIN
Q := 0;
TWOCOUNT := 0;
THREECOUNT :=0;
FOURCOUNT := 0;
E_1COUNT := 0:
ECOUNT := 0;
INDEX:=0;
READ (HEADERCOUNT) ;
FOR I := 1 UNTIL HEADERCOUNT DO
BEGIN
HEADER (I) := " ";
READCARD (HEADER(I));
WRITE (HEADER(I));
END;
IOCONTROL (2);
WRITE(" ");
IOCONTROL (2);
TP := 80
END INITIALIZE;
```

```
COMMENT  ***************************************************
PROCEDURE GENERATETWO PROVIDES THE 25 TWO-SOUND CLASS LABEL
DESIGNATIONS IN ALPHABETICAL ORDER.  THESE LABELS ARE
STORED IN THE ONE DIMENSION ARRAY "SOUNDTWO."  THE LABELS
CORRESPOND TO THE 25 ROWS OF THE REAL ARRAY "SOUNDTWOSTAT."
************************************************************;

PROCEDURE GENERATETWO;
BEGIN
INTEGER Y;
Y := 1;
FOR I := 1 UNTIL 5 DO
BEGIN
SOUNDONE(I) := 0;
FOR J := 1 UNTIL 5 DO
BEGIN
SOUNDTWO (Y)(0|1) := CLASS (I);
SOUNDTWO (Y)(1|1) := CLASS (J) ;
SOUNDTWOSTAT(Y) := 0;
Y := Y + 1
END;
END;
END GENERATETWO;


COMMENT  ***************************************************
PROCEDURE GENERATETHREE TAKES THE 25 ELEMENT ARRAY
"SOUNDTWO" AND GENERATES THE 125 POSSIBLE COMBINATIONS OF
THE FIRST THREE SOUNDS OF A WORD IN ALPHABETICAL ORDER.
THE 125 ELEMENTS ARE STORED IN THE ONE DIMENSIONAL ARRAY
"SOUNDTHREE" AND ARE USED AS LABELS FOR THE 125 ROWS OF THE
REAL ARRAY "SOUNDTHREESTAT."
************************************************************;

PROCEDURE GENERATETHREE;
BEGIN
INTEGER Y;
Y := 1;
FOR I := 1 UNTIL 25 DO
BEGIN
FOR J := 1 UNTIL 5 DO
BEGIN
SOUNDTHREE(Y)(0|2) := SOUNDTWO(I);
SOUNDTHREE(Y)(2|1) := CLASS(J);
SOUNDTHREESTAT(Y) := 0;
Y := Y+1
END;
END;
END GENERATETHREE;


PROCEDURE GENERATEFOUR;
BEGIN
INTEGER Y;
Y := 1;
FOR I := 1 UNTIL 125 DO
BEGIN
FOR J := 1 UNTIL 5 DO
BEGIN
SOUNDFOUR(Y)(0|3) :=SOUNDTHREE(I);
SOUNDFOUR(Y)(3|1) := CLASS(J);
SOUNDFOURSTAT(Y) := 0;
Y := Y+1
END;
END;
END GENERATEFOUR;
```

```
COMMENT  *****************************************************
PROCEDURE GETARRAY READS IN THE 101 CHARACTERS USED BY THIS
PROGRAM TO REPRESENT THE INTERNATIONAL PHONETIC ALPHABET.
EACH SOUND REPRESENTATION IS PLACED IN A TWO COLUMN ARRAY
"ROMI."   COLUMN ONE IS THE SOUND REPRESENTATION AND COLUMN
TWO CONTAINS THE CLASS: EITHER NASAL, FRICATIVE, STOP, OR
VOWEL.   THE NUMBER OF CARDS USED TO READ IN "ROMI" CANNOT
BE VARIED WITHOUT ALTERING THE PROGRAM.   THE NUMBER OF
ELEMENTS IN "ROMI" IS NOT VARIABLE.   "ROMI" IS PRINTED OUT
A CARD AT A TIME, JUST AS IT IS TYPED IN THE DATA CARDS.
*****************************************************;

PROCEDURE GETARRAY;
BEGIN
STRING(81) PHONOL;
INTEGER B,A;
B:=1;
A := 1;
WRITE ( "'ROMI' IS THE PHONETIC CHARACTER REPRESENTATION
USED IN THE IMPLEMENTATION OF THIS PROGRAM");
WRITE (" ");
IOCONTROL (2);
WHILE (A <= 27) DO
BEGIN
PHONOL := " ";
READCARD (PHONOL);
WRITE(PHONOL);
TP:=0;
WHILE(TP <= 79)DO
BEGIN
WHILE (PHONOL (TP|1) =" ") AND (TP<=79) DO
TP:= TP+1;
IF ( TP ¬= 80 ) THEN
BEGIN
IF (PHONOL (TP+1|1)¬="-") THEN
BEGIN
ROMI (B,1): = PHONOL (TP|1);
TP: = TP+2;
END
ELSE
BEGIN
ROMI (B,1)(0|1): = PHONOL (TP|1);
ROMI (B,1)(1|1): = PHONOL (TP+2|1);
TP:=TP+4;
END;
ROMI (B,2): = PHONOL (TP|1);
B: = B+1;
TP := TP+1
END
ELSE
A := A+1;
END
END;
TP := 80
END GETARRAY;
```

```
COMMENT  **************************************************
LOGICAL PROCEDURE GETWORD READS THE INPUT DATA WRITTEN ON
"ROMI" INTO A BUFFER CALLED "TEXT," ONE CARD AT A TIME AND
ECHO PRINTS EACH CARD WITH A SEQUENTIAL NUMBER IDENTIFIER
"INDEX."   THE BUFFER, "TEXT," IS SCANNED ONE CHARACTER AT
A TIME UNTIL A "WORD" IS BUILT.  LEADING AND TRAILING
BLANKS DELIMIT A "WORD."   "WORD" IS THE GLOBAL PARAMETER
OF STRING TYPE OF MAXIMUM LENGTH OF 36 CHARACTERS WHICH
IS DELIVERED TO THE MAIN PROGRAM AND SUBSEQUENT PROCEDURES.
WHEN THE CHARACTER "%" IS ENCOUNTERED ON A DATA CARD IN
OTHER THAN CARD COLUMN 1 THE LOGICAL VALUE OF "GETWORD" IS
SET TO FALSE THUS SIGNALLING THE END OF A DATA SET.
"WORDS" MUST START AND END ON THE SAME CARD.
**************************************************;

LOGICAL PROCEDURE GETWORD (STRING(36) RESULT WORD);
BEGIN
WP  := 0;
IF(TP=80) THEN
BEGIN
INDEX:=INDEX+1;
 READCARD (TEXT);
WRITE(INDEX,"    ",TEXT);
TP:=0
END;
WHILE(TEXT(TP|1)=" ")  DO
BEGIN
TP:=TP+1;
IF (TP=80) THEN
BEGIN
 READCARD (TEXT);
INDEX:=INDEX+1;
WRITE (INDEX,"    ", TEXT);
TP:=0
END
END;
WORD:=" ";
WHILE (TP<80) AND (TEXT(TP|1)¬=" ") DO
BEGIN
WORD (WP|1):= TEXT(TP|1);
WP  := WP + 1;
TP  := TP + 1
END;
WP  := WP-1;
    WORD¬="%"
END GETWORD;
```

```
COMMENT ***********************************************************
PROCEDURE "CONVERT" OPERATES ON THE GLOBAL VARIABLE "WORD"
OBTAINED FROM "GETWORD."   "WORD" IS SCANNED CHARACTER BY
CHARACTER TO OBTAIN EACH "ROMI" SOUND REPRESENTATION.
A BINARY SEARCH OF THE ARRAY "ROMI" IS CONDUCTED TO
OBTAIN THE SOUND CLASS EQUIVALENT FOR EACH "SOUND" OF
"WORD."   A NEW STRING, "CODE," IS GENERATED FROM THE FOUR
SOUND DESIGNATIONS CORRESPONDING TO THE INTERNATIONAL
PHONETIC ALPHABET SPELLING OF "WORD."   "CONVERT" IS ALSO
CAPABLE OF EDITING INPUT DATA AND PRINTS OUT A WARNING
LABEL ALONG WITH THE PARTICULAR "SOUND" REPRESENTATION
WHICH DID NOT APPEAR IN "ROMI" AND THE "WORD" IN WHICH THE
ERROR OCCURRED.   A SAMPLE OUTPUT FROM "CONVERT" WOULD BE
"FVNSFVNN."  THE COUNTER "Z" IS ALSO A GLOBAL VARIABLE,
AND TELLS THE NUMBER OF IPA SOUNDS IN EACH STRING "CODE."
***********************************************************;

PROCEDURE CONVERT (STRING(36) RESULT CODE);
BEGIN
STRING(2) SOUND;
INTEGER I,TEST;
INTEGER LISTEND, LISTBEGIN, POINTER;
CODE := " ";
BAD := TRUE;
I := 0;
Z := 0;
WHILE I<=WP DO
BEGIN
TEST := 0;
FOUND := TRUE;
SOUND:=" ";
IF WORD (I+1|1)¬= "-" THEN
BEGIN
SOUND       : = WORD (I|1);
I:=I+1; END
ELSE
BEGIN
SOUND(0|1):= WORD(I|1);
SOUND(1|1):= WORD(I+2|1);
I:=I+3
END;
 LISTBEGIN:= 1;
LISTEND  := 101;
POINTER := LISTEND;
WHILE ( FOUND ) DO
BEGIN
TEST := TEST+1;
IF SOUND < ROMI(POINTER,1) THEN
 LISTEND:= POINTER;
IF SOUND > ROMI(POINTER,1) THEN
 LISTBEGIN:= POINTER;
IF ( SOUND = ROMI ( POINTER,1 ))THEN
FOUND := FALSE
ELSE
BEGIN
IF (TEST = 12 ) THEN
BEGIN
WRITE("ERONEOUS DATA","WORD = ",WORD, "SOUND = ",SOUND);
FOUND := FALSE;
I := WP+1;
BAD := FALSE
END;
POINTER := TRUNCATE ( ( LISTEND + LISTBEGIN ) / 2 )
END;
END;
CODE ( Z|1 ) := ROMI ( POINTER, 2 ) (0|1 );
Z := Z+1      ;
IF (BAD = FALSE) THEN
Z := 0;
END
END CONVERT;
```

```
COMMENT   ***************************************************
PROCEDURE "ONESOUND" LOOKS AT THE FIRST SOUND OF A STRING
OF "CODE" AND KEEPS A RUNNING TALLY OF THE TOTAL NUMBER
OF OCCURANCES  OF EACH SOUND CLASS DESIGNATION IN POSITION
ONE OF EACH STRING OF "CODE."   A COUNTER "Q" TALLIES THE
TOTAL NUMBER OF WORDS IN EACH DATA SET (SAMPLE).   THE DATA
OBTAINED IS STORED IN THE REAL ARRAY "SOUNDONE."   A BINARY
SEARCH IS USED TO MATCH THE FIRST SOUND OF A WORD WITH THE
FIVE POSSIBILITIES.  THE LIST "POINTER" USED IN THE BINARY
SEARCH IS A GLOBAL VARIABLE AND IS USED AS A STARTING POINT
FOR THE SEARCH OF THE LIST OF ALL POSSIBLE LEADING TWO
SOUND COMBINATIONS.
*****************************************************************;

PROCEDURE ONESOUND;
BEGIN
INTEGER LISTBEGIN, LISTEND;
Z := Z - 1;
IF (Z >= 0) AND (BAD = TRUE) THEN
BEGIN
Q := Q + 1;
LISTBEGIN := 1;
LISTEND := 5;
POINTER :=   LISTEND;
WHILE (CODE(0|1) ¬= CLASS (POINTER)) DO
BEGIN
IF (CODE(0|1) < CLASS (POINTER))  THEN
LISTEND := POINTER;
IF (CODE (0|1) > CLASS (POINTER)) THEN
LISTBEGIN := POINTER;
POINTER := (LISTBEGIN + LISTEND) DIV 2
END;
SOUNDONE (POINTER) := SOUNDONE (POINTER) + 1
END
END ONESOUND;
```

```
COMMENT  *************************************************
PROCEDURE "TWOSOUND" DOES AN ANALOGOUS OPERATION ON THE
FIRST TWO SOUNDS OF A WORD AS "ONESOUND" PERFORMED.   BY
REMEMBERING THE VALUE OF THE LIST "POINTER" FROM "ONESOUND"
WE NEED ONLY SEARCH THE APPROPRIATE PORTION OF THE ARRAY
"SOUNDTWO" (FIVE ELEMENTS) FOR THE SECOND SOUND.   THE
NUMBER OF OCCURANCES OF EACH COMBINATION IS STORED IN THE
REAL ARRAY "SOUNDTWOSTAT."   "TWO COUNT" TALLIES THE NUMBER
OF WORDS FROM THE DATA SET WITH TWO OR MORE SOUNDS.
*************************************************************;

PROCEDURE TWOSOUND;
BEGIN
INTEGER LISTBEGIN, LISTEND;
IF (Z>= 2) THEN
BEGIN
TWOCOUNT := TWOCOUNT + 1;
LISTBEGIN := POINTER*5-4 ;
LISTEND := LISTBEGIN + 4;
POINTER := LISTEND;
WHILE (CODE(0|2) ¬= SOUNDTWO(POINTER)) DO
BEGIN
IF (CODE (0|2) < SOUNDTWO (POINTER))  THEN
LISTEND := POINTER;
IF (CODE(0|2) > SOUNDTWO (POINTER))  THEN
LISTBEGIN := POINTER;
POINTER := (LISTBEGIN + LISTEND)  DIV 2
END ;
SOUNDTWOSTAT (POINTER) := SOUNDTWOSTAT (POINTER) + 1
END;
END TWOSOUND;
```

```
COMMENT  ************************************************************
        THE OPERATION OF PROCEDURES THREESOUND AND FOUR-
SOUND IS IDENTICAL TO PROCEDURE TWOSOUND.
REMEMBERING THE VALUE OF THE LISTPOINTER REQUIRES A SEARCH
OF ANLY FIVE ELEMENTS FOR EACH SUBSEQUENT SOUND OF A STRING
"CODE."   THE CORRESPONDING ARRAYS ARE "SOUNDTHREESTAT"
AND "SOUNDFOURSTAT."   COUNTERS FOR EACH CATEGORY OF
LENGTH ARE "THREECOUNT" AND "FOURCOUNT."
*************************************************************;

PROCEDURE THREESOUND;
BEGIN
INTEGER LISTBEGIN, LISTEND;
IF ( Z>= 3 ) THEN
BEGIN
THREECOUNT := THREECOUNT + 1 ;
LISTBEGIN := POINTER*5-4;
LISTEND := LISTBEGIN + 4 ;
POINTER := LISTEND;
WHILE (CODE(0|3) ¬= SOUNDTHREE(POINTER)) DO
BEGIN
IF (CODE(0|3) < SOUNDTHREE(POINTER)) THEN
LISTEND := POINTER;
IF ( CODE(0|3) > SOUNDTHREE(POINTER)) THEN
LISTBEGIN := POINTER;
POINTER := (LISTBEGIN+LISTEND)DIV 2
END;
SOUNDTHREESTAT(POINTER) := SOUNDTHREESTAT(POINTER)+1
END;
END THREESOUND;


PROCEDURE FOURSOUND;
BEGIN
INTEGER LISTEND,LISTBEGIN;
IF ( Z >= 4) THEN
BEGIN
FOURCOUNT := FOURCOUNT + 1;
LISTBEGIN := POINTER * 5 - 4;
LISTEND := LISTBEGIN + 4;
POINTER := LISTEND;
WHILE (CODE (0|4) ¬= SOUNDFOUR (POINTER)) DO
BEGIN
IF (CODE (0|4) < SOUNDFOUR (POINTER )) THEN
LISTEND := POINTER;
IF (CODE (0|4) > SOUNDFOUR (POINTER)) THEN
LISTBEGIN := POINTER;
POINTER := (LISTBEGIN + LISTEND) DIV 2
END;
SOUNDFOURSTAT (POINTER) := SOUNDFOURSTAT (POINTER) + 1
END;
END FOURSOUND;
```

```
PROCEDURE ENDSOUNDONE;
BEGIN
INTEGER LISTEND,LISTBEGIN;
IF (Z>=4) AND (BAD=TRUE) THEN
BEGIN
ECOUNT := ECOUNT + 1;
LISTBEGIN := 1;
LISTEND := EPOINT := 5 ;
WHILE (CODE(Z|1)¬=CLASS(EPOINT))DO
BEGIN
IF(CODE(Z|1)<CLASS(EPOINT))THEN
LISTEND := EPOINT;
IF(CODE(Z  |1)>CLASS(EPOINT))THEN
LISTBEGIN := EPOINT;
EPOINT :=(LISTEND+LISTBEGIN)DIV 2
END;
ENDONE(EPOINT) := ENDONE(EPOINT)+1
END;
END ENDSOUNDONE;


PROCEDURE ENDSOUNDTWO ;
BEGIN
INTEGER LISTEND,LISTBEGIN;
STRING(2) SOUND;
IF(Z>=5)THEN
BEGIN
E_1COUNT := E_1COUNT + 1;
LISTBEGIN := EPOINT*5-4;
LISTEND := EPOINT := LISTBEGIN+4;
SOUND(0|1):=CODE(Z|1);SOUND(1|1):=CODE(Z-1|1);
WHILE(SOUND(0|2)¬=SOUNDTWO(EPOINT)) DO
BEGIN
IF(SOUND(0|2)<SOUNDTWO(EPOINT)) THEN
LISTEND := EPOINT;
IF(SOUND(0|2)>SOUNDTWO(EPOINT)) THEN
LISTBEGIN := EPOINT;
EPOINT :=(LISTEND+LISTBEGIN)DIV 2
END;
ENDTWOSTAT(EPOINT):=ENDTWOSTAT(EPOINT)+1
END;
END ENDSOUNDTWO;
```

```
PROCEDURE PRINTONE;
BEGIN
IOCONTROL (3);
WRITE ("PERCENTAGE OF WORDS WITH FIRST SOUND AS NOTED.");
WRITE(" ");
WRITE("NUMBER OF WORDS IN SAMPLE OF STATED LENGTH IS :");
WRITE(Q);
WRITE(" ");
FOR I := 1 UNTIL HEADERCOUNT DO
WRITE (HEADER(I));
WRITE(" ");
FOR I := 1 UNTIL 5 DO
WRITEON("              ",CLASS(I),"     ");
IOCONTROL (2);
WRITE(" ");
IOCONTROL (2);
FOR J := 1 UNTIL 5 DO
BEGIN
IF (SOUNDONE(J) = 0 )THEN
WRITEON(SOUNDONE(J))
ELSE
BEGIN
SOUNDONE(J) := SOUNDONE(J)/Q*100;
STATMUX1 (N,J) := SOUNDONE(J);
WRITEON(SOUNDONE(J))
END;
END;
END PRINTONE;
```

```
PROCEDURE PRINTTWO;
BEGIN
INTEGER P;
IOCONTROL (3);
WRITE("    PERCENTAGE OF WORDS LONGER THAN ONE SOUND WITH
BEGINING TWO-SOUND COMBINATIONS AS NOTED.");
WRITE(" ");
WRITE("NUMBER OF WORDS IN SAMPLE OF STATED LENGTH IS :");
WRITE(TWOCOUNT);
WRITE(" ");
FOR I := 1 UNTIL HEADERCOUNT DO
WRITE (HEADER(I));
WRITE(" ");
FOR K := 1 UNTIL 5 DO
BEGIN
IOCONTROL (2);
FOR L := 4 STEP -1 UNTIL 0 DO
BEGIN
WRITEON("              ",SOUNDTWO(K*5-L),"    ");
END;
IOCONTROL (2);
WRITE(" ");
IOCONTROL (2);
FOR M := 4 STEP -1 UNTIL 0 DO
BEGIN
P := K*5-M;
IF (SOUNDTWOSTAT(P) = 0) THEN
WRITEON(SOUNDTWOSTAT(P))
ELSE
BEGIN
SOUNDTWOSTAT(P) := (SOUNDTWOSTAT(P)/TWOCOUNT)*100;
VECTWO(5-M) := VECTWO(5-M) + SOUNDTWOSTAT(P);
WRITEON(SOUNDTWOSTAT(P))
END;
END;
END;
WRITE(" ");
WRITE (" ") ;
IOCONTROL (2) ;
WRITE("ANALYSIS OF SECOND SOUND AS THE PERCENTAGE OF");
WRITE("OCCURANCE IN THE SECOND POSITION.");
WRITE(" ");
IOCONTROL (2) ;
FOR J := 1 UNTIL 5 DO
WRITEON(VECTWO(J))
END PRINTTWO;
```

```
PROCEDURE PRINTTHREE;
BEGIN
INTEGER P;
IOCONTROL (3);
WRITE("  PERCENTAGE OF WORDS LONGER THAN TWO SOUNDS WITH
BEGINING THREE-SOUND COMBINATIONS AS NOTED.");
WRITE(" ");
WRITE("NUMBER OF WORDS IN SAMPLE OF STATED LENGTH IS :");
WRITE(THREECOUNT);
WRITE(" ");
FOR I := 1 UNTIL HEADERCOUNT DO
WRITE (HEADER(I));
WRITE(" ");
FOR I := 1 UNTIL 25 DO
BEGIN
IOCONTROL (2);
FOR J := 4 STEP -1 UNTIL 0 DO
WRITEON("          ",SOUNDTHREE(I*5-J),"     ");
IOCONTROL (2);
WRITE(" ");
IOCONTROL (2);
FOR M := 4 STEP -1 UNTIL 0 DO
BEGIN
P := I*5-M;
IF(SOUNDTHREESTAT(P) = 0) THEN
WRITEON(SOUNDTHREESTAT(P))
ELSE
BEGIN
SOUNDTHREESTAT(P) := (SOUNDTHREESTAT(P)
/THREECOUNT)*100;
VECTHREE(5-M) := VECTHREE(5-M) + SOUNDTHREESTAT(P);
WRITEON(SOUNDTHREESTAT(P))
END;
END;
IOCONTROL (2);
WRITE(" ")
END;
WRITE(" ");
WRITE(" ");
IOCONTROL (2) ;
WRITE("ANALYSIS OF THIRD SOUND AS TO THE PERCENTAGE OF");
WRITE("OCCURANCE IN THE THIRD POSITION.");
WRITE(" ");
IOCONTROL (2) ;
FOR K := 1 UNTIL 5 DO
WRITEON (VECTHREE(K))
END PRINTTHREE;
```

```
PROCEDURE PRINTFOUR;
BEGIN
INTEGER P;
IOCONTROL (3);
WRITE("PERCENTAGE OF WORDS LONGER THAN THREE SOUNDS WITH
BEGINING FOUR-SOUND COMBINATIONS AS NOTED.");
WRITE (" ");
WRITE("NUMBER OF WORDS IN SAMPLE OF STATED LENGTH IS :");
WRITE(FOURCOUNT);
WRITE(" ");
FOR I := 1 UNTIL HEADERCOUNT DO
WRITE (HEADER(I));
WRITE (" ");
FOR I := 1 UNTIL 125 DO
BEGIN IOCONTROL (2);
FOR M := 4 STEP -1 UNTIL 0 DO
BEGIN
P := I * 5 - M;
IF (SOUNDFOURSTAT (P) = 0 ) THEN
ELSE
BEGIN
SOUNDFOURSTAT (P) := (SOUNDFOURSTAT (P) / FOURCOUNT ) * 100;
WRITE(SOUNDFOUR(P),SOUNDFOURSTAT(P));
WRITE(" ")
END;
END;
END;
END PRINTFOUR;


PROCEDURE PRINTEND;
BEGIN
IOCONTROL (3);
WRITE (" PERCENTAGE OF WORDS WITH LAST  SOUND AS NOTED.");
WRITE(" ");
WRITE("NUMBER OF WORDS IN SAMPLE OF STATED LENGTH IS :");
WRITE(ECOUNT);
WRITE(" ");
FOR I := 1 UNTIL HEADERCOUNT DO
WRITE(HEADER(I));
WRITE(" ");
FOR I := 1 UNTIL 5 DO
WRITEON("              ",CLASS(I),"    ");
IOCONTROL (2);
WRITE(" ");
IOCONTROL (2);
FOR J := 1 UNTIL 5 DO
BEGIN
IF(ENDONE(J)=0) THEN
WRITEON(ENDONE(J))
ELSE
BEGIN
ENDONE(J) := ENDONE(J)/ECOUNT*100    ;
WRITEON(ENDONE(J))
END;
END;
END PRINTEND;
```

```
PROCEDURE PRINTENDTWO;
BEGIN
INTEGER P;
IOCONTROL (3);
WRITE("    PERCENTAGE OF WORDS LONGER THAN ONE SOUND WITH
ENDING   TWO SOUND COMBINATIONS AS NOTED.");
WRITE(" ");
WRITE("NUMBER OF WORDS IN SAMPLE OF STATED LENGTH IS :");
WRITE(E_1COUNT);
WRITE(" ");
FOR I := 1 UNTIL HEADERCOUNT DO
WRITE(HEADER(I));
WRITE(" ");
FOR K := 1 UNTIL 5 DO
BEGIN
IOCONTROL (2);
FOR L := 4 STEP -1 UNTIL 0 DO
BEGIN
WRITEON("            ",SOUNDTWO(K*5-L),"    ");
END;
IOCONTROL (2);
WRITE(" ");
IOCONTROL (2);
FOR M := 4 STEP -1 UNTIL 0 DO
BEGIN
P := K*5-M;
IF (ENDTWOSTAT  (P) = 0) THEN
WRITEON(ENDTWOSTAT  (P))
ELSE
BEGIN
ENDTWOSTAT(P) := (ENDTWOSTAT(P)/E_1COUNT)*100;
ENDVEC2(5-M):=ENDVEC2(5-M)+ENDTWOSTAT(P);
WRITEON(ENDTWOSTAT(P))
END;
END;
END;
WRITE(" ");
WRITE("ANALYSIS OF NEXT TO LAST SOUND AS TO THE PERCENTAGE")
WRITE("OF OCCURANCE IN THAT POSITION.");
WRITE(" ");
IOCONTROL (2);
FOR J := 1 UNTIL 5 DO
WRITEON(ENDVEC2(J))
END PRINTENDTWO              ;
```

```
PROCEDURE PRINTMUX1;
BEGIN
FOR S := 1 UNTIL 5 DO
BEGIN
B := 0;
FOR R := 1 UNTIL N DO
BEGIN
B := B + STATMUX1(R,S)
END;
FOR T := 1 UNTIL N DO
BEGIN
PMUX1(S,T) := (STATMUX1(T,S)/B);
END;
END;
WRITE(" ");
WRITE(" ");
IOCONTROL (3);
IOCONTROL (2);
WRITE ("   PROBABILITY MATRIX FOR FIRST SOUNDS:    ");
WRITE (" ");
IOCONTROL (2);
FOR I := 1 UNTIL 5 DO
BEGIN
WRITEON ("        ",CLASS(I),"        ")
END;
IOCONTROL (2);
WRITE (" ");
IOCONTROL (2);
FOR K := 1 UNTIL 5 DO
BEGIN
IOCONTROL (2);
WRITE (" ");
WRITE (" ");
IOCONTROL (2);
FOR J := 1 UNTIL N DO
BEGIN
WRITEON (PMUX1(J,K))
END;
END;
END PRINTMUX1;
```

```
PROCEDURE CLEARMEM;
BEGIN
INTEGER P, S, T;
P := 0;
S := 0;
T := 0;
FOR I := 1 UNTIL 5 DO
BEGIN
VECTWO(I) := 0 ;
VECTHREE(I) := 0 ;
VECFOUR (I) := 0;
ENDVEC2(I) := 0;
ENDONE(I) := 0;
SOUNDONE (I) := 0;
FOR J := 1 UNTIL 5 DO
BEGIN
P := P + 1;
ENDTWOSTAT (P) := 0;
SOUNDTWOSTAT (P) := 0;
FOR K := 1 UNTIL 5 DO
BEGIN
S := S + 1;
SOUNDTHREESTAT (S) := 0;
FOR L := 1 UNTIL 5 DO
BEGIN
T := T + 1;
SOUNDFOURSTAT (T) := 0
END;
END;
END;
END;
END CLEARMEM;
```

```
CLASS (1) := "F";
CLASS (2) := "N";
CLASS (3) := "P";
CLASS (4) := "S";
CLASS (5) := "V";
GENERATETWO;
GENERATETHREE;
GENERATEFOUR;
GETARRAY;
N := 1;
WHILE ( N > 0) DO
BEGIN
IOCONTROL (3);
INITIALIZE;
CLEARMEM;
WHILE (GETWORD (WORD)) DO
BEGIN
CONVERT (CODE);
ONESOUND;
TWOSOUND;
THREESOUND;
FOURSOUND;
ENDSOUNDONE;
ENDSOUNDTWO
END;
PRINTONE;
PRINTTWO;
PRINTTHREE;
PRINTFOUR;
PRINTEND;
PRINTENDTWO;
READ (N)
END;
N := 5;
PRINTMUX1
END.


COMMENT**************************************************
ROMI MUST ALWAYS BE THE FIRST DATA SET FOLLOWING THE "%DATA"
CARD (JCL).
NOTE THAT ROMI IS WRITTEN SUCH THAT EACH CHARACTER OF ROMI
IS FOLLOWED IMMEDIATELY (SEPARATED BY ONE SPACE) BY THE
CATEGORY TO WHICH IT BELONGS.  TWO LETTERS JOINED BY A "-"
ARE READ AS A SINGLE LETTER.
*********************************************************;
```

```
*************************************************************

        C O M P U T E R   P R O G R A M   NUMBER 2

*************************************************************


//COOK8 JOB (0405,1404,MA52),'SMC 2737',TIME=5
// EXEC ALGOLW,REGION=350K
//SYSPRINT DD SYSOUT=A,DCB=BLKSIZE=665,SPACE=(CYL,(3,1))
//ALGOL.SYSIN DD *
%ALGOL T=300,P=300
$NOLIST


BEGIN
COMMENT
*************************************************************
*************************************************************

FINAL PROGRAM OUTLINE:


1.   DIMENSION AREA
2.   PROCEDURE INITIALIZE
3.         "     SETCOUNT
4.         "     GENERATETWO
5.         "     GENERATETHREE
6.         "     GENERATEFOUR
7.         "     GENERATESIX
8.         "     GETARRAY
9.   LOGICAL PROCEDURE GETWORD
10.  PROCEDURE CONVERT
11.        "     PROB_TWO
12.        "     DECIDE
                (INCLUDES PATHS FOR 1,2,3,4,5,6, SOUND WORDS)
13.        "     COMPUTE_NORM
14.        "     RAW_ANALYSIS
                (INCLUDES PATHS FOR 1,2,3,4,5,6, SOUND WORDS)
15'        "     COMPUTE_ANY
16         "     CLEARMEM
17         "     FINDSOUND
18         "     ONESOUND
19         "     GRAPH
20. MAIN PROGRAM

*************************************************************
*************************************************************
;
```

```
COMMENT INPUT DATA WRITTEN IN "ROMI" WITH AT LEAST ONE BLANK
BETWEEN WORDS.
WORD MUST START AND END ON SAME CARD.;


INTEGER DIM;
INTEGER SAMPLESIZE;
PROCEDURE DUMMY;
BEGIN
COMMENT ***** GLOBAL DECLARATIONS:        ;
INTEGER TWOPOINT,ONEPOINT,SPACE;
INTEGER INDEX, POINTER, Z,TP, WP, HEADCOUNT, EPOINT, NUMBER;
INTEGER FOURPOINT;
INTEGER THREEPOINT;
INTEGER N;
STRING (80) TEXT;
STRING (36) WORD, CODE;
LOGICAL FOUND, BAD;
STRING (2) ARRAY ROMI (1::101,1::2);
STRING(6) ARRAY CLASS(0::3);
STRING(6) ARRAY SOUNDTWO(0::15);
STRING(6) ARRAY SOUNDTHREE(0::63);
STRING(6) ARRAY SOUNDFOUR(0::255);
STRING(6) ARRAY SOUNDFIVE(0::1023);
STRING(6) ARRAY SOUNDSIX(0::4095);
STRING (80) ARRAY HEADER (1::10);
REAL ARRAY ONE1,ONE2,ONE3,ONE4,ONE5,ONE6,E6 (0::4,0::DIM);
REAL ARRAY ONETWO2,ONETWO3,ONETWO4,ONETWO5(0::16,0::DIM);
REAL ARRAY ONETWO6,FIRSTLAST5,FIRSTLAST6(0::16,0::DIM);
REAL ARRAY EE26(0::16,0::DIM);
REAL ARRAY ONETWO33,ONETWO34,ONETWO35,
           ONETWO36(0::64,0::DIM);
REAL ARRAY ONETWO344,ONETWO345,ONETWO346,
           F12EE26(0::256,0::DIM);
REAL ARRAY ONETWO34E6,ONETWO34E5(0::1024,0::DIM);
REAL ARRAY ALLSOUND(0::4096,0::DIM);
```

```
COMMENT  ****************************************************
PROCEDURE INITIALIZE RESETS THE VALUES OF THE GLOBAL
COUNTERS USED THROUGHOUT THE PROGRAM.  INITIALIZE ALSO READS
AND PRINTS THE DATA SET IDENTIFIER "HEADER," SETTING
"TEXT POINTER," "TP," TO 80 CAUSES THE PROCEDURE GETWORD
TO READ THE FIRST DATA CARD.
*********************************************************;

COMMENT**********************
HEADERCOUNT IS THE NUMBER OF CARDS PRECEEDING THE DATA
WHICH DESCRIBE THE TYPE/QUALITY/ETC. OF THE DATA BUT ARE
NOT TO BE READ AS PART OF THE DATA.
******************************;

PROCEDURE INITIALIZE;
BEGIN
INDEX:=0;
READ (HEADCOUNT);
FOR I := 1 UNTIL HEADCOUNT DO
BEGIN
HEADER(I) := " ";
READCARD(HEADER(I));
END;
IOCONTROL (2);
WRITE(" ");
IOCONTROL (2);
TP := 80
END INITIALIZE;



PROCEDURE SETCOUNT;
BEGIN
FOR I := 1 UNTIL DIM DO
BEGIN
ONE1(4,I):=0;
ONE2(4,I):=0;
ONE3(4,I):=0;
ONE4(4,I):=0;
ONE5(4,I):=0;
ONE6(4,I):=0;
E6(4,I):=0;
ONETWO2(16,I):=0;
ONETWO3(16,I):=0;
ONETWO4(16,I):=0;
ONETWO5(16,I):=0;
ONETWO6(16,I):=0;
FIRSTLAST5(16,I):=0;
FIRSTLAST6(16,I):=0;
EE26(16,I):=0;
ONETWO33(64,I):=0;
ONETWO34(64,I):=0;
ONETWO35(64,I):=0;
ONETWO36(64,I):=0;
ONETWO344(256,I):=0;
ONETWO345(256,I):=0;
ONETWO346(256,I):=0;
F12EE26(256,I):=0;
ONETWO34E6(1024,I):=0;
ONETWO34E5(1024,I):=0;
ALLSOUND(4096,I):=0;
END;
END SETCOUNT;
```

```
COMMENT   ************************************************
PROCEDURE GENERATETWO PROVIDES THE 16 TWO-SOUND CLASS LABEL
DESIGNATIONS IN ALPHABETICAL ORDER.  THESE LABELS ARE
STORED IN THE ONE DIMENSION ARRAY "SOUNDTWO."  THE LABELS
CORRESPOND TO THE 16 ROWS OF THE SET OF REAL ARRAYS NAMED
"ONETWO_","FIRSTLAST_", AND "FF26."
************************************************;

PROCEDURE GENERATETWO;
BEGIN
INTEGER Y;
Y := 0;
FOR I := 0 UNTIL 3 DO
BEGIN
FOR J := 0 UNTIL 3 DO
BEGIN
SOUNDTWO(Y) := " ";
SOUNDTWO (Y)(0|1) := CLASS (I)(0|1);
SOUNDTWO (Y)(1|1) := CLASS (J)(0|1);
Y := Y + 1
END;
END;
END GENERATETWO;


COMMENT   ************************************************
PROCEDURE GENERATETHREE TAKES THE 16 ELEMENT ARRAY
"SOUNDTWO" AND GENERATES THE  64 POSSIBLE COMBINATIONS OF
THE FIRST THREE SOUNDS OF A WORD IN ALPHABETICAL ORDER.
THE  64 ELEMENTS ARE STORED IN THE ONE DIMENSIONAL ARRAY
"SOUNDTHREE" AND ARE USED AS LABELS FOR THE  64 ROWS OF THE
SET OF REAL ARRAYS "ONETWO3_."
************************************************;


PROCEDURE GENERATETHREE;
BEGIN
INTEGER Y;
Y := 0;
FOR I := 0 UNTIL 15 DO
BEGIN
FOR J := 0 UNTIL 3 DO
BEGIN
SOUNDTHREE(Y) := " ";
SOUNDTHREE(Y)(0|2) := SOUNDTWO(I)(0|2);
SOUNDTHREE(Y)(2|1) := CLASS(J)(0|1);
Y := Y+1
END;
END;
END GENERATETHREE;

PROCEDURE GENERATEFOUR;
BEGIN
INTEGER Y;
Y :=0;
FOR I := 0 UNTIL 63 DO
BEGIN
FOR J := 0 UNTIL 3 DO
BEGIN
SOUNDFOUR(Y):=" ";
SOUNDFOUR(Y)(0|3) := SOUNDTHREE(I)(0|3);
SOUNDFOUR(Y)(3|1):=CLASS(J)(0|1);
Y := Y+1
END;
END;
END GENERATEFOUR;
```

```
PROCEDURE GENERATESIX;
BEGIN
INTEGER X,Y;
Y := 0:
X := -1;
FOR I:=0 UNTIL 255 DO
BEGIN
FOR J := 0 UNTIL 3 DO
BEGIN
X :=X+1;
SOUNDFIVE(X):=" ":
SOUNDFIVE(X)(0|4):=SOUNDFOUR(I)(0|4);
SOUNDFIVE(X)(4|1):=CLASS(J)(0|1);
FOR K := 0 UNTIL 3 DO
BEGIN
SOUNDSIX(Y):=" ";
SOUNDSIX(Y)(0|5):=SOUNDFIVE(X)(0|5);
SOUNDSIX(Y)(5|1):=CLASS(K)(0|1);
Y:=Y+1
END;
END;
END;
END GENERATESIX;

COMMENT  ***********************************************
PROCEDURE GETARRAY READS IN THE 101 CHARACTERS USED BY THIS
PROGRAM TO REPRESENT THE INTERNATIONAL PHONETIC ALPHABET.
EACH SOUND REPRESENTATION IS PLACED IN A TWO COLUMN ARRAY
"ROMI."   COLUMN ONE IS THE SOUND REPRESENTATION AND COLUMN
TWO CONTAINS THE CLASS: EITHER NASAL, FRICATIVE, STOP, OR
VOWEL.   THE NUMBER OF CARDS USED TO READ IN "ROMI" CANNOT
BE VARIED WITHOUT ALTERING THE PROGRAM.   THE NUMBER OF
ELEMENTS IN "ROMI" IS NOT VARIABLE.   "ROMI" IS PRINTED OUT
A CARD AT A TIME, JUST AS IT IS TYPED IN THE DATA CARDS.
*********************************************************;


PROCEDURE GETARRAY;
BEGIN
STRING(81) PHONOL;
INTEGER B,A;
B:=1;
A := 1:
WHILE (A <= 27) DO
BEGIN
PHONOL := " ";
READCARD (PHONOL);
TP:=0;
WHILE(TP <= 79)DO
BEGIN
WHILE (PHONOL (TP|1) =" ") AND (TP<=79) DO
TP:= TP+1;
IF ( TP ¬= 80 ) THEN
BEGIN
IF (PHONOL (TP+1|1)¬="-") THEN
BEGIN
ROMI (B,1): = PHONOL (TP|1);
TP: = TP+2;
END
ELSE
BEGIN
ROMI (B,1)(0|1): = PHONOL (TP|1);
ROMI (B,1)(1|1): = PHONOL (TP+2|1);
TP:=TP+4;
END;
ROMI (B,2): = PHONOL (TP|1);
B: = B+1;
TP := TP+1
END
ELSE
A := A+1;
END
```

```
END;
TP := 80
END GETARRAY;
```

```
COMMENT  ***************************************************
LOGICAL PROCEDURE GETWORD READS THE INPUT DATA WRITTEN ON
"ROMI" INTO A BUFFER CALLED "TEXT," ONE CARD AT A TIME AND
ECHO PRINTS EACH CARD WITH A SEQUENTIAL NUMBER IDENTIFIER
"INDEX."   THE BUFFER, "TEXT," IS SCANNED ONE CHARACTER AT
A TIME UNTIL A "WORD" IS BUILT.  LEADING AND TRAILING
BLANKS DELIMIT A "WORD."    "WORD" IS THE GLOBAL PARAMETER
OF STRING TYPE OF MAXIMUM LENGTH OF 36 CHARACTERS WHICH
IS DELIVERED TO THE MAIN PROGRAM AND SUBSEQUENT PROCEDURES.
WHEN THE CHARACTER "%" IS ENCOUNTERED ON A DATA CARD IN
OTHER THAN CARD COLUMN 1 THE LOGICAL VALUE OF "GETWORD" IS
SET TO FALSE THUS SIGNALLING THE END OF A DATA SET.
"WORDS" MUST START AND END ON THE SAME CARD.
************************************************************;

LOGICAL PROCEDURE GETWORD (STRING(36) RESULT WORD);
BEGIN
WP := 0;
IF(TP=80) THEN
BEGIN
INDEX:=INDEX+1;
 READCARD (TEXT);
TP:=0
END;
WHILE(TEXT(TP|1)=" ")  DO
BEGIN
TP:=TP+1;
IF (TP=80) THEN
BEGIN
 READCARD (TEXT);
INDEX:=INDEX+1;
TP:=0
END
END;
WORD:=" ";
WHILE (TP<80) AND (TEXT(TP|1)¬=" ") DO
BEGIN
WORD (WP|1):= TEXT(TP|1);
WP := WP + 1;
TP := TP + 1
END;
WP := WP-1;
    WORD¬="%"
END GETWORD;
```

```
COMMENT  ***********************************************************
PROCEDURE "CONVERT" OPERATES ON THE GLOBAL VARIABLE "WORD"
OBTAINED FROM "GETWORD."   "WORD" IS SCANNED CHARACTER BY
CHARACTER TO OBTAIN EACH "ROMI" SOUND REPRESENTATION.
A BINARY SEARCH OF THE ARRAY "ROMI" IS CONDUCTED TO
OBTAIN THE SOUND CLASS EQUIVALENT FOR EACH "SOUND" OF
"WORD."   A NEW STRING, "CODE," IS GENERATED FROM THE FOUR
SOUND DESIGNATIONS CORRESPONDING TO THE INTERNATIONAL
PHONETIC ALPHABET SPELLING OF "WORD."   "CONVERT" IS ALSO
CAPABLE OF EDITING INPUT DATA AND PRINTS OUT A WARNING
LABEL ALONG WITH THE PARTICULAR "SOUND" REPRESENTATION
WHICH DID NOT APPEAR IN "ROMI" AND THE "WORD" IN WHICH THE
ERROR OCCURRED.   A SAMPLE OUTPUT FROM "CONVERT" WOULD BE
"FVNSFVNN."  THE COUNTER "Z" IS ALSO A GLOBAL VARIABLE,
AND TELLS THE NUMBER OF IPA SOUNDS IN EACH STRING "CODE."
"Z" IS USED TO DRIVE LATER PORTIONS OF THE PROGRAM.
IF A MISTAKE IS MADE ON A DATA CARD SUCH THAT A CHARACTER
STRING IS FOUND THAT IS NOT A MEMBER OF ROMI, THE "WORD"
CONTAINING THE UNKNOWN "SOUND" IS PRINTED OUT ALONG WITH
AN ERROR MESSAGE.  WHEN THIS CONDITION ARISES, "BAD" IS SET
TO FALSE AND NO ANALYSIS IS DONE ON THE "CODE" BY THE
REMAINDER OF THE PROGRAM.
***********************************************************;

PROCEDURE CONVERT (STRING(36) RESULT CODE);
BEGIN
STRING(2) SOUND;
INTEGER I,TEST;
INTEGER LISTEND, LISTBEGIN, POINTER;
CODE := " ";
BAD := TRUE;
I := 0;
Z := 0;
WHILE I<=WP DO
BEGIN
TEST := 0;
FOUND := TRUE;
SOUND:=" ";
IF WORD (I+1|1)¬= "-" THEN
BEGIN
SOUND      : = WORD (I|1);
I:=I+1; END
ELSE
BEGIN
SOUND(0|1):= WORD(I|1);
SOUND(1|1):= WORD(I+2|1);
I:=I+3
END;
 LISTBEGIN:= 1;
LISTEND  := 101;
POINTER := LISTEND;
WHILE ( FOUND ) DO
BEGIN
TEST := TEST+1;
IF SOUND < ROMI(POINTER,1) THEN
 LISTEND:= POINTER;
IF SOUND > ROMI(POINTER,1) THEN
 LISTBEGIN:= POINTER;
IF ( SOUND = ROMI ( POINTER,1 ))THEN
FOUND := FALSE
ELSE
BEGIN
IF (TEST = 12 ) THEN
BEGIN
WRITE(INDEX,TEXT);
WRITE("ERONEOUS DATA","WORD = ",WORD, "SOUND = ",SOUND);
FOUND := FALSE;
I := WP+1;
BAD := FALSE
END;
POINTER := TRUNCATE ( ( LISTEND + LISTBEGIN ) / 2 )
```

```
END;
END;
CODE ( Z|1 ) := ROMI ( POINTER, 2 ) (0|1 );
Z := Z+1          ;
IF (BAD = FALSE) THEN
Z := 1
END;
Z := Z-1
END CONVERT;
```

```
COMMENT*********************************************************
PROCEDURE PROB_TWO CALCULATES THE PROBABILISTIC PROGRESSIONS
(PP'S). THE PP'S, WHEN CALCULATED, ARE REASSIGNED TO THE
CORRESPONDING BASE DATA FREQUENCIES ARRAYS. THE CALLING
ARGUEMENTS ARE:
1E. REAL ARRAY BASE: CORRESPONDS TO THE CUMMULATIVE
SOUND COMBINATION FOR WHICH THE PP IS BEING CALCULATED.
2) REAL ARRAY PARTPROB: THE ARRAY CONTAINING THE
"CONDITIONAL" PORTION OF THE PP.
3) REAL ARRAY OUTPUT: THE ARRAY TO WHICH THE PP'S ARE TO
BE ASSINGED. OUTPUT MUST BE DIMENSIONED IDENTICALLY WITH
BASE.
4) LC: THE NUMBER OF ELEMENTS IN PARTPROB LESS ONE.
5) FNID: THE NUMBER OF ELEMENTS IN BASE DIVIDED BY THE
NUMBER OF ELEMENTS IN PART PROB LESS ONE.
*********************************************************;

PROCEDURE PROB_TWO(REAL ARRAY BASE(*,*);REAL ARRAY PARTPROB
(*,*);REAL ARRAY OUTPUT(*,*);INTEGER VALUE LC,FNID);
BEGIN
REAL B;
INTEGER T,P;
T :=-1;
FOR H := 0 UNTIL LC DO
BEGIN
FOR I := 0 UNTIL FNID DO
BEGIN
B := 0;
T := T+1 ;
FOR J := 1 UNTIL NUMBER DO
B := B+BASE(T,J)*PARTPROB(H,J);
FOR K := 1 UNTIL NUMBER DO
BEGIN
IF(B¬=0) THEN
OUTPUT(T,K) := BASE(T,K)*PARTPROB(H,K)/B
ELSE
OUTPUT(T,0):=-1
END;
END;
END;
END PROB_TWO;
```

```
COMMENT*************************************************************
PROCEDURE DECIDE READS TEST SAMPLES, ANALYSES THEM AND
PRESENTS A SET OF NUMBERS TO THE ANALYSIST FOR A DECISION.
"SAMPLE SIZE" IS THE NUMBER OF WORDS IN THE TEST SAMPLE
LESS ONE.
"Q" COUNTS THE NUMBER OF WORDS IN A TEST SAMPLE FOR
PRINTING A IDENTIFICATION.
"N" COUNTS THE NUMBER OF SAMPLES (WHICH IS LIMITED ONLY BY
TIME AVAILABLE ON THE COMPUTER).
THE "CASE" STATEMENT ALLOWS THE SELECTION OF ONE PROGRAM
SEGMENT FROM A LIST FOR EXECUTION DEPENDING ON THE
ORDINAL VALUE OF THE "CASE" STATEMENT PARAMETER "P."  THE
VALUE OF "P" DEPENDS ON "Z" AND IS ASSIGNED BY AN "IF"
STATEMENT.
**BY VARYING THE RANGE OF "P" WITH CHANGES TO THE "IF"
STATEMENT IN "DECIDE" AND "RAW_ANALYSIS" THE PROGRAMMER
CAN SELECT THE PARTITION OF THE DATA TEST SETS AND SAMPLE
TEST SETS ACCORDING TO WORD LENGTH.
"T" TELL LATER PARTS OF "DECIDE" HOW MANY ROWS OF THE
ARRAY "DECISION" WERE USED FOR THAT WORD...
************************************************************;
PROCEDURE DECIDE;
BEGIN
REAL C,B;
REAL ARRAY DECISION(0::9,1::NUMBER);
REAL ARRAY SAVEDECISION(0::SAMPLESIZE,1::NUMBER);
INTEGER P,T,Q;
INTFIELDSIZE:=3;
Q :=0;
N := N+1;
WRITE("SAMPLE NUMBER",N);
FOR K:=0 UNTIL SAMPLESIZE DO
BEGIN


Q := Q+1;
FOUND:=GETWORD(WORD);
CONVERT(CODE);
WRITE(" "); IOCONTROL(2);
WRITE("SAMPLE WORD NUMBER",Q,"IN CODE FORM IS:",CODE);
WRITE(" "); IOCONTROL(2);

IF(BAD=TRUE) THEN
BEGIN


IF(Z<=5) THEN P:=Z+1 ELSE P:=6;
CASE  P  OF   BEGIN


COMMENT**** WORD LENGTH = ONE*****;

BEGIN
T:=0;
FINDSOUND(0);
ONEPOINT:=POINTER;
FOR I := 1 UNTIL NUMBER DO
DECISION(0,I):=ONE1(POINTER,I)
END;


COMMENT**** WORD LENGTH = TWO*****;

BEGIN
FINDSOUND(0);
T := 1;
ONEPOINT:=POINTER;
FOR I:=1 UNTIL NUMBER DO
DECISION(0,I):=ONE2(POINTER,I);
FINDSOUND(1);
POINTER:=ONEPOINT*4+POINTER;
FOR I:=1 UNTIL NUMBER DO
```

```
DECISION(1,I):=ONETWO2(POINTER,I);
END;


COMMENT**** WORD LENGTH = THREE*****;

BEGIN
T := 2;
FINDSOUND(0):
ONEPOINT:=POINTER:
FOR I:=1 UNTIL NUMBER DO
DECISION(0,I):=ONE3(POINTER,I);
FINDSOUND(1);
POINTER:=ONEPOINT*4+POINTER:
TWOPOINT:=POINTER;
FOR I:=1 UNTIL NUMBER DO
DECISION(1,I):=ONETWO3(POINTER,I):
FINDSOUND(2);
POINTER:=TWOPOINT*4+POINTER;
FOR I:=1 UNTIL NUMBER DO
DECISION(2,I):=ONETWO33(POINTER,I);
END;


COMMENT**** WORD LENGTH = FOUR*****;

BEGIN
T := 3;
FINDSOUND(0):
ONEPOINT:=POINTER;
FOR I:=1 UNTIL NUMBER DO
DECISION(0,I):=ONE4(POINTER,I):
FINDSOUND(1);
POINTER:=ONEPOINT*4+POINTER:
TWOPOINT:=POINTER:
FOR I:=1 UNTIL NUMBER DO
DECISION(1,I):=ONETWO4(POINTER,I):
FINDSOUND(2):
POINTER:=TWOPOINT*4+POINTER;
ONEPOINT:=POINTER;
FOR I:=1 UNTIL NUMBER DO
DECISION(2,I):=ONETWO34(POINTER,I);
FINDSOUND(3);
POINTER:=ONEPOINT*4+POINTER:
FOR I:=1 UNTIL NUMBER DO
DECISION(3,I):=ONETWO344(POINTER,I)
END;


COMMENT**** WORD LENGTH = FIVE*****;

BEGIN
T := 4;
FINDSOUND(0);
ONEPOINT:=POINTER:
FOR I := 1 UNTIL NUMBER DO
DECISION(0,I):=ONE5(POINTER,I);
FINDSOUND(1);
POINTER:=ONEPOINT*4+POINTER:
TWOPOINT:=POINTER:
FOR I:=1 UNTIL NUMBER DO
DECISION(1,I):=ONETWO5(POINTER,I);
FINDSOUND(2):
POINTER:=TWOPOINT*4+POINTER;
THREEPOINT:=POINTER:
FOR I:=1 UNTIL NUMBER DO
DECISION(2,I):=ONETWO35(POINTER,I);
FINDSOUND(3);
POINTER:=THREEPOINT*4+POINTER:
TWOPOINT:=POINTER:
FOR I:=1 UNTIL NUMBER DO
DECISION(3,I):=ONETWO345(POINTER,I);
```

```
FINDSOUND(4);
POINTER:=TWOPOINT*4+POINTER;
FOR I:=1 UNTIL NUMBER DO
DECISION(4,I):=ONETWO345(POINTER,I)
END;


COMMENT**** WORD LENGTH = SIX & GREATER*****;

BEGIN
T := 8;
FINDSOUND(0);
ONEPOINT:=POINTER;
FOR I:=1 UNTIL NUMBER DO
DECISION(0,I):=ONE6(POINTER,I);
FINDSOUND(1);
POINTER:=ONEPOINT*4+POINTER;
TWOPOINT:=POINTER;
FOR I:=1 UNTIL NUMBER DO
DECISION(1,I):=ONETWO6(POINTER,I);
FINDSOUND(2);
POINTER:=TWOPOINT*4+POINTER;
THREEPOINT:=POINTER;
FOR I:=1 UNTIL NUMBER DO
DECISION(2,I):=ONETWO36(POINTER,I);
FINDSOUND(3);
POINTER:=THREEPOINT*4+POINTER;
FOURPOINT:=POINTER;
FOR I:=1 UNTIL NUMBER DO
DECISION(3,I):=ONETWO346(POINTER,I);
FINDSOUND(Z);
EPOINT:=POINTER;
POINTER := ONEPOINT*4+POINTER;
FOR I:=1 UNTIL NUMBER DO
DECISION(4,I):=FIRSTLAST6(POINTER,I);
POINTER:=FOURPOINT*4+EPOINT;
FOR I:=1 UNTIL NUMBER DO
DECISION(8,I):=ONETWO34E6(POINTER,I);
SPACE:=Z-1;
FINDSOUND(SPACE);
POINTER:=EPOINT*4+POINTER;
EPOINT:=POINTER;
FOR I:=1 UNTIL NUMBER DO
DECISION(5,I):=EE26(POINTER,I);
POINTER:=FOURPOINT*15+EPOINT;
FOR I:=1 UNTIL NUMBER DO
DECISION(6,I):=ALLSOUND(POINTER,I);
POINTER:=TWOPOINT*15+EPOINT;
FOR I:=1 UNTIL NUMBER DO
DECISION(7,I):=E12EE26(POINTER,I)
END;
END;



COMMENT*************************************************************
HERE A COLUMN PRODUCT IS OBTAINED FROM THE ARRAY
"DECISION" AND ASSIGNED TO THE "KTH" ROW OF "SAVEDECISION."
ONE ROW IN "SAVEDECISION" IS GENERATED FOR EACH WORD IN THE
TEST SAMPLE SET.
*************************************************************;

FOR J:=1 UNTIL NUMBER DO
BEGIN
DECISION(9,J):=1;
FOR I:=0 UNTIL T DO
DECISION(9,J):=DECISION(9,J)*DECISION(I,J);
END;
WRITE("VALIDITY FACTORS :");
WRITE(" "); IOCONTROL(2);
FOR J := 0 UNTIL  T DO
BEGIN
```

```
FOR I := 1 UNTIL NUMBER DO
WRITEON(DECISION(J,I));
WRITE(" ")
END;
FOR J := 1 UNTIL NUMBER DO
SAVEDECISION(K,J):=DECISION(9,J)
END;
END;


COMMENT*****************************************************************
THE NUMBERS FOR THE DECISION CRITERIA ARE ABTAINED FROM THE
ARRAY "SAVEDECISION" BY TAKING BOTH A COLUMN PRODUCT AND
COLUMN ARETHMETIC MEAN.
************************************************************************;

IF(SAMPLESIZE=0) THEN
BEGIN
FOR J:=1 UNTIL NUMBER DO WRITEON(SAVEDECISION(0,J))
END
ELSE  BEGIN
WRITE(" "); IOCONTROL(2);
WRITE("VALIDITY FACTOR PRODUCTS:");
FOR J:=0 UNTIL Q-1  DO
BEGIN
WRITE(" ");
FOR I:=1 UNTIL NUMBER DO
WRITEON(SAVEDECISION(J,I))
END;
FOR J:=1 UNTIL NUMBER DO
BEGIN
B:=1;
C:=0;
FOR I := 0 UNTIL SAMPLESIZE DO
BEGIN
B := B*SAVEDECISION(I,J);
C:=C+SAVEDECISION(I,J)
END;
C:=C/(SAMPLESIZE+1)    ;
WRITE(" "); IOCONTROL(2);
WRITE(B);
WRITEON(C);
WRITE(" ")
END;
END;
END DECIDE;
```

```
COMMENT*********************************************************
PROCEDURE COMPUTE_NORM REPRESENTS A CONVENIENT METHOD TO
GROUP MULTIPLE CALLS TO THE SAME PROCEDURE WITHOUT
CLUTTERING UP THE MAIN PROGRAM.
****************************************************************;

PROCEDURE COMPUTE_NORM;
BEGIN
COMPUTE_ANY(3,ONE1);
COMPUTE_ANY(3,ONE2);
COMPUTE_ANY(3,ONE3);
COMPUTE_ANY(3,ONE4);
COMPUTE_ANY(3,ONE5);
COMPUTE_ANY(3,ONE6);
COMPUTE_ANY(3,E6);
COMPUTE_ANY(15,ONETWO2);
COMPUTE_ANY(15,ONETWO3);
COMPUTE_ANY(15,ONETWO4);
COMPUTE_ANY(15,ONETWO5);
COMPUTE_ANY(15,ONETWO6);
COMPUTE_ANY(15,FIRSTLAST5);
COMPUTE_ANY(15,FIRSTLAST6);
COMPUTE_ANY(15,EE26);
COMPUTE_ANY(63,ONETWO33);
COMPUTE_ANY(63,ONETWO34);
COMPUTE_ANY(63,ONETWO35);
COMPUTE_ANY(63,ONETWO36);
COMPUTE_ANY(255,ONETWO344);
COMPUTE_ANY(255,ONETWO345);
COMPUTE_ANY(255,ONETWO346);
COMPUTE_ANY(255,F12EE26);
COMPUTE_ANY(1023,ONETWO34E5);
COMPUTE_ANY(1023,ONETWO34E6);
COMPUTE_ANY(4095,ALLSOUND)
END COMPUTE_NORM;
```

```
COMMENT*****************************************************************
PROCECURE RAW_ANALYSIS USES AN IDENTICAL "CASE" STATEMENT
TO THAT USED IN "DECIDE" TO PARTITION THE DATA BASE SETS.
****************************************************************;

PROCEDURE RAW_ANALYSIS;
BEGIN
INTEGER P;
IF(BAD=TRUE) THEN
BEGIN
POINTER:=0;
IF(Z<=5) THEN P:=Z+1 ELSE P:=6;
CASE P OF     BEGIN

COMMENT*****WORDS WITH ONE SOUND******;

ONESOUND(ONE1,0,4);


COMMENT*****WORDS WITH TWO SOUNDS******;

BEGIN
ONESOUND(ONE2,0,4);
ONESOUND(ONETWO2,1,16)
END;


COMMENT*****WORDS WITH THREE SOUNDS******;

BEGIN
ONESOUND(ONE3,0,4);
ONESOUND(ONETWO3,1,16);
ONESOUND(ONETWO33,2,64)
END;


COMMENT*****WORDS WITH FOUR SOUNDS******;

BEGIN
ONESOUND(ONE4,0,4);
ONESOUND(ONETWO4,1,16);
ONESOUND(ONETWO34,2,64);
ONESOUND(ONETWO344,3,256)
END;


COMMENT*****WORDS WITH FIVE SOUNDS******;

BEGIN
ONESOUND(ONE5,0,4);
ONEPOINT:=POINTER;
ONESOUND(ONETWO5,1,16);
ONESOUND(ONETWO35,2,64);
ONESOUND(ONETWO345,3,256);
ONESOUND(ONETWO345,4,1024);
POINTER:=ONEPOINT;
ONESOUND(FIRSTLAST5,4,16)
END;


COMMENT*****WORDS WITH SIX & GREATER SOUNDS******;

BEGIN
ONESOUND(ONE6,0,4);
ONEPOINT:=POINTER;
ONESOUND(ONETWO6,1,16);
TWOPOINT:=POINTER;
ONESOUND(ONETWO36,2,64);
ONESOUND(ONETWO346,3,256);
FOURPOINT:=POINTER;
POINTER:=0;
```

```
ONESOUND(E6,Z,4);
SPACE:=ONEPOINT*4+POINTER;
FIRSTLAST6(SPACE,NUMBER):=FIRSTLAST6(SPACE,NUMBER)+1;
FIRSTLAST6(16,NUMBER):=FIRSTLAST6(16,NUMBER)+1;
FIRSTLAST6(SPACE,0):=1;
SPACE:=FOURPOINT*4+POINTER;
ONETWO34E6(SPACE,NUMBER):=ONETWO34E6(SPACE,NUMBER)+1;
ONETWO34E6(SPACE,0):=1;
ONETWO34E6(1024,NUMBER):=ONETWO34E6(1024,NUMBER)+1;
SPACE:=Z-1;
ONESOUND(FE26,SPACE,16);
SPACE:=FOURPOINT*16+POINTER;
ALLSOUND(SPACE,NUMBER):=ALLSOUND(SPACE,NUMBER)+1;
ALLSOUND(SPACE,0):=1;
ALLSOUND(4096,NUMBER):=ALLSOUND(4096,NUMBER)+1;
SPACE:=TWOPOINT*16+POINTER;
F12EE26(SPACE,NUMBER):=F12EE26(SPACE,NUMBER)+1;
F12EE26(SPACE,0):=1;
F12EE26(256,NUMBER):=F12EE26(256,NUMBER)+1
END;


END;
END;
END RAW_ANALYSIS;
```

```
COMMENT*************************************************************
PROCEDURE COMPUTE_ANY IS EXECUTED ONCE FOR EACH ARRAY
CONTAINING FREQUENCIES PER LANGUAGE.  THE LAST ROW OF EACH
DATA ARRAY CONTAINS THE TOTAL NUMBER OF WORDS WHICH HAVE
FREQUENCY DATA STORED IN THE CORRESPONDING COLUMN.  THIS
TOTAL IS DIVIDED INTO EACH ENTRY IN THE COLUMN GIVING
A PERCENTAGE OF OCCURRANCE FOR THAT SOUND COMBINATION
WITHIN THE GIVEN PARTITION.
THE CALLING ARGUEMENTS ARE:
1).  INTEGER ITERATION:  THE SUBSCRIPT OF THE LAST ROW IN
THE ARRAY WHICH HAS FREQUENCY INFORMATION IN IT.  IN
GENERAL IT ROW DIMENSION OF THE ARRAY BEING PASSED IN LESS 1
2)  RAW_NUMBERS:  THE ARRAY BEING NORMALIZED.
***************************************************************;

PROCEDURE COMPUTE_ANY(INTEGER VALUE ITERATION;REAL ARRAY
RAW_NUMBERS(*,*));
BEGIN
INTEGER T;
T := ITERATION+1;
IF(RAW_NUMBERS(T,NUMBER)¬=0) THEN
BEGIN
FOR K := 0 UNTIL ITERATION DO
RAW_NUMBERS(K,NUMBER):=RAW_NUMBERS(K,NUMBER)/RAW_NUMBERS
(T,NUMBER)
END;
END COMPUTE_ANY;
```

```
COMMENT*****************************************************************
SINCE ALGOL-W HAS NO EQUIVALENT TO A FORTRAN DATA STATEMENT,
ARRAYS MUST BE INITIALIZED TO THE DESIRED VALUE.  THIS IS
DONE ONE COLUMN AT A TIME.  "ZIP" IS THE VALUE GIVEN TO EACH
ARRAY ELEMENT.  COLUMN ZERO IN EACH ARRAY IS SET TO MINUS
ONE. THIS VALUE IS CHANGED TO PLUS ONE WHEN AN ENTRY OF
FREQUENCY INFORMATION IS MADE IN THAT PARTICULAT ROW.  BY
CHECKING COLUMN ZERO IN LATER PARTS OF THE PROGRAM,
DIVISION BY ZERO IS ELIMINATED.
*******************************************************************;

PROCEDURE CLEARMEM(REAL VALUE ZIP);
BEGIN
INTEGER J,K,L,R,N;
J:=K:=L:=R:=N:=0;
FOR I := 0 UNTIL 3 DO
BEGIN
ONE1(I,NUMBER):=ZIP;
ONE2(I,NUMBER):=ZIP;
ONE3(I,NUMBER):=ZIP;
ONE4(I,NUMBER):=ZIP;
ONE5(I,NUMBER):=ZIP;
ONE6(I,NUMBER):=ZIP;
E6(I,NUMBER):=ZIP;
FOR P := 0 UNTIL 3 DO
BEGIN
ONETWO2(J,NUMBER):=ZIP;
ONETWO3(J,NUMBER):=ZIP;
ONETWO4(J,NUMBER):=ZIP;
ONETWO5(J,NUMBER):=ZIP;
ONETWO6(J,NUMBER):=ZIP;
FIRSTLAST5(J,NUMBER):=ZIP;
FIRSTLAST6(J,NUMBER):=ZIP;
EE26(J,NUMBER):=ZIP;
J:=J+1;
FOR S := 0 UNTIL 3 DO
BEGIN
ONETWO33(K,NUMBER):=ZIP;
ONETWO34(K,NUMBER):=ZIP;
ONETWO35(K,NUMBER):=ZIP;
ONETWO36(K,NUMBER):=ZIP;
K:=K+1;
FOR T := 0 UNTIL 3 DO
BEGIN
ONETWO344(L,NUMBER):=ZIP;
ONETWO345(L,NUMBER):=ZIP;
ONETWO346(L,NUMBER):=ZIP;
F12EE26(L,NUMBER):=ZIP;
L:=L+1;
FOR M := 0 UNTIL 3 DO
BEGIN
ONETWO34E6(R,NUMBER):=ZIP;
ONETWO34E5(R,NUMBER):=ZIP;
R:=R+1;
FOR O := 0 UNTIL 3 DO
BEGIN
ALLSOUND  (N,NUMBER):=ZIP;
N:=N+1
END;
END;
END;
END;
END;
END;
END CLEARMEM;
```

```
COMMENT*****************************************************
PROCEDURE FINDSOUND IS CALLED FROM "DECIDE" AND RETURNS A
VALUE OF "POINTER" CORRESPONDING TO THE SUBSCRIPT  OF THE
SOUND IN THE ARRAY "CLASS."  BECAUSE ALL THE SOUND
COMBINATIONS ARE POWERS OF FOUR, IT IS A SIMPLE MATTER TO
COMPUTE SAY A TWO-SOUND COMBINATION BY REMEMBERING WHAT THE
FIRST SOUND WAS, MULTIPLYING THE FIRST SOUND SUBSCRIPT BY 4
AND ADDING THE SECOND SOUND SUBSCRIPT.  THE BASIC IDEA OF
MODULO 4 ARITHMETIC.  NO MATTER WHAT SOUND COMBINATION IS
THE OBJECTIVE OF THE SEARCH, THE PROGRAM ONLY SEARCHES A
LIST OF FOUR ELEMENTS.  "WS" TELL "FINDSOUND" WHICH SOUND
IN THE WORD TO LOOK AT.
***********************************************************;

PROCEDURE FINDSOUND(INTEGER VALUE WS);
BEGIN
POINTER:=0;
WHILE(CODE(WS|1)¬=CLASS(POINTER)) DO
POINTER:=POINTER+1;
END FINDSOUND;


COMMENT*****************************************************
PROCEDURE ONESOUND IS CALLED FROM RAW ANALYSIS.  IT
OPERATES ON THE SAME PRINCIPLE AS "FINDSOUND" EXCEPT THAT IT
ALSO INCREMENTS THE FREQUENCY COUNT IN THE APPROPRIATE ARRAY
"OUTPUT" AND INCREMENTS THE TOTAL WORD COUNTER WITHIN
"OUTPUT."  THE RAW OF "OUTPUT" WHICH CONTAINS THE WORD
COUNTER MUST BE SPECIFIED BY THE ARGUENENT "LENGTH."
***********************************************************;

PROCEDURE ONESOUND(REAL ARRAY OUTPUT(*,*):INTEGER VALUE
WS,LENGTH);
BEGIN
INTEGER PIN;
PIN := POINTER;POINTER:=0;
WHILE(CODE(WS|1)¬=CLASS(POINTER))DO
POINTER:=POINTER+1;
POINTER:=PIN*4+POINTER;
OUTPUT(POINTER,NUMBER):=OUTPUT(POINTER,NUMBER)+1;
OUTPUT(LENGTH,NUMBER):=OUTPUT(LENGTH,NUMBER)+1;
OUTPUT(POINTER,0):=1
END ONESOUND;
```

```
COMMENT****************************************************************
PROCEDURE SINGLE_PROBS COMPUTES THE ACTUAL PROBABILITY OF A
LANGUAGE GIVEN ANY SINGLE SOUND WHICH THE PROGRAM LOOKS AT
AS A SEPARATE ENTITY.  I.E., THE FIRST SOUND OF EACH WORD
AND THE LAST SOUND OF WORDS SIX SOUND AND GREATER IN LENGTH.
THE ASSUMPTION IS MADE THAT ALL LANGUAGES ARE EQUALLY
PROBABLE.
****************************************************************;

PROCEDURE SINGLE_PROBS;
BEGIN
REAL B,C,D,E,F,G,H;
FOR I := 0 UNTIL 3 DO
BEGIN
B:=C:=D:=E:=F:=G:=H:=0;
FOR J := 1 UNTIL NUMBER DO
BEGIN
B := B+ONE1(I,J);
C := C+ ONE2(I,J);
D := D+ ONE3(I,J);
E := E+ ONE4(I,J);
F := F+ ONE5(I,J);
G := G+ ONE6(I,J);
H:=H+E6(I,J)
END;
FOR K := 1 UNTIL NUMBER DO
BEGIN
IF(B¬=0) THEN
ONE1(I,K):=ONE1(I,K)/B   ;
IF(C¬=0) THEN
ONE2(I,K):=ONE2(I,K)/C   ;
IF(D¬=0) THEN
ONE3(I,K):=ONE3(I,K)/D   ;
IF(E¬=0) THEN
ONE4(I,K):=ONE4(I,K)/E    ;
IF(F¬=0) THEN
ONE5(I,K):=ONE5(I,K)/F       ;
IF(G¬=0) THEN
ONE6(I,K):=ONE6(I,K)/G   ;
IF(H¬=0) THEN
E6(I,K):=E6(I,K)/H;
END;
END;
END SINGLE_PROBS;
```

```
COMMENT***************************************************************
PROCEDURE GRAPH PROVIDES A CONVENIENT GRAPHICAL
REPRESENTATION OF ANY DATA ARRAY USED AT ANY POINT IN THE
PROGRAM.  THE CALLING ELEMENTS ARE:
1)  DATA:  THE ARRAY TO BE GRAPHED.
2)  LABEL:  THE STRING ARRAY WITH LABELS CORRESPONDING TO
DATA.
3)  COUNT:  THE LAST ROW SUBSCRIPT OF "DATA" WHICH CONTAINS
INFORMATION TO BE GRAPHED, I,E., ROW DIMENSION OF DATA LESS
ONE.
4)  SCALE:  NUMBER OF PERCENTAGE POINTS PER PRINT SPACE,
EITHER 1 OR 2.
5)  HEADING:  10 IF GRAPH IS TO SHOW RANGE OF ZERO TO ONE
HUNDRED.  5 IF RANGE IS ZERO TO FIFTY.
6)  WIDTH:  WIDTH OF GRAPH PRINT SPACE, 50 OR 100.
7)  TOGGLE:  COLUMN TO CHECK TO SEE IF THAT ROW CONTAINS
ANY NUMBER OTHER THAN ZERO.  IS SET TO ZERO FOR ALL CASES.

GRAPH WILL PRINT A LETTER CORRESPONDING TO THE ORDER OF THE
BASE DATA SETS.  A = LANGUAGE 1, B = LANGUAGE 2, ETC.
IF TWO VALUES ARE IDENTICAL ON A LINE OF THE GRAPH "*" IS
PRINTED.  IF A WIDTH OF 50 AND SCALE OF "1" IS SELECTED,
NORMALIZED SETS OF DATA CAN BE REPRESENTED BECAUSE THE
VALUE WHICH IS GREATER THAN 50 WILL BE PRINTED IN THE RIGHT
MARGIN OF THE GRAPH.
*****************************************************************;

PROCEDURE GRAPH (REAL ARRAY DATA(*,*);
STRING(6) ARRAY LABEL(*);
INTEGER VALUE COUNT,SCALE,HEADING ,WIDTH,TOGGLE);
BEGIN
STRING(1) OVERFLOW;
INTEGER AMT,BIG,LINES;
LOGICAL TOO_BIG;
STRING(17) FLAG;
STRING(1) ARRAY MATRIX (0::WIDTH);

PROCEDURE HORIZ_LINE;
BEGIN
IOCONTROL(2); WRITE("         ");
FOR I :=0 UNTIL WIDTH DO
BEGIN
IF(I REM 5 = 0 ) THEN
WRITEON("+") ELSE WRITEON("-")  END;
END HORIZ_LINE;

PROCEDURE HORIZ_HEAD;
BEGIN
INTEGER P;
WRITE(" ");
IOCONTROL(2); WRITE("              ");
IF(WIDTH=50) THEN INTFIELDSIZE:=3 ELSE INTFIELDSIZE:=8;
FOR I := 1 UNTIL 10 DO
BEGIN
P := I*HEADING ;  WRITEON(P)
END;
END HORIZ_HEAD;

FLAG:=" ABCDEFGHIJKLMNPR";
HORIZ_HEAD;
HORIZ_LINE;
LINES:=0;
FOR I := 0 UNTIL COUNT DO
BEGIN
TOO_BIG:=FALSE;
OVERFLOW:=" ";
IF(DATA(I,TOGGLE)¬=-1) THEN
BEGIN
LINES:=LINES+1;
FOR J := 0 UNTIL WIDTH DO
BEGIN
MATRIX(J) := " ";
```

```
FOR K := 1 UNTIL NUMBER DO
BEGIN
AMT:=ROUND((DATA(I,K)*100)/SCALE);
IF(J=WIDTH)AND(AMT>J) THEN      BEGIN
TOO_BIG:=TRUE;
OVERFLOW(0|1):= FLAG(K|1);
BIG:=AMT
END;
IF(AMT=J) THEN
BEGIN
IF(MATRIX(J)¬=" ") THEN
MATRIX(J) := "*"
ELSE  MATRIX(J) := FLAG(K|1)
END;
END;
END;
IOCONTROL(2);
WRITE(LABEL(I),"|");
FOR H := 0 UNTIL WIDTH DO
BEGIN
WRITEON(MATRIX(H))
END;
WRITEON("|");
INTFIELDSIZE:=3;
IF(TOO_BIG=TRUE) THEN  WRITEON(OVERFLOW,BIG) ;
IF(COUNT<16) THEN
BEGIN
WRITE("      |");
FOR T := 0 UNTIL WIDTH DO  WRITEON(" "); WRITEON("|")
END;
END;
IF(LINES=50) THEN
BEGIN
LINES:=0;
HORIZ_LINE;
HORIZ_HEAD;
IOCONTROL(3);  WRITE("CONTINUED");
WRITE(" ");
HORIZ_HEAD;
HORIZ_LINE
END;
END;
IOCONTROL(2);
HORIZ_LINE;
HORIZ_HEAD ;
WRITE(" "); WRITE("SAMPLE SIZES ARE :");
AMT := COUNT+1;
IOCONTROL(2);
FOR I := 1 UNTIL NUMBER DO WRITEON(DATA(AMT,I))
END GRAPH;
```

```
COMMENT**************************************************************
MAIN PROGRAM:  GREATER FLEXIBILITY WAS REALIZED BY
ENCLOSING THE ENTIRE PROGRAM IN THE "DUMMY" PROCEDURE.
THIS ALLOWS THE PROGRAMMER TO SPECIFY, WITH THE VALUE OF
"DIM" HOW MANY LANGUAGES ARE TO BE LOADED AS BASE DATA.
*****************************************************************;

CLASS(0):= " ":
CLASS(0)(0|1):="F";
CLASS(1):=" ";
CLASS(1)(0|1):="N";
CLASS(2):=" ";
CLASS(2)(0|1):="S";
CLASS(3):=" ";
CLASS(3)(0|1):="V";

GENERATETWO;
GENERATETHREE;
GENERATEFOUR:
GENERATESIX:

GETARRAY;

SETCOUNT;

NUMBER := 0;
CLEARMEM(-1);
N := 1;


WHILE(N>0) AND(NUMBER<DIM) DO
BEGIN
NUMBER := NUMBER + 1 ;

INITIALIZE;
CLEARMEM(0);


WHILE(GETWORD(WORD)) DO
BEGIN
CONVERT(CODE) :
RAW_ANALYSIS
END;

COMPUTE_NORM;

READ(N)
END;
```

```
GRAPH(ONE1  ,CLASS,3,2,10,50,0);
GRAPH(ONE2,CLASS,3,1, 5,50,0);
GRAPH(ONE3,CLASS,3,1, 5,50,0);
GRAPH(ONE4,CLASS,3,1, 5,50,0);
IOCONTROL(3);
GRAPH(ONE5  ,CLASS,3,1, 5,50,0);
GRAPH(ONE6  ,CLASS,3,1, 5,50,0);
GRAPH(E6   ,CLASS,3,1, 5,50,0);
IOCONTROL(3);
GRAPH(ONETWO2       ,SOUNDTWO, 15, 1,  5, 50, 0 );
GRAPH(ONETWO3       ,SOUNDTWO, 15, 1,  5, 50, 0 );
IOCONTROL(3);
GRAPH(ONETWO4       ,SOUNDTWO, 15, 1,  5, 50, 0 );
GRAPH(ONETWO5       ,SOUNDTWO, 15, 1,  5, 50, 0 );
IOCONTROL(3);
GRAPH(ONETWO6       ,SOUNDTWO, 15, 1,  5, 50, 0 );
GRAPH(FIRSTLAST5    ,SOUNDTWO, 15, 1,  5, 50, 0 );
IOCONTROL(3);
GRAPH(FIRSTLAST6    ,SOUNDTWO, 15, 1,  5, 50, 0 ):
GRAPH(EE26          ,SOUNDTWO,15,1, 5,50,0):
IOCONTROL (3);
GRAPH (ONETWO33      ,SOUNDTHREE , 63, 1,  5, 50, 0);
GRAPH (ONETWO34      ,SOUNDTHREE , 63, 1,  5, 50, 0);
IOCONTROL(3);
GRAPH (ONETWO35      ,SOUNDTHREE , 63, 1,  5, 50, 0);
GRAPH (ONETWO36      ,SOUNDTHREE , 63, 1,  5, 50, 0):
IOCONTROL(3);
GRAPH (ONETWO344     ,SOUNDFOUR , 255, 1,  5, 50, 0);
IOCONTROL(3):
GRAPH (ONETWO345     ,SOUNDFOUR , 255, 1,  5, 50, 0);
IOCONTROL(3);
GRAPH (ONETWO346     ,SOUNDFOUR , 255, 1,  5, 50, 0):
IOCONTROL(3);
GRAPH (F12EE26       ,SOUNDFOUR , 255, 1,  5, 50, 0);
IOCONTROL(3);
GRAPH (ONETWO34E5,SOUNDFIVE ,1023 ,1  , 5 ,50 ,0 );
IOCONTROL(3);
GRAPH (ONETWO34E6,SOUNDFIVE ,1023 ,1  , 5 ,50 ,0 );
IOCONTROL(3):
GRAPH (ALLSOUND ,SOUNDSIX ,4095 ,1 , 5 ,50 ,0 );
```

```
SINGLE_PROBS:
PROB_TWO(ONETWO2, ONE2, ONETWO2, 3, 3);
PROB_TWO(ONETWO3, ONE3, ONETWO3, 3, 3);
PROB_TWO(ONETWO33,ONETWO3,ONETWO33,15,3);
PROB_TWO(ONETWO4, ONE4, ONETWO4, 3, 3);
PROB_TWO(ONETWO34, ONETWO4, ONETWO34, 15, 3);
PROB_TWO(ONETWO344, ONETWO34, ONETWO344, 63, 3);
PROB_TWO(ONETWO5, ONE5, ONETWO5, 3, 3);
PROB_TWO(ONETWO35, ONETWO5, ONETWO35, 15, 3);
PROB_TWO(ONETWO345, ONETWO35, ONETWO345, 63, 3);
PROB_TWO(ONETWO34E5, ONETWO345, ONETWO34E5, 255, 3);
PROB_TWO(FIRSTLAST5, ONE5, FIRSTLAST5, 3, 3);
PROB_TWO(ONETWO6, ONE6, ONETWO6, 3, 3);
PROB_TWO(ONETWO36, ONETWO6, ONETWO36, 15, 3);
PROB_TWO(ONETWO346, ONETWO36, ONETWO346, 63, 3);
PROB_TWO(ONETWO34E6, ONETWO346, ONETWO34E6, 255, 3);
PROB_TWO(EE26, E6, EE26, 3, 3);
PROB_TWO(F12EE26,ONETWO6,F12EE26,15,15);
PROB_TWO(ALLSOUND,ONETWO346,ALLSOUND,255,15);
PROB_TWO(FIRSTLAST6, ONE6, FIRSTLAST6, 3, 3);
```

```
GRAPH(ONE1  ,CLASS,3,1, 5,50,0);
GRAPH(ONE2,CLASS,3,1, 5,50,0);
GRAPH(ONE3,CLASS,3,1, 5,50,0);
GRAPH(ONE4,CLASS,3,1, 5,50,0);
IOCONTROL(3);
GRAPH(ONE5  ,CLASS,3,1, 5,50,0);
GRAPH(ONE6  ,CLASS,3,1, 5,50,0);
GRAPH(E6   ,CLASS,3,1, 5,50,0);
IOCONTROL(3);
GRAPH(ONETWO2      ,SOUNDTWO, 15, 1,  5, 50, 0 );
GRAPH(ONETWO3      ,SOUNDTWO, 15, 1,  5, 50, 0 );
IOCONTROL(3);
GRAPH(ONETWO4      ,SOUNDTWO, 15, 1,  5, 50, 0 );
GRAPH(ONETWO5      ,SOUNDTWO, 15, 1,  5, 50, 0 );
IOCONTROL(3);
GRAPH(ONETWO6      ,SOUNDTWO, 15, 1,  5, 50, 0 );
GRAPH(FIRSTLAST5   ,SOUNDTWO, 15, 1,  5, 50, 0 );
IOCONTROL(3);
GRAPH(FIRSTLAST6   ,SOUNDTWO, 15, 1,  5, 50, 0 );
GRAPH(EE26         ,SOUNDTWO,15,1, 5,50,0);
IOCONTROL(3);
GRAPH (ONETWO33     ,SOUNDTHREE , 63, 1,  5, 50, 0);
GRAPH (ONETWO34     ,SOUNDTHREE , 63, 1,  5, 50, 0);
IOCONTROL(3);
GRAPH (ONETWO35     ,SOUNDTHREE , 63, 1,  5, 50, 0);
GRAPH (ONETWO36     ,SOUNDTHREE , 63, 1,  5, 50, 0);
IOCONTROL(3);
GRAPH (ONETWO344    ,SOUNDFOUR , 255, 1,  5, 50, 0);
IOCONTROL(3);
GRAPH (ONETWO345    ,SOUNDFOUR , 255, 1,  5, 50, 0);
IOCONTROL(3);
GRAPH (ONETWO346    ,SOUNDFOUR , 255, 1,  5, 50, 0);
IOCONTROL(3);
GRAPH (F12EE26      ,SOUNDFOUR , 255, 1,  5, 50, 0);
IOCONTROL(3);
GRAPH (ONETWO34E5,SOUNDFIVE ,1023 ,1 , 5 ,50 ,0 );
IOCONTROL(3);
GRAPH (ONETWO34E6,SOUNDFIVE ,1023 ,1 , 5 ,50 ,0 );
IOCONTROL(3);
GRAPH (ALLSOUND ,SOUNDSIX ,4095 ,1 , 5 ,50 ,0 );
IOCONTROL(3);
```

```
READ(SAMPLESIZE);
N := 0  ;

WHILE(SAMPLESIZE¬=-1) DO
BEGIN
TP := 80;
DECIDE;
IOCONTROL(3);
READ(SAMPLESIZE)
END;

END;


READ(DIM);
DUMMY
END.
```

```
COMMENT*****************************************************
THE DATA DECK IS TO BE STACKED AS FOLLOWS:
1).  CARD WITH THE NUMBER OF LANGUAGES FOR BASE DATA
ALANYSIS.
2).  ROMI.
3).  CARD WITH THE NUMBER OF CARDS WHICH CONTAIN
IDENTIFYING INFORMATION ABOUT THE BASE DATA ELEMENTS.
THIS IS ASSIGNED AS HEADCOUNT.
4).  DATA CARDS CONTAINING ROMI REPRESENTATIONS OF THE IPA
SPELLING OF WORDS.  A WORD CANNOT BE STARTED ON ONE CARD AND
CONTINUED ON THE NEXT.
5).  CARD WITH "%" IN COLUMN 9 TO SIGNAL END OF DATA BASE
SET.
6).  CARD WITH -1, IF NO MORE BASE DATA SETS FOLLOW.  ANY
NUMBER OTHER THAN ZERO WILL SIGNAL THAT  ANOTHER SET
WILL FOLLOW.  SUGGESTED SEQUENCE IS  ONE THROUGH FOUR WITH A
ZERO FOLLOWING THE FINAL BASE DATA SET, THEN THEN NUMBERS
ONE THROUGH "N" FOR THE TEST DATA SAMPLES, WITH A -1
FOLLOWING THE FINAL TEST DATA SET.
************************************************************;

COMMENT*****************************************************
ROMI MUST ALWAYS BE THE FIRST DATA SET FOLLOWING THE "%DATA"
CARD (JCL).  THE NUMBER "5" SHOWN BELOW TELL HOW MANY SETS
OF REFERENCE DATA WILL BE USED IN THE INITIAL SETUP.
NOTE THAT ROMI IS WRITTEN SUCH THAT EACH CHARACTER OF ROMI
IS FOLLOWED IMMEDIATELY (SEPARATED BY ONE SPACE) BY THE
CATEGORY TO WHICH IT BELONGS.  TWO LETTERS JOINED BY A "-"
ARE READ AS A SINGLE LETTER.
************************************************************;
```

```
%DATA
          5
A  V      A-A  V    A-B  V    A-E  V    A-I  V    A-N  F    A-R  V    A-U  V
B  S      B-B  F    B-V  S
C  V      C-C  S    C-F  V    C-H  F    C-I  V    C-Q  V
D  S      D-D  S    D-Z  S
E  V      E-E  V    E-I  V    E-Q  V    E-R  V    E-S  V
F  F      F-F  S
G  S      G-A  V    G-B  F    G-F  F    G-G  S
H  F      H-H  F    H-W  N
I  V      I-I  V    I-R  V    I-T  V    I-U  V
J  N      J-A  F    J-N  N    J-U  V
K  S      K-X  S    K-Z  F
L  N      L-L  N
M  N      M-J  N
N  N      N-J  N    N-N  N    N-O  N    N-U  S
O  V      O-F  V    O-O  V    O-U  V
P  S      P-F  S
Q  F      Q-A  F    Q-Q  S
R  N      R-B  F    R-R  N    R-T  N
S  F      S-H  F    S-S  F
T  S      T-C  S    T-H  F    T-L  S    T-M  S    T-S  S    T-T  S
U  V      U-H  V    U-Q  V    U-R  F    U-S  V    U-T  V    U-U  V
V  F      V-V  V
W  N      W-W  V
X  F      X-X  F
Y  V      Y-U  N    Y-Y  V
Z  F      Z-C  F    Z-H  F    Z-Z  F
3-R  V
```

```
%DATA
     5
A  V    A-A  V    A-B  V    A-E  V    A-I  V    A-N  F    A-R  V    A-U  V
B  S    B-B  F    B-V  S
C  V    C-C  S    C-E  V    C-H  F    C-I  V    C-Q  V
D  S    D-D  S    D-Z  S
E  V    E-E  V    E-I  V    E-Q  V    E-R  V    E-S  V
F  F    F-F  S
G  S    G-A  V    G-B  F    G-F  F    G-G  S
H  F    H-H  F    H-W  N
I  V    I-I  V    I-R  V    I-T  V    I-U  V
J  N    J-A  F    J-N  N    J-U  V
K  S    K-X  S    K-Z  F
L  N    L-L  N
M  N    M-J  N
N  N    N-J  N    N-N  N    N-Q  N    N-U  S
O  V    O-E  V    O-O  V    O-U  V
P  S    P-F  S
Q  F    Q-A  F    Q-Q  S
R  N    R-B  F    R-R  N    R-T  N
S  F    S-H  F    S-S  F
T  S    T-C  S    T-H  F    T-L  S    T-M  S    T-S  S    T-T  S
U  V    U-H  V    U-Q  V    U-R  F    U-S  V    U-T  V    U-U  V
V  F    V-V  V
W  N    W-W  V
X  F    X-X  F
Y  V    Y-U  N    Y-Y  V
Z  F    Z-C  F    Z-H  F    Z-Z  F
3-R  V
```

LIST OF REFERENCES

1. UPI article printed in *Monterey Pennsula Herald*, December 28, 1975.

2. Rosenblum, H.,"*Voice Systems*," presentation at the Naval Postgraduate School, Monterey, California, 14 October 1975.

3. Potter, R. K., Kopp, G. A., and Kopp, H. G., *Visible Speech*, Dover Publications, Inc., New York, 1966.

4. "program of the 90th Meeting." *The Journal of the Acoustical Society of America*, vol. 58, supplement no. 1, Fall 1975.

5. The Principles of the I. P. A., 1949

6. Cartier, Francis A. and Todaro, Martin T., *The Phonetic Alphabet*, Wm. C. Brown Company Publishers, Dubuque, Iowa, 1971.

7. Discussion with Dr. Cartier on 1/10/76.

8. Carterette, Edward C. and Jones, Margaret H., *Informal Speech, Alphabetic and Phonemic text With Statistical Analyses and Tables*, University of California Press, Berkeley, California, 1974.

9. Kenyon, John S. and Knott, Thomas A., *A Pronouncing Dictionary of American English*, G. & C. Merriam Company, Publishers, Springfield, Massachusetts, 1944.

10. Hess, Wolfgang J., "A Pitch/Synchronous Digital Feature Extraction System for Phonemic Recognition of Speech," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, volume ASSP-24, no. 1, February 1976.

11. Personal interview with various language specialists were conducted, as required, at the Defense Language Institute, The Presido, Monterey, California 93940. Commendable assistance was received from the Research Department.

12. Personal interwiews were conducted for the purpose of achieving the proper pronunciation of words selected for transcription. Extremely useful interviews were conducted with Dr. Maria Baird of DLI, and LCDR Charles A. Pulfrey of the Naval Postgraduate School abid.

13. Rau, Morton David, *Language Identification by Statistical Methods*, M. S. Thesis, United States Naval Postgraduate School, Monterey, California, September 1974.

14. Personal interview with Dr. Thomas Arkwright, computer processing consultant, Defense Language Institute, Presidio, Monterey, California 93940

15. Thomas, Michael C., several discussions at length on the facilities available and capable of processing phonemic analysis. Lt. Thomas is a student at the Naval Postgraduate school, Monterey, California.

16. Hancock, John C. and Wintz, Paul A., Signal Detection Theory, McGraw Hill Book Company, New York, 1966.

17. Lipschutz, Seymour, Schaum's Outline of Theory and Problems of Probability, Table 6.1, McGraw-Hill Book Company, New York, New York, 1965.

18. Flanagan, J. L., Speech Analysis, Synthesis, and Perception, 2nd ed., Springer-Verlag, Berlin, Germany, 1972.

19. Doherty, Thomas E., Evaluation of Selected Acoustic Parameters for use in Speaker Identification, paper presented at 90th meeting of the Acoustic Society of America (see ref. 14), San Francisco, California, 7 November 1975.

20. Young, R. B., Investigation of Speaker Identification Based on Nasal Phonotation, M. S. Thesis, Naval Postgraduate School, Monterey, California, June 1971.

21. Wohlford, Robert E., personnal discussion at length, on various subjects including and specifically on the occurances and identification of pauses and word boundries, January 19, 1976.

22. Flanagan, James L. and Rabiner, L. R., Speech Synthesis, Dowden, Hutchinson, & Ross, Inc., Stroudsburg, Pennsylvania, 1973.

23. Koenig, W., Dunn, H. K., and Lacy, L. W., "The Sound Spectrograph," Journal of the Acoustic Society of America, (JASA), 18, 19-49, 1946.

24. Lehiste, Ilse, Readings in Acoustic Phonetics, The M. I. T. Press, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1967.

25. Chao, Yuen Ren. Language and Symbolic Systems, Cambridge at the University Press, London, England, 1968.

26. Judson, L. S. and Weaver, A. T., Voice Science, F. S. Crofts & Co., New York, 1942.

27. United States Air Force Cambridge Research Labratories Report AFCRL-69-0371, Speech Analysis, by Arthur S. House and George W. Hughes, September 1969.

28. ROME Air Development Center Report AD-758-397, Automatic Language Identification, by Gary R. Leonard, et al, August 1974.

29. Carrell, J. and Tiffany W. R., Phonetics: Theory and Application to Speech Improvement, McGraw Hill Book Company, Inc., New York, 1960.

30. Fletcher, H., Speech and Hearing in Communication, D. Van Nostrand Company, Inc., Princeton, New Jersey, 1958.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	2
4.	Professor Stephen Jauregui, Code 33, Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
5.	Asst. Professor Alan R. Washburn, Code 55WA, Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
6.	LT Michael C. Thomas, USN 1304 Fechteler Drive Monterey, California 93940	1
7.	LT Roger D. Cook, USN 1333 Spruance Road Monterey, California 93940	1

8. Director 1
National Security Agency
Attn: R, H. Rosenblum
R-5, Dr. C. Wood
Ft. Meade, Maryland 20755

9. Commander 1
Naval Security Group Command
Naval Security Group Command Headquarters
Attn: Commander H. Shoemaker
3801 Nebraska Avenue
Washington, D. C. 20390

10. Defense Language Institute 1
Presidio
Attn: DLIF-1-CS, Dr. F. Cartier
Monterey, California 93940