capable of enabling students to function at level three. More likely, traditional materials are, to a significant degree, an important component to functioning at level three, but additional training materials in text and discourse processing are required. If the former is the case, there is probably little reason to automate traditional activities. If the latter is the case, portions of both traditional and innovative activities should be considered for CAI.

From DOD's experience, cloze type activities for both reading and listening appear to be very valuable in teaching students some of the skills necessary for level three. However, in order to implement such activities on a computer it is necessary to display more text than is possible on a 24 line by 40 character screen. As Wyatt points out, an 80 character screen suffers from resolution problems, and in order to prevent a crowding of lines of text it is necessary to print on every other line, giving the user only twelve lines of text. Many micros are incapable of displaying at least a half typewritten page. Scrolling materials on the screen would probably tax students' short-term memory. While an accompanying hard copy text could be used for other activities, it is just as easy to access text files on line, especially if the system is capable of displaying more than one window with significant amounts of text. Trying to coordinate hard copy with software may turn out to be more trouble than it is worth, the more so if students must also handle various floppy discs.

Some systems now have the capability of displaying more than one window on the screen simultaneously. Such systems can display two full pages or many partial pages with excellent resolution. With multiple window display for cloze and other reading activities, students could access lists of deleted items or on-line reference materials. Such windowing capability should be exploited in future reading activities.

The issue of transportability was raised early in Wyatt's paper with concern being focused on the problem of transporting software from mainframe to micro. While this is significant, the more appropriate point to raise is the issue of the limited memory available on most microcomputers. For example, while one can extract short modules or lessons from a larger mainframe CAI system, one sacrifices the tremendous capacity for record keeping, student tracking, and general CMI available on systems such as Pilot/Superpilot. The need for loading lessons on to floppies is a direct result of the limited memory/storage space on which Pilot operates. Separating the course materials into smaller modules and storing them on floppies, in order to run on micros, requires the instructor to handle a lot of discs for each student and workstation even if the micros are netted together. The present capabilities of authoring languages such as Superpilot allow for record keeping on each floppy, but the instructor must collect discs and feed them into a drive to gather the relevant data. This process is akin to collecting machine scorable homework assignments or test papers. The amount of time required for a disc drive to read a dozen or so floppies is not trivial and someone must be present to load and unload the floppies.

Another equipment-related issue is the ability to support various orthographies on one system. Assuming that a language department wanted to contruct a CALL lab, it would be desirable to have a standard system (set of hardware and software) supporting the various required exotic orthographies. The computer can provide a medium for learning non-Roman scripts at the beginning level by introducing individual graphemes and words. However, text processing, desirable and sometimes necessary for creation of longer text materials, is difficult on most systems. It would be an advantage for one system to have generic word processing for all orthographies. It should be noted that the foreign scripts are so stylized on some small systems that even native speakers have difficulty in reading their own orthography. Such systems have *ad hoc* modifications, and only English and one particular language can operate on them. It would be a logistic nightmare to require computer support for ten Cyrillic/English workstations from one set of vendors, six Chinese/English workstations from another, and eight Farsi/English workstations from yet another.

These requirements for sufficient text, centrally stored lesson files, and multilingual capabilities require some type of integrated architecture with distributed processing whereby an intelligent workstation could download and copy a lesson from a CPU and operate as a stand-alone, yet when the lesson was completed, the CMI and record keeping could be directly stored at the CPU. It is difficult to believe that a significant dictionary could be stored in the limited memory of a microcomputer to enable any globally useful facilitative activity. An integrated architecture could more likely support this activity, not only eliminating the need for floppies, but also allowing for students to access a usable on-line dictionary.

One man-machine interface issue which has not received much discussion, although raised in part

by Wyatt, is the
the one hand the
Roman character
text materials fo
tical method to d
be developed. O
dent's point of v
drance in the
keyboard interfe
students to enter
keyed to an answ
form of multiple
one of the answer
in the curriculum
productively and c
own. This require
through a keyboa
cent hand-entry de
interfaced to a co
approximately thr
characters and d
stored representa
alphabetic sets of t
be easy for the sys
ty is tantamount
and would elimina
the student as well

At this stage o
audio device such
vide high quality a
accurate accessing
difficult to access a
and the access time
feature virtually
capabilities of ana
speech at this stage
is extremely costly.
sample rate to deli
amount of memory
for language learn
of 50-10K Hz. Unli
who can do with
distinguish fricativ
of a foreign langua
tion of the audible
phonological contr
have the ability to s
tion. However, wha
synthesis is present
foreign language te
on micros at le
Phonological contra
suprasegmental in
especially at senten
There may be a t

ppies, in
ructor to
worksta-
. The pre-
s such as
ach flop-
and feed
lata. This
scorable
e amount
l a dozen
must be

ability to
system.
vanted to
le to have
software)
otic or-
medium
eginning
mes and
able and
nger text
would be
eric word
be noted
me small
difficulty
systems
glish and
them. It
computer
ons from
worksta-
i/English

centrally
oilities re-
ture with
ntelligent
a lesson
yet when
id record
PU. It is
lictionary
ory of a
ly useful
chitecture
not only
so allow-
on-line

h has not
d in part

by Wyatt, is the interaction with the keyboard. On the one hand the instructor will need to access non-Roman character sets via the keyboard to compose text materials for courseware. If there is no practical method to do this, very little courseware will be developed. On the other hand, from the student's point of view, the keyboard may be a hindrance in the learning process. To minimize keyboard interference, most CAI systems require students to enter a character or number which is keyed to an answer. This format results in some form of multiple choice, and students know that one of the answers has to be correct. At some point in the curriculum, students should have to act more productively and come up with the answers on their own. This requires the student to enter the answer through a keyboard or some other device. A recent hand-entry device consisting of a graphics pad interfaced to a computer is capable of recognizing approximately three thousand handwritten Chinese characters and displays each character from a stored representation. The company claims that alphabetic sets of two to three dozen symbols would be easy for the system to recognize. This capability is tantamount to writing with pencil and paper and would eliminate any keyboard hindrance for the student as well as allowing for real production.

At this stage of technology, a random access audio device such as the Instavox is needed to provide high quality audio material as well as fast and accurate accessing. As pointed out by Wyatt, it is difficult to access accurately analogue tape devices, and the access time is usually quite long. This single feature virtually eliminates the interactive capabilities of analogue tape for CAI. Digitizing speech at this stage of technological development is extremely costly. In order to have a high enough sample rate to deliver high quality speech, a large amount of memory is required. Reproduced speech for language learners ought to have a bandwidth of 50-10K Hz. Unlike the native speaker of English who can do without the high frequencies to distinguish fricatives over a telephone, the student of a foreign language needs a more substantial portion of the audible spectrum to distinguish foreign phonological contrasts. Several microcomputers have the ability to synthesize speech using digitization. However, what is known in the area of speech synthesis is presently too primitive to be of use for foreign language teaching. Its only present value, on micros at least, is for entertainment. Phonological contrasts are sometimes lacking, and suprasegmental information is all but absent, especially at sentential levels.

There may be a technical limitation to random access devices such as the Instavox. While one can reach any indexed portion of the floppy disc, when a longer duration of speech is stored, it is difficult to access randomly shorter stretches of a verbal presentation. For example, if an author indexed a thirty-second passage on the floppy, the student can always reach the indexed beginning of that thirty seconds. If a student had trouble with the last part of the sentence ten seconds into the passage, it would be difficult or impossible to reach that particular portion because it was not originally indexed. No such indexing restrictions occur with analogue tape. Here the random access audio device offers extremely fast access time to make it a useful interactive peripheral for CAI, but it requires substantial indexing, and for some applications this may be very challenging, e.g. indexing every phrase or word so that the student can access any portion of a long speech.

The desire to manipulate large stretches of language, i.e. text and discourse, is a result of the requirement to bring students up to level three. Many micros at this time appear to be somewhat limited in physical capability to meet this requirement. However, the presence of newer systems appears to address the superficial problems of display, exotic orthographies, and storage. One area that must be pushed further is the soft end: in particular, both software and the design of programs to deal with more semantics. In all existing systems, the nature of computation is to compare student input answers with stored ones. This type of isomorphic string matching capability is of a relatively elementary level. This is fine for matching a student multiple-choice input of 5 against the stored answer 5 or the student's input 自然 against the stored answer 自然 The Wyatt paper discussed the difficulty of a machine's accepting open-ended answers. However, one area that appears to have promise in expanding the capability of accepting more opened-ended answers is the field of artificial intelligence in conjunction with natural language processing. Complex programs will be required to handle open-ended answers. Radically different types of systems are now becoming available. They utilize new and extremely powerful software tools and techniques based on the notion of exploratory programming, the conscious intertwining of system design and implementation. With such tools, it is possible to experiment with collaborative activities with more open-ended answering capability. The strength of these tools is that one can quickly prototype materials in a variety of programming algorithms/paradigms and not be constrained by

less powerful languages such as BASIC or PASCAL. The availability of such languages as INTERLSIP-D, SMALLTALK, or LOOPS will allow us to construct small scale knowledge-based, or intelligent computer-assisted instructional systems that will store and manipulate various student choices and be able to globally handle a higher level of strings. This will move us a step further in dealing with open-ended answers and a higher-level interaction with students.

The several new approaches to foreign language CAI, collaborative and facilitative activities, are promising. The only caution that one might observe is attempting to implement such activities on systems that may be too limited in their capabilities. Otherwise, potentially effective courseware will be restricted by equipment limitations, and the actual goals of the activity may be sacrificed. It would be prohibitive for us to attempt courseware development on a supercomputer, but there are affordable systems on which we can begin developing new and more intelligent types of activities for foreign language CAI.

Languag

Wyatt has give
ble applications of
testing of the rec
positive, forward-
could have in th
listening skills, hi
presents a numbe
deserve expansion

The Limitations
The limited abili
has implications
development of r
popular microcon
most CALL appli
microcomputer) is
have a larger 80 x
the text much less
page is a better me
amounts of text.

The ultimate ai
to have a learner c
substance, and und
segmentally, frame
will be smaller than
is needed for q
features—the stud
following the logic
continuity will ad

Wyatt proposes
page displays text,
questions and fee
compromise has be
sity of Southern C
the outcome have
must, however, wor
inconvenienced by
between two medi
ponents of the same
vantages of this ap
balanced by the ad

ivities, are
ght observe
tivities on
apabilities.
are will be
the actual
t would be
e develop-
affordable
ng new and
or foreign

ing
um-
nch
s 32
the
art,
ated
plus

The Computer and Limitations

Glyn Holmes
Language Laboratories, University of Western Ontario

Wyatt has given us an excellent survey of possible applications of the computer to the teaching and testing of the receptive skills. While essentially a positive, forward-looking picture of the role CALL could have in the development of reading and listening skills, his paper is not exclusively so: he presents a number of caveats, and some of these deserve expansion.

The Limitations of the Computer

The limited ability of the computer to display text has implications for the role of CALL in the development of reading skills. The size of many popular microcomputer screens (and we assume most CALL applications will involve the use of microcomputer) is a 40 x 20 matrix. Some micros have a larger 80 x 40 matrix though this can make the text much less readable. Basically the printed page is a better medium for displaying substantial amounts of text.

The ultimate aim in reading comprehension is to have a learner confront a text, usually of some substance, and understand it. If a text is presented segmentally, frame by frame—and often the frame will be smaller than the 40 x 20 matrix, since space is needed for questions, feedback, or help features—the student will have greater difficulty following the logic of the text. This decreased visual continuity will adversely affect comprehension.

Wyatt proposes a merger of media: the printed page displays text, whilst the computer provides questions and feedback, help features, etc. This compromise has been in operation at the University of Southern California, though no reports on the outcome have been forthcoming as yet. We must, however, wonder to what extent students are inconvenienced by having to divide their attention between two media to perform different components of the same activity, and whether the disadvantages of this approach are sufficiently counterbalanced by the advantages.

If we accept the premise that it is preferable to have a student concentrate on only one visual area (text or screen), we must argue that the best applications of CALL are those in which only small amounts of text need to be displayed at any one time. It might therefore be that this limits CALL to the development of reading sub-skills such as use of contextual and morphological clues, anaphoric reference, logical connectives, vocabulary building, and the like. On another level, the limited text-displaying capacity of the computer would permit continuous reading activities where learners need only a small amount of information at one time, and where they move forward through a text, as is the case with the Adventure-type games.

As far as the listening skills are concerned, the computer is at least as limited as with the reading skill. Some systems have the ability to deliver the raw materials of listening comprehension—the spoken word—via speech synthesizers, though they are too crude to be used for language study. The computer must therefore link itself to another medium: the audiotape or the audiodisc. The computer therefore assumes a secondary role: it controls the other medium, or it can act independently as a testing device, or it can manage the student's progress. A question arises about many such potential CALL applications: Is the computer's role really necessary? This question will be dealt with in the course of the following section.

When Is the Computer Inappropriate?

In his conclusion, Wyatt raises the vital question of the "relative importance of different types of computer involvement. What should our priorities be?" Not all computer applications are of equal value. The development of a CALL application is usually a costly and time-consuming commitment. It could also create problems insofar as the CALL activity might be intended to displace an activity

involving traditional technologies (including the teacher), and any change is potentially disruptive. Before such moves are made it is wise to ensure that there are real gains to be made with CALL. It is arguable that in some applications few real gains would be forthcoming.

One particular example might be the use of the computer in listening comprehension exercises, traditionally done in the language laboratory. A "standard" listening exercise in the laboratory involves a number of components. A passage is played. The student can, in many labs, use a recap button to replay any segment, and this manual recap is generally quite accurate. The student can also return to the beginning of the passage, again with accuracy, especially if an electronic counter is involved. The exercise will include some comprehension questions, which will ideally be presented in spoken form, and these could require a yes/no reply, be of the either/or variety, or demand vocabulary/content recall.

What advantages would such an activity have if a computer-controlled cassette or audiodisc were used? Stopping the passage at any point and asking questions is also possible with a regular audiotape. Perhaps the one thing that can be done by computer alone is to reroute the student, automatically, to a given segment of the recorded material, if the student fails to demonstrate comprehension. Wyatt points out that computer-controlled cassettes can be inaccurate. If the audiodisc were used, the cost of both computer and audiodisc (approximately $3,825 with educational discount), must be weighed against the educational benefits of such a feature.

The question is simple: Do the benefits justify the time, energy, and money? Indeed there are other, similar questions that might be asked. Are we using the computer simply for the sake of it? Are we trying to do more with the computer than we need to? Are we building impressive systems where the actual, provable gains over apparently outmoded technologies are minimal, if not non-existant? It must be remembered that use of the computer is not a *de facto* justification.

Such questions perhaps need to be raised before attempting to implement other possible applications. For example, do we really have much to gain by using the computer to administer interactive post-tests? In which context would we want such an application? Do not most teachers prefer to correct and evaluate these tests of ultimate achievement and apply their uniquely human powers of discretion and discrimination? Furthermore, programming the computer to evaluate student input

is a difficult and time-consuming occupation. To my mind no satisfactory computer assessment of ability can replace a teacher. We might also ask to what extent we require branched testing techniques, which of course would be eminently well handled by the computer. Do not most teachers want their students to write a common test? Is there much to gain by using the computer to monitor reading speed, if all that can be done is have the computer act as a timer? Would it not be easier and cheaper to give each student a stop-watch?

Conclusion

This reaction paper has emphasized some of the caveats concerning the limitations of the computer and the fact that there are many reading and listening activities that are perhaps best performed by media other than the computer. With this in mind, we can underline Wyatt's eminently sensible statement that CALL activities are "only one component of a lesson." Indeed, in many situations, reason will dictate that the computer not be used at all.

General R
Wyatt's
comments
laborative
tioners, he
projection
technology
novative d
quired to e
tional me
This article
expatiating
paper: the
proficiency
Wyatt's
reading ma
on the side
be done, ra
writer's san
puter softv
much less a
grams whi
reading m
commercial
may be ava
(*Le Prison*
level, is an
hours to pl
than severa
fered for th
ship on rea
proaches li
which desc
and Germa
teachers.
In the ar
by technol
soon have

ition. To
sment of
so ask to
chniques,
I handled
vant their
: much to
r reading
computer
I cheaper

me of the
computer
and listen-
ormed by
; in mind,
ible state-
e compo-
ituations,
at be used

The Interactive Computer Testing
of Reading Proficiency

James P. Pusack
The University of Iowa

General Response

Wyatt's paper stimulates a cascade of reflections, comments, project ideas, counter-points, and collaborative urges. Like other seasoned CAI practitioners, he has chosen to keep his speculations and projections well within the bounds of current technology. He lays out both traditional and innovative directions in CAI in the kind of detail required to evoke a realistic impression of the instructional methods and accomplishments involved. This article will touch on two specific points before expatiating freely on one main topic of Wyatt's paper: the interactive computer testing of reading proficiency.

Wyatt's survey of possibilities in courseware in reading may be unintentionally misleading; it errs on the side of optimism by dwelling on what can be done, rather than on what has been done. This writer's sampling of currently available microcomputer software for reading reveals a state of the art much less advanced than Wyatt's portrayal. Programs which go beyond sentence-sized bites of reading material are nearly non-existent on the commercial foreign language market, although they may be available on the ESL market. One program (*Le Prisonnier*), cited as "low intermediate" in level, is an adventure game which may take several hours to play; during the course of play, no more than several dozen words and phrases may be offered for the student's reading. Likewise, scholarship on reading via CAI is really very meager; approaches like those in the cited Weible reference, which describes a technique using mixed English and German text, may not satisfy most language teachers.

In the area of listening, which is heavily bound by technology, it is doubtful that video*disc* will soon have widespread impact in our schools, col-

leges, and universities. The cost of the equipment is less a concern here than the availability of materials. However, computer-controlled video*tape*, at possibly half the cost of videodisc equipment, can provide complete frame accuracy on materials which are as current as today's television broadcasts. For short interactive listening tasks typical of language instruction, access delays may well be negligible, particularly if they can be made to coincide with periods of comprehension checking undertaken by a computer program. The jury is still out on the relative advantages of these two technologies which, in tandem, promise radically to restructure the teaching of aural and cultural comprehension.

Focus: Testing Reading

The possibilities for developing sophisticated methods in the testing of reading proficiency deserve closer scrutiny. This subject combines serious challenges in the design of computer materials with the greatest potential for success.

What standard must be demanded of computer programs which accurately test reading? To my way of thinking, they must attain—in every way—the adaptive power inherent in the *Oral* Proficiency Interview as developed by the Interagency Language Roundtable. The oral interview is an unequalled model for adaptive testing. The test candidate's level is checked, specific skills are evaluated, the candidate has some degree of control over the precise questions and answers, and a relatively reliable result is produced. The prospect of using a computer to achieve this kind of interaction on behalf of reading is nothing less than exhilarating. Indeed, the underlying computer programs which could accomplish such a task would produce not only a single proficiency score, but also a thorough

diagnosis of the candidate's strengths and weaknesses. Such a diagnosis might well be more valuable as a teaching tool than the test results themselves.

One of the reasons a serious project to deal with reading proficiency testing seems so promising, compared to other language skills, is that computers are very good at the display of text—much better than they are at understanding language. As Wyatt points out, answer-processing for even simple sentence answers is extremely complex and often unreliable as a testing device. Reading skills, on the other hand, involve sequences of decisions and discriminations which lend themselves to algorithmic analysis. Since a reading test would, by definition, exclude active language production, the harrowing problems of language analysis can be avoided. Instead, the tremendous memory and analytical/combinatorial powers of the machine can be harnessed to the task of building a model of the individual student's reading ability. The complexity of the program would lie not in the analysis of language, but in the mapping of relationships and in the decision-making processes aimed at exploring a student's specific strengths and weaknesses. Unlike paper-and-pencil tests, interactive reading tests would attempt to capture and analyze the step-by-step *process* of reading, rather than its *product*.

This vision of a highly adaptive form of reading examination is strongly colored by artificial intelligence approaches to analysis of student errors. One striking example is Brown and Burton's (1) description of a diagnostic modeling system for automatically synthesizing a deep-structure model of a student's misconceptions or "bugs." The computer model was used successfully to analyze errors and to train both students and teachers to discover so-called "buggy" procedures in subtraction. The project used a representation technique called "procedural networks" which captured misconceptions by generating both correct and incorrect procedures. It is worth noting that the program was tested on a data base of 19,500 problems by 1,300 students.

While the testing of mathematical skills cannot be transferred directly to methods of testing reading, the level of complexity in the BUGGY project and many of the design concepts seem consistent with the view that reading proficiency testing must and can achieve the diagnostic power of the oral interview. Certainly we can move far beyond the discrete-item printed format which is now almost universal practice.

Interaction Types

The computer, then, seems to offer a mechanism to traverse the branches and nodes of a student's reading proficiency. This section sketches some speculative but not unrealistic ideas for the shape of an interactive diagnostic reading test.

At the heart of the question lies the issue of whether we really understand reading proficiency well enough to test it in the precise fashion suggested. The ACTFL Proficiency Guidelines specify an interlocking set of proficiency levels and precise cells of skill and knowledge. Before looking at the Guidelines more closely, let us explore some of the ways a computerized interaction could isolate steps—some of them false—in the process of reading a foreign language text.

First, the computer as a display medium can exercise far more control over the text than can the printed page. It goes without saying that different texts can be presented as the "interview" homes in on the candidate's putative skill level. Texts and comprehension-checking mechanisms can be adjusted to performance. Some texts can be generated by the program—not simply reprinted—to verify discovered strengths or weaknesses.

Second, access to the text can be restricted or manipulated. For example, the student can be given a full opportunity to read a text, but the text can then be removed during comprehension-checking. This eliminates the possibility so common in printed reading tests that the questions are used by students as a kind of pony for the text. The questions, too, can be presented in isolation, so that information contained in a subsequent question will not invalidate a previous item. The whole questioning strategy changes once this restriction is lifted.

Third, timing can be exercised over the reading experience. If skimming and scanning are the skills to be tested, some measure of the time required may be a useful indicator of proficiency.

Fourth, access to reading aids can be provided and taken into account in the diagnosis. A major weakness of traditional reading tests lies in the fact that they seldom handle lexical knowledge well. One of the best students I ever had in an intensive intermediate German course, a graduate student in music theory, described how she had finally completed her departmental reading exam in German for the Ph.D., administered by one of her professors. She was given a sight translation in her field and allowed to use a dictionary—a reasonable testing method for a research skill. When the test had been graded, she went to her professor to find out if she had passed. He grudgingly admitted that her translation had been, in fact, excellent, but said

nechanism
student's
ches some
the shape
it.
e issue of
roficiency
shion sug-
nes specify
ind precise
king at the
ome of the
ild isolate
process of

im can ex-
an can the
t different
w" homes
Texts and
can be ad-
e generated
—to verify

stricted or
an be given
ie text can
-checking.
1 in printed
oy students
tions, too,
iformation
ill not in-
juestioning
is lifted.
he reading
re the skills
quired may

e provided
s. A major
i in the fact
ledge well.
n intensive
e student in
inally com-
in German
of her pro-
in her field
reasonable
ien the test
ssor to find
mitted that
nt, but said

he hesitated to pass her because he had observed her using the dictionary rather too frequently!

That which, at first glance, seems unfair in this anecdote reminds us of an important consideration: we should test not only results, but also efficiency in reading. A computer-administered test can make lexical aids and other explanations available, but take them into account in the scoring. Since few of our students will ever learn to read without the help of a dictionary, a testing mode which measures the use of this tool promises a much fairer test than we can now construct. At the upper end of the proficiency ladder, where "educated native" proficiency is required, target-language dictionary entries will also be an option.

How would a test "item" look in an interactive reading test? It would certainly be quite different from a static chunk of text followed by the ever-unpopular multiple-choice questions. Some items, to be sure, would use the cloze method suggested by Wyatt. But while cloze tests may be excellent for measuring the educational level of a native speaker, as diagnostic mechanisms, they remind me of the "sudden-death" vocabulary quizzes I try to entice my students into taking on cold winter Fridays: instead of the typical litany of words for translation, I offer to give them just a single, fateful, word. If they know it, they get an A; otherwise they get an F. In like fashion, cloze testing is fast, efficient, but may not lend itself to any large-scale, sensitive diagnosis of strengths and weaknesses, and, in any case, it reintroduces the answer-processing issue.

The ideal "reading interview" would be one in which the text itself, as it were, interviews the student. A text would be displayed in conjunction with a comprehension task. The task could be any one from our standard inventory: scan for facts, skim for global meaning, anticipate the way the text would continue, pick an appropriate title, reconstruct a series of events, draw a conclusion. In particular, the technique of hypothesis formation described by Wyatt appeals to a higher order of comprehension. It is a kind of cloze at the textual, rather than lexical level. It is less cruel.

As long as each task in a series is completed successfully, the reader moves on to different areas of knowledge and higher levels of skill. At the point where lack of comprehension appears, however, the interaction moves to an exploration of the sources of misunderstanding. Is it vocabulary? Is it a particular syntactic structure? Is it a more general inability to synthesize meanings from isolated facts? The goal now is to plumb the depths of the student's mal-comprehension and to chart the subsur-face landscape of his or her reading skills.

To turn now to the precise ACTFL Provisional German Descriptions for Reading Proficiency, as a concrete case, let us put three questions to them: 1) Is there anything there which can be tested by the computer? 2) Is there anything there which *must* be tested this way? and 3) Is there anything which would elude interactive computer testing?

Of course, there will always be room to argue about specific guidelines. Overall, however, a surprising number of points of knowledge and skill promise relatively easy conversion to an interactive environment. These areas include the recognition of fixed vocabulary and survival expressions. Programming to check for this rather static knowledge component will certainly not strain our capacities, nor will the approach differ radically from that of the past. However, a number of *conditional* skill descriptions could really only be tested efficiently under interactive conditions—be they directed by machines or by human testing agents: "Where vocabulary has been mastered, can read standardized messages." "Misunderstandings may arise when grammatical cues are overlooked." "Reads within the limits of identifiable vocabulary." These and many more similar requirements set up testing conditions where certain assumptions must be validated before the test portion begins.

Another area where the ability of the computer interactively to control the test content becomes critical lies in descriptions which relate to the student's individual situation and preferences: "if the content is familiar or of high interest," "in response to perceived needs," "authentic prose on familiar subjects," "concrete topics related to special interests," "understanding of specialized items depends upon individual interests and background," etc., etc.

Two specific kinds of requirements make computer control over text display not only critical, but essential: "May have to read several times before understanding," and "Able to read at a normal rate of speed (at least 220 WPM)." Given the rigor of these criteria, it is difficult to see how they could be tested reliably without the computer.

Finally, there are certainly some identifiable kinds of requirements which do elude interactive computer testing: "Can read books." "Appreciates descriptive material." "Can read for pleasure." and "Can follow simple printed directions for cooking." It is intriguing to imagine ways to test these elements interactively. Following certain kinds of directions can, in any case, be handled by the computer.

This brief scrutiny suggests that an interactive computer delivery system is a compelling solution to the assessment of reading skills as they are defined in the Guidelines.

Development and Delivery of Interactive Reading Tests

So much for speculation. What would it take to develop and deliver a testing system of this nature? Practical answers must provide for both software development and on-site delivery mechanisms.

There can be no doubt that a good deal of expertise in testing, cognitive science, computer programming, and language teaching are needed to realize the potential described. The interactive model would be generalized enough to apply to many foreign languages, but the production of test materials would also depend on extensive work unique to each set of skill descriptions in the Guidelines. This is a major project which will not be completed by hammering away on our Apples late into the night.

A recently published report of a research conference on computers in education outlines a national strategy for advancing in areas of vital concern. In particular, it describes the effort required to confront such cognitive issues as expert and novice thinking, comprehension and writing strategies, knowledge structure, mental models, and cognitive psychometrics. Diagnosis is one of the major areas identified for computer applications. Language-teaching professionals can not only learn from such endeavors, which are aimed in large measure at native language skills; we can also move to collaborate with colleagues from other fields in the design of the instructional computer systems for conducting activities common to many kinds of teaching (Lesgold and Reif, 2).

The question of hardware is skirted here because it is fundamentally dependent on market forces, rather than the inherent capacity of computers today. One of these is the issue of screen display of text, which Wyatt has tackled using the "booklet approach." This approach may be meaningful for non-testing environments which use relatively long texts. The testing of reading proficiency would use shorter texts, amounting to no more than several paragraphs. For this reason, expected advances in technology, coupled with predictable price reductions, will make available much more powerful machines, with bigger screens and enough memory to store many pages of text for immediate display at the touch of a finger. In designing a sophisticated testing system, we must look ahead to the next round of equipment.

Reading tests will also require large disk storage capacity to hold both lexical and textual materials, as well as test items. If students are allowed a choice of subject matters and a range of proficiency levels, the move will be to hard discs.

As a result, adaptive, interactive reading tests will definitely *not* be distributed on diskettes to thousands of high schools across the land for use on their Apples. A smaller number of testing centers will have the required equipment and administer the test, with proper security. While this equipment will be more expensive than paper-and-pencil tests, it will certainly cost much less, over the long term, than the human delivery system we need for oral proficiency interviews. Computers do not need to attend $1,000 workshops, generation after generation. It is clear that the time involved in administering oral interviews will always weigh more heavily in the balance than the cost of any small computer.

Meanwhile, all those Apples are floating around our schools and colleges. And we wonder whether "keyboard shock" will sully the results of our marvelously adaptive tests. Pared-down versions of our testing procedures—with fewer texts and less ability to adapt to levels—can serve well as interactive pretests administered on low-priced microcomputers. Students can become familiar with this kind of reading experience, and we can exploit the investment our institutions are already making in instructional computing.

In sum, the development of such a system will doubtless require substantial grant funding, but the delivery system will not cost much more than the microcomputers now in use, given foreseeable growth in computer power for the dollar. The major continuing expense will lie in the maintenance of the item pool, which will be extraordinarily complex by today's standards.

Conclusions

A project to develop interactive computer testing procedures for reading represents a natural outgrowth of our profession's concern with standards of proficiency. If we are serious about the standards which have emerged so far, we must also look toward ways to implement them. Any mechanism which offers less adaptive power than an interactive computer system will do a grave injustice to the concept of proficiency we have so recently attained. Experience in a full-scale reading project, moreover, will contribute mightily to methods of testing listening, culture, and writing. Many of the strategies are transferable, but they are best developed for the skill we are now most

able to hand
reading.

1. Brown,
   "Diagnos
   Mathema
   155-92.

able to handle without additional technologies— reading.

REFERENCES

1. Brown, John Seely and Richard R. Burton. "Diagnostic Models for Procedural Bugs in Basic Mathematical Skills." *Cognitive Science* 2 (1978): 155-92.

2. Lesgold, Alan M. and Frederick Reif. *Computers in Education: Realizing the Potential,* 2 vols. [Report of a Research Conference, Pittsburgh, PA: Nov. 20-24, 1982. Washington, DC: U.S. Government Printing Office, 1983. (1983-381-054: 13415)].

ACTFL Comprehensive Membership

A bargain at $55.00!

When you renew your ACTFL membership for 1985, consider taking a Comprehensive Membership.

A Comprehensive Member receives all the direct benefits of a Regular Member:

* six issues of *Foreign Language Annals* (February, April, May, September, October, December)

* a 25% discount on ACTFL Materials Center publications

* a discount on fees for ACTFL institutes, seminars, and workshops

* an opportunity to participate in three group insurance plans

* a ballot for the annual election of ACTFL officers

* an invitation to the Annual Meeting.

In addition, a Comprehensive Member receives:

* six issues of the *Public Awareness Network Newsletter* (January, March, May, July, September, November)

* the annual volume of the Foreign Language Education Series

* additional publications (1983 titles were: *Academic Preparation for College: What Students Need To Know and Be Able To Do; Functional-Notional Concepts: Adapting the FL Textbook;* and *Internationalizing your School: A Handbook & Resource Guide for Teachers, Administrators, Parents and School Board Members*).

A Joint Comprehensive Membership (for two members residing at the same address) is $60.00.

Suggestions for Comprehension-Based Computer-Assisted Instruction in German

Karen C. Kossuth
Pomona College

In the ten years since foreign language departments began to acknowledge that computers could be used for foreign language instruction, a great deal of effort has been spent getting equipment, generating fundamental programs, keying in the exercises from standard texts and evaluating the effectiveness of computer-assisted foreign language instruction. At the Claremont Colleges, we have progressed since the mid-1970's from those first tentative steps of optional, ancillary drills for the weak or the ambitious, to mandatory sessions with a sophisticated accounting procedure developed in 1982 for John Poynter's classes at Claremont McKenna College and in use also at Pomona since fall 1983. All our evaluations of CAI show marked superiority in the grammar test scores of students who use the computer over those who don't, bearing out the comment of Benseler and Schulz that CAI "appears to be especially useful for self-paced, programmed instruction of the grammatical system of a language and the reading skill."[1]

But very many CAI programs are in the pedagogical Dark Ages. Their level remains at the old-fashioned grammar-translation stage in historical foreign language pedagogy, with occasional steps up to the transformational drills of the audio-lingual Middle Ages. The drawback of these exercises is that they are drills; there is a right answer, and though the students may not know exactly what it is, they are rarely surprised to find out. They are never delighted. In short, CAI may be good for our students but in most cases it is a crashing bore.

Boring and predictable CAI is the antithesis of the Renaissance going on in our classrooms. There we bring the real world to class, involving our students in discussions, however rudimentary, of issues of interest and importance to them personally. We base our approach on a notional syllabus rather than on the introduction of grammatical categories. Research is demonstrating that foreign languages are learned most effectively not through organized presentations of morphology and syntax (Krashen calls these presentations "language appreciation"), but through successful and active comprehension of interesting material presented exclusively in the foreign language. Krashen hypothesizes that effective language acquisition depends on input which is: 1) comprehensible, 2) interesting and/or relevant to the acquirer, 3) not grammatically sequenced, and 4) provided in sufficient quantity.[2]

After a decade of growing pains, CAI is now also ready to offer our students comprehension-based instruction. We have the technology; many colleges and universities and some high schools have the machines. Now, before we discourage generations of teachers and students because CAI has been so predictable, we need to interject elements of surprise and variety into the sessions with the computer. The purpose of this article is to report on a project in communicative CAI which we have initiated at the Claremont Colleges.

It all started with ELIZA, an exercise in artificial intelligence programmed in the middle 1960's by Joseph Weizenbaum.[3] Weizenbaum used a Rogerian psychotherapeutic setting for a program which seems to engage users in conversation about their psychological problems. The program is based on a system of keywords, which if they appear in a user's responses, evoke a programmed response from the computer. The genius of ELIZA lies in its ability to combine parts of the user's utterances into new ones it gives back. This parsing ability is based on Weizenbaum's understanding of English syntax, and presents the principal difficulty for us in translating ELIZA into a foreign syntax. Nevertheless, it is clear that ELIZA could serve as an example of Krashen's "friendly native-speaker" and conduct conversations with the students. In June 1982, we rechristened ELIZA as LIESL and began programming. We thought our application of ELIZA to CAI was unique until we were introduced to John Underwood of Mills College, who is interested in conversational computer-assisted instruction and whose article on ELIZA in Spanish had just appeared.[4] Though Weizenbaum and Underwood worked in the programming language LISP, we have adapted ELIZA in BASIC from Ahl's *More Basic Computer Games* in the expectation that it will be more accessible to language labs using microcomputers.[5] We have ten scripts available to students wishing to talk to LIESL. One of these is a German adaptation of Weizenbaum's psychotherapist program; several more discuss stories from a forthcoming reader, *Alte Legenden und neue Literatur*,[6] and the remainder deal with topics from a notional syllabus we use in class, such as "Was hast du am Wochenende gemacht?" or "Welche Kurse belegst du?" or "Sternbilder" or "Freizeit." The possibilities are limited only by the patience and imagination of the programmer. Because we offer a new script every other week, we are able to maintain LIESL's conversational skills at an acceptably interesting level. LIESL's responses are lively—even sassy—and she frequently delights the user with a response so unexpectedly appropriate as to elicit a chuckle. We require students to submit weekly printouts of a conversation they had with LIESL, such as the ones below, both taken from a *Wochenende* conversation:

Example 1

Tag, Liesl hier. Wie war das Wochenende?
SUPER!   (keyword: super)
Toll, erzähl mal davon!
ICH WAR AUF EINER SCHÖNEN PARTY.   (Party)
Wo war die Party? Ich habe nichts davon gehört.   (Party)
DIE PARTY WAR IM STUDENTENHEIM.

Hast du nette Leute kennengelernt?
NEIN, ICH WAR MIT MEINEM FREUND DA.   (Freund)
Wie heißt dein Freund?   (no keyword)
ER HEISST PAUL.
Erzählst du mir die ganze Geschichte?   (no keyword)
JA, ICH MACHE ALLES MIT PAUL.
Erzähl mir mehr davon.
WIR SIND ZUSAMMEN ZUM STRAND GEFAHREN.   (fahren)
Ja, der Strand ist schön. Wie war das Wetter?
ES WAR GUT.   (gut)
Schön, das freut mich. Was hast du gemacht?
NICHTS! JETZT MUSS ICH GEHEN. TSCHÜSS!   (Tschüß)
Also Tschüß!

Example 2

Tag, Liesl hier. Wie war das Wochenende?
SO, SO.   (so so)
Uninteressant, wie? Was ist passiert?
ICH HABE DIE GANZE ZEIT GESCHLAFEN.   (schlaf)
Schläfst du während der Woche nicht?
DOCH, ABER NICHT GENUG.
Warum hast du MICH am Wochenende nicht besucht?   (no keyword)
ICH MUSSTE STUDIEREN.   (studier)
Du bist ein sehr fleißiger Student, was?
NEIN, DAHER MUSS ICH SO VIEL STUDIEREN.   (studier)
Man kann nicht immer studieren. Ich habe selbst
gar nichts getan.
JA, DU MUSST NICHT ARBEITEN.   (arbeit)
Was für einen Job hast du?
ICH ARBEITE IN DER BIBLIOTHEK.   (arbeit)
Verdienst du gut dabei?
GENUG.   (no keyword)
Hast du sonst etwas gemacht?
JA, ICH HABE FERNGESEHEN.   (fern)
Was hast du gesehen?
EIN FUSSBALLSPIEL.   (Fußball)
Amerikanischen oder europäischen?   (no keyword)
AMERIKANISCHEN. NATÜRLICH.
Das ist sehr interessant.
JA. TSCHÜSS, LIESL.   (Tschüß)
Also Tschüß.

Using these logs, we can monitor student progress and also improve our scripts; for example, the cue *Fußball*, a masculine noun, evokes a masculine agreement from LIESL, even though it occurred in a compound word, *Fußballspiel*, which is neuter. We may want to use two different keywords to cover the two different genders. This problem is typical of those we meet with each new script.

We control the vocabulary in LIESL's responses to match the level at which we anticipate our user to be. But in an effort to keep the program as user-friendly as possible, we have also provided a dictionary, so that the student can get a dictionary entry for any word in LIESL's language by typing that word preceded by a question mark.

We have encountered some problems. In order for LIESL to embed a student response into her own counterresponse, she has to recognize the

verb and move it to the end. We are using several heuristic devices to produce correct German word order, but so far we are only getting about eighty percent of the verbs. The literature mentions a German version of ELIZA, but it seems to be unpublished and Weizenbaum has not responded to our request for the reference or a copy of the program. If any of our readers has been working on a German ELIZA, we would be interested in exchanging ideas on how to get LIESL to use correct word order.

We have also had some difficulty dealing with irresponsible student input. We are encountering three types of problem responses:

1) We have programmed LIESL to look for English (defined as language specific combinations of letters) and to respond:

"Sag mal, kannst du denn kein Deutsch?"
or "Weißt du nicht, daß ich kein Englisch kann?"
or "Genug mit dem Englisch! Laß uns jetzt Deutsch sprechen!"

2) In a similar response control section in the program, we have listed common German obscenities and programmed LIESL to respond:

"Hör mal, das war nicht nett von dir!"
or "Schimpf nicht mit mir! Ich schimpfe auch nicht mit dir!"
or "Bleib bitte beim Thema!"

At the fourth obscenity, LIESL says "Halt's Maul" and logs the loudmouth off.

3) Particularly the intermediate students like to test the computer's intelligence by typing in outrageous responses. They often try to trap LIESL into leading or contradictory reponses. We cannot anticipate most of this as well as the obscenities, which are a particular category of outrageousness. We can only hope that students will be bored by the noncommunication which LIESL gives them when they are off the subject, and thus fail to give the keywords needed for meaningful responses, so that they will get back to the business of having a conversation with a more or less coherent, friendly, and very patient "native" speaker. By the second semester, students equipped with a "cheat sheet" of suggested questions or keywords can stay on the topical track well enough to enjoy LIESL's company.

For beginning students, we offer a simpler format, one which requires only passive participation in the form of comprehension of questions and yes/no responses. Our first version is TIERE, based on the "Animal Game" in Ahl.[7] This is a guessing game in which the student is instructed to think of an animal. LIESL then proceeds through a hierarchy of guesses, to which the student answers yes or no, until LIESL discovers the answer. If the program reaches the end of its data without guessing the animal, LIESL confesses ignorance and asks how that animal is different from the last one guessed. The computer then files this information and uses it in subsequent games. Students are amused at the types of guesses LIESL makes and delighted when they can stump the computer. Because this becomes more difficult as more information is added to the data file, the students' competence is challenged with each new round. Since all but the first two questions LIESL asks were written by the students using the

game, the German is assuredly at student level. The data is regularly edited, to insure felicitous and grammatical responses. We have noticed that the game can become considerably more complex in only a few days of student use. Pomona's German animal game was adapted by student Michael Kirk; parallel ones on professions and emotions are by student Kirsten Harrison. Here is a sample of the dialog in Kirk's animal game:

Denk an ein Tier! Der Computer wird versuchen es zu erraten.
Kennst du ein Tier? (ja/nein/Liste/speichern/zurückbringen)
JA
Schwimmt es?
NEIN
Ist es ein Vogel?
NEIN
Du denkst wohl an . . . (einen . . . /eine . . . /ein . . .)
EINE GIRAFFE
Formuliere bitte eine Frage, die eine Giraffe von einem Vogel unterscheidet!
FRISST ES GRAS?
Wenn es eine Giraffe wäre, wäre die Antwort ja oder nein?
JA

Kennst du ein Tier?
JA
Schwimmt es?
NEIN
Frißt es Gras?
JA
Ist es eine Giraffe?
NEIN
Du denkst wohl an . . . (einen . . . /eine . . . /ein . . .)
EIN PFERD
Formuliere bitte eine Frage, die ein Pferd von einer Giraffe unterscheidet!
REITET MAN DARAUF?
Wenn es ein Pferd wäre, wäre die Antwort ja oder nein?
JA

Kennst du ein Tier?

and so forth, as the computer incorporates the student responses and uses them. Harrison's "Berufe" offers the student more vocabulary and tries harder to guess an answer, but does not incorporate the student's input to add data. Here is a sample:

Tag. Lies hier. Willst du das Spiel Berufe mit mir spielen?
JA
Denke an einen Beruf, und ich werde ihn erraten.
Ist dein Beruf gelernt oder an einer Universität studiert?
GELERNT
Ist dein Beruf eine der Künste?
NEIN
Ich hab's! Du bist Verkäufer(in)!
NEIN
Bist du etwa Handwerker?
NEIN
Du mußt doch Kellner sein!
NEIN
Bist du Putzfrau? Oder Putzmann??

NEIN
Bist du Friseur oder Friseuse?
NEIN
Bist du Mechaniker(in)?
NEIN
Bist du Polizist(in)?
NEIN
Bist du Krankenpfleger oder Krankenschwester?
NEIN
Bist du Klempner(in)?
NEIN
Bist du Müllabfuhrmann/frau?
NEIN
Also, ich gebe auf. Du bist sehr klug. Was bist du?
MASSEUSE!

Willst du noch einmal spielen? . . . .

In the first semester we also make use of Hangman (der Henker) to teach syllable structure. It is an easy game and keeps up interest for several rounds, after which it has to be withdrawn for a month or two. Hangman is available commercially for most microcomputers.

The purpose of this article is to open a dialogue with other teachers of German about the possibilities of comprehension-based CAI. We are at the threshold of broad innovations in format and scope. Any number of standard word games can be used for foreign language vocabulary building on the computer, and some of these are already available in English. Given scripts tuned to the level of the students and the dictionary at ready access, we can make computer-assisted teaching programs which fit Krashen's criteria for optimal acquisition. They can be readily comprehensible; using a notional format, we can make them interesting and relevant (also the very idea of working on the computer seems relevant to students); the similarity to real conversation gives us a normal range of grammatical forms rather than a planned sequence; and the quantity of material covered will depend on the students' stamina at the terminal (assuming that we have truly used material which is interesting and relevant). Thus comprehension-based CAI can serve a communicative syllabus. We need only our imagination and our enthusiasm to apply to the computer some of the techniques we use so ably in our classrooms, making the computer some language lab a valuable adjunct to comprehension-based classroom instruction in German.

Notes

[1]David P. Benseler and Renate A. Schulz. "Methodological Trends in College Foreign Language Instruction." Modern Language Journal, 64 (1980), pp. 88-96.

[2]Stephen D. Krashen. Principles and Practice in Second Language Acquisition (Oxford: Pergamon, 1982), p. 125.

[3]Joseph Weizenbaum. "ELIZA—A Computer Program for the Study of Natural Language Communication between Man and Machine." Communications of the Association for Computer Machinery, 9 (1966), pp. 36-45.

[4]John Underwood. "Using PILOT for 'Conversational' Foreign Language Programs." Educational Computer Magazine, 1 (1981). p. 33. See also "Simulated Conversation as a CAI Strategy." Foreign Language Annals, 15 (1982) pp. 209-12.

[5]David H. Ahl. More Basic Computer Games (New York: Workman, 1979). p. 56.

[6]Karen C. Kossuth, David Antal, and Deborah Shaw. Alte Legenden und neue Literatur (New York: Holt Rinehart and Winston, 1985).

[7]David H. Ahl, Basic Computer Games (New York: Workman, 1978). p. 4.

Dialect as Means to Preserve Berliner Identity

In order to update the status report on the nearly legendary Berlin dialect, the Free University in West Berlin sent two linguists equipped with tape recorder and questionnaire into the city streets to talk and listen to the people. The pair of researchers found the dialect to be ubiquitous, spoken by persons of all ages and from all walks of life. In the upper-class Zehlendorf section of the city, dialect is considered unsuitable. But in Wedding and Kreuzberg, traditional blue-collar neighborhoods, even young Turks were heard "berlinning." The dialect is best preserved in Prenzlauer Berg, an East Berlin neighborhood. This is a part of town that has remained largely unaffected by the influx of newcomers from the Saxon countryside, and its residents cultivate the dialect as a badge of their status as native Berliners. Otherwise, the researchers found hardly any differences between the dialect spoken in the nine East Berlin districts and the twelve districts in the Western part of the city.
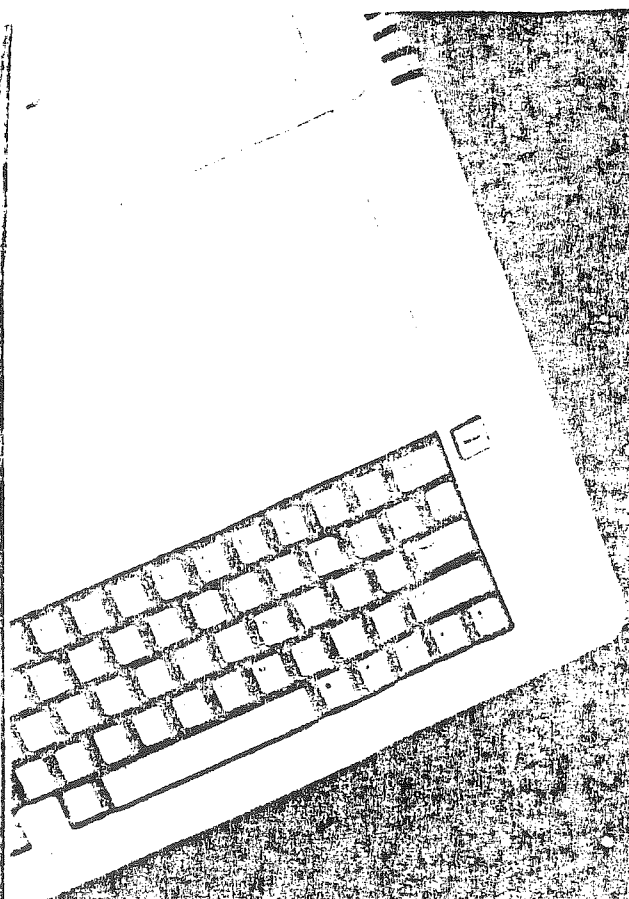
GETTING VOICE OUTPUT FROM YOUR APPLE II COMPUTER

SEASON'D WITH A C

To hear an Apple II speak its first phrases may not have quite the emotional impact that hearing a child's first halting words has, but in both cases the power of speech opens up new frontiers of communication. With the right hardware and software, an Apple can speak out loud, reciting text files, programs, and numeric values.

Speech output makes computer use possible for people who cannot read a video screen. It helps in applications where a sighted user cannot look at the monitor. Oral cues can reinforce educational programs; words, and even songs, can be added to games; and, for many people, making the computer speak becomes a fascinating project in itself.

Every Apple II-series computer comes equipped to make simple musical sounds, but its tiny speaker and output circuitry were not designed for the demands of spoken output. Adding speech to the computer means adding hardware, usually a card that plugs into an Apple II, II Plus, or IIe. For the Apple IIc, or for the other models equipped with a serial card, several

speech devices exist that link up to the Apple with a serial connection.

Good News and Bad News

To see how practical speech has become on the Apple II, I looked at a sample of available products. In general, each package included the speech-output hardware, software for creating and editing sound codes, and a program for translating text in the computer to spoken words.

The good news about voice boards for the Apple computer is that you can buy a number of boards and boxes that will produce understandable output, even from normal text. None of the products sound completely human, and for best sound quality you need to edit the input text with all sorts of special codes. For most text, though, if you have some idea of the context and the vocabulary, it is not difficult at all to

eak its first
ite the emo-
a child's first
oth cases the
new frontiers
h the right
n Apple can
xt files, pro-
s.
omputer use
annot read a
applications
ot look at the
force educa-
d even songs,
nd, for many
ter speak be-
t in itself.
s computer
simple musi-
ker and out-
gned for the
out. Adding
eans adding
at plugs into
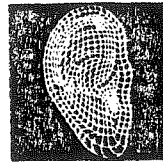For the Ap-
her models
ard, several

follow the audio output.

The bad news is that you have to be somewhat of a hacker, or very patient and determined, to make sense of these products. Of the half-dozen speech products that I looked at, not a single one had a manual that was clear, accurate, and complete. Most products, in fact, assume that you're an accomplished Applesoft-BASIC programmer, ready to PEEK and POKE your way through memory. That's fine if you're an experimenter at heart, but if you're only a computer user, you may need some help from your dealer, a friend, or a paid expert.

A good way to start looking at speech boards is to determine how they plug in. Some products, such as Street Electronic's Echo+ or Don't Ask Software's S.A.M., plug into an Apple II, II Plus, or IIe just like any other peripheral card does. Electrically, the cards will go in any slot, but if you want to use the provided software, you have to select the slot it specifies—which in most cases is slot 4.

If you have a IIc model, or you can't

Automated Mouth, indicates, it is basically a software system (and on several other brands of computers it uses only the computer's built-in hardware). Instead of a special speech chip, S.A.M. does all the speech processing in software, leaving only the conversion from digital form to the electrical waves needed for sound, as a task for the hardware. In my listening tests, the resulting quality was surprisingly good.

The SC-01 chip from Votrax has been one of the more popular speech-output chips, used not only in Votrax's

**With the right
hardware
and software, an
Apple II-series
computer
can speak out loud.**

The speech chip or data converter determines only how good the speech quality is if you send the chip codes for the right sounds. In normal use, however, you want merely to send English text to the speech system, rather than painstakingly translating what you want to say into sound codes. We'd like the computer to do any needed translations from text to code values.

Unfortunately, figuring out how to pronounce the English language is somewhat of an inexact science. Although some general rules work for most words and phrases, an endless number of exceptions seems to exist. Even the computer, as it turns out, can't keep track of all the possibilities.

All of the units I looked at came with text-to-sound-code software. Normally, this software consisted of some dozens or hundreds of general-pronunciation rules, with the hooks needed to link the software to Applesoft, routines for direct entry of text for oral output, and a demonstration or two. The exception was the two Votrax units, which had their software built in.

A GRACIOUS VOICE

nk up to the
tion.

eech has be-
looked at a
ts. In gener-
the speech-
for creating
d a program
computer to

oice boards
hat you can
d boxes that
ble output,
one of the
human, and
need to edit
ts of special
ugh, if you
ext and the
ult at all to

spare the slot, or you'd rather use a serial card and keep the cover closed on your Apple, several firms make speech-output units with serial connections. Votrax's Type 'N Talk and Personal Speech System and the Intex-Talker are all in this class. Naturally, these units tend to be more expensive because they need a case and power supply as well as a circuit board.

Another approach to making sense out of the various speech systems is to look at the hardware technology. Speech boards use a number of different technologies (see "The Dialects of Speech Output" on page 42), usually embodied in a speech-output chip. A better chip means more understandable speech, with lower memory-storage requirements.

The S.A.M. board is an exception to this pattern. As its full name, Software

own Type 'N Talk and Personal Speech System, but also in the Intex Talker. Apparently, you can expect usable, but not outstanding, sound quality from a system based on the SC-01.

Several successors to the SC-01 seemed to do a better job. The SSI-263 appears to be the current favorite, used in the Sweet Microsystem Mockingboard and the Micromint Sweettalker II. Speech produced by this chip is reasonably understandable. Unbelievably enough, this chip also adds musical tones to words, on command, letting boards that use this chip literally sing.

The TMS 5520, used in the Echo Cricket and the MultiTech SSB-Apple, also seems to do a creditable job. This chip comes from the same people who developed the Speak 'N Spell toy, so they've had a lot of experience in speech development.

If you're willing to live with some strange-sounding locutions, you can accept what you get from this automated translation process with no further efforts. Once you've linked the software to Applesoft, you can output text almost as if you were sending it to the printer. If that's not good enough for you, however, all of the boards have provisions to let you overrule what would otherwise be the result.

The easiest method to change the way the word gets pronounced is to change the spelling. If you want to make sure that the system says the name of the language correctly, spell it *Inglish* rather than *English*. Similarly, say you *red a book yester day* and found what was *allouwed*.

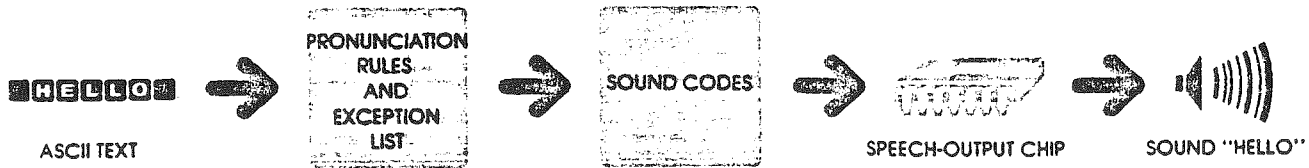If you want to go further, the Echo+, for example, lets you change the pronunciation rules. All of the

Figure 1: A speech-output system turns text into sound codes, which drive a speech-output chip to produce the appropriate sound waves.

boards let you enter text as phonemes, and some, such as the Votrax Personal Speech System, let you enter a list of exceptions to be pronounced as specified instead of by the rules.

Voices

Should you be planning to use a voice board to add speech to an existing application, note that even when the speech programs integrate with Applesoft, they still require added memory and disk space. You may have problems with protected software, which generally cannot run with other programs in memory and is often not very graceful about sharing disks.

To add a voice for the use of a visually impaired person, remember that most existing programs, particularly word processors, rewrite the same information quite frequently. It's worth the trouble to get a program specifically designed for oral use.

Learning from Experience

In researching this article, a friend and I spent a lot of time plugging in various speech boards and seeing what we could get them to do. Here's a selec-

tion from some of our notes.

S.A.M. Board

The S.A.M. board wasn't hard to install, but it ended in stripped wires rather than a speaker jack. The board itself was a tiny bit oversized, so we had a lot of trouble getting it back out of slot 4. The manual was easy to read if you knew Applesoft, a bit mysterious otherwise. We liked the liberal use of examples and had no problem making the board pronounce the words of text files and programs. Voice quality was good, with enough inflection to keep it from sounding oppressive.

Echo+

The Echo+ uses ProDOS for its software, and it assumes you know how to back up disks and otherwise run ProDOS functions. The manual seems intended more for assembly-language hackers than for casual users. If you want to use the female voice, which seems to speak with more clarity and inflection, you have to have an Extended 80-Column Card; otherwise you're limited to the more mechanical-sounding male voice. This board also plays musical tones and sound effects. If you ever need to simulate a train going through your living room, you can ask the Echo+ to do it between recitations.

SSB-Apple

SSB-Apple assumes the board is in slot 2 rather than slot 4. Because this system uses preencoded words, you have to swap the four vocabulary disks back and forth to build up your sentences. You can save the "compiled" sentences to disk. Again, this manual is more for dedicated users, having no in-

THE DIALECTS OF SPEECH OUTPUT

Three techniques for producing speech output account for almost all current designs. One, called linear predictive coding (LPC) models the way the human voice actually works. The other two store and reproduce speech based on the waveforms of the component sounds.

With linear predictive coding, a set of sound generators creates initial tones, much like the resonating of the human vocal chords and the rushing sounds of air. Next, these initial sounds pass through a set of filters that shape the sounds further, in much the way that the tongue, lips, and teeth modify sounds in human speech.

It takes many fewer bits to specify the sound generators and filter positions than it does to record the actual soundwaves, so LPC lets an Apple speak for longer periods, using fewer bits of storage. There's a slight cost in speech qual-

ity, unfortunately, because the modeling of the speech process is not perfect, and the parameters don't cover all the fine gradations. LPC requires between 400 and 2000 bits per second of speech.

When speech must be recorded for playback, or as an alternative to LPC for synthesizing messages, another popular choice is digitized speech using CVSD. This method does record sound waves rather than throat positions, but it saves memory space by encoding only voice changes. It offers higher-quality output than LPC does, but only at greater numbers of bits per second.

For the ultimate in simplicity and quality, actual sound waveforms can be converted to digital codes, stored, and replayed. This requires some 3000 to 20,000 bytes for each second of speech—far more than is practical for general personal-computer applications.
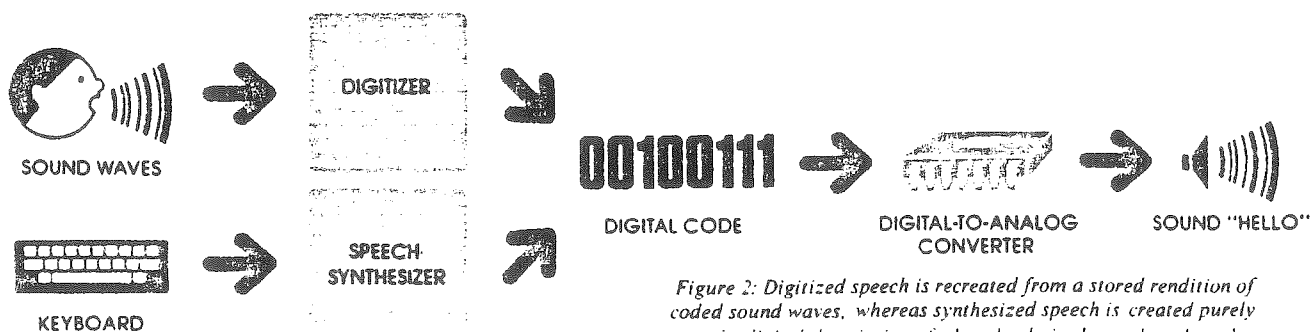

Figure 2: Digitized speech is recreated from a stored rendition of coded sound waves, whereas synthesized speech is created purely out of a digital description of what the desired sound ought to be.

dex and paying little attention to layout. The speech output is a female voice, which is particularly clear within words but sometimes choppy in the transitions.

Sweettalker II

The Sweettalker II we looked at wouldn't say a word. Apparently, it died on its way to us. Fortunately, its speech-output functions were included in the more complex Lis'ner 1000 speech-input board, from the same manufacturer (The Micromint), so we were able to try everything out anyway. The board was easy to install, but the manual was confusing—it tended to jump around, rather than proceed in logical fashion. You need one of those tiny screwdrivers to set the volume control. When the board is plugged in but not running, it makes a steady ringing sound. On the other hand, the Sweettalker demo program was by far the best of the lot. It runs through its repertoire of voices, reads aloud, and even sings a few songs. The speech quality is generally good, with an inflection that sounds almost like that of someone whose original language was Romanian.

Type 'N Talk, Personal Speech System

The Votrax Type 'N Talk and Personal Speech System (which are external boxes) require special cables, unless you run them through a modem. Unfortunately, the cable-wiring diagrams have some errors! The manuals, although attractively typeset, seem to take a particular delight in jargon. Once you get the cable squared away, connecting either system is quite easy. The Type 'N Talk gives no indication of when it is on, but the Personal Speech System has a power light. Both systems produce reasonably good speech, they take up no space on the disk or in memory, and are as easy to use as a printer. The Personal Speech System is supposed to be able to sing and play music, but we weren't able to get much in the way of understandable lyrics out of it.

The Medium is the Message

If you're just going to experiment, you're in for some fun. Although you may not find that your Apple has anything profound to say, in some respects the medium is the message. And perhaps if machines learn how to talk, maybe humans can someday learn better how to listen. ✦

Steve Rosenthal is a writer and lexicographer who tries to maintain a perspective on other joys and problems in the world besides those involving personal computers.

VOICE-BASED LEARNING

William D. Wagers

ABSTRACT

A voice-based learning system is a computer system which uses voice input and voice output in an educational environment. Such systems promise two-way communication with the computer using natural language. In addition to reading and writing skills, the computer can now teach listening and speaking skills.

Foreign language educators around the country have been experimenting with voice-based learning to determine how best to utilize these new tools in the classroom and in the language lab.

The Babel fish is a fictional creature in the novel *Hitchhiker's Guide to the Galaxy.* This fabulous creature has the ability to translate from any foreign tongue into one's native language. Simply insert the fish into your ear and you have a universal translator. The Babel fish does not understand what is being said. It is a passive device like a hearing aid.

A voice-based learning system is built around this very capability. It can understand and speak virtually any language. However, instead of inserting the fish into your ear, the fish is in the computer's ear. In this case, the fish translates from spoken language to computer language. While the computer does not understand what it hears in the human sense, the computer can associate sound patterns with written words and phrases. Embodying these associations in the form of a lesson is the function of a voice-based learning system.

A voice-based learning system includes a computer, a voice input terminal and a voice output device. It interacts with the student in oral language. While a computer does not possess the intelligence to interact fully as a human would, it can communicate.

Voice Input

Before describing a voice-based learning system in detail, a few words need to be said about voice input. A voice entry-terminal is simply a computer peripheral which accepts spoken input, compares that utterance against several known utterances, and when it finds a match, supplies the computer with the character string corresponding to that utterance. In other words, when you say, freund, the voice entry terminal selects the character string *freund.* The association between the spoken and written word is arbitrary from the computer's point of view. The associations are established by the author of the lesson. The author could just as easily associate *freund* with *friend.*

Voice entry terminals fall into two categories. *Speaker independent* devices recognize a limited number of words when spoken by any user. The vocabulary, the utterances it recognizes, of such a device is fixed. This characteristic limits its usefulness to multiple choice or truealse questions. Also, speaker independent systems are quite expensive.

Speaker dependent devices must be trained to recognize the voice of a particular speaker. You train the device by repeating each utterance into the system microphone several times. The vo-

William D. Wagers received his B.A. in 1971 and M.L.A. in 1977 from Southern Methodist University, where he studied Computer Science and Business Administration. He has worked in many phases of the computer business with assignments in the United Kingdom and Germany. He describes himself as quasi-lingual in four languages. He is currently Product Manager at Scott Instruments Corporation. His work involves researching and developing the tools required by educators for producing quality courseware for special education and foreign language training.

cabulary is created by the user (author) and is kept on disk, so each lesson can have a separate vocabulary. Speaker dependent devices are available on many microcomputers for under $1,000.

Typically, when speaking to a voice entry terminal, you must speak clearly and consistently. You must also pause between the phrases you want recognized. The pauses tell the computer when an utterance begins and ends.

35

Voice Output

High quality audio is critical in voice-based learning, so speech synthesizers were never an option. While synthesizers are cheap and reliable, they simply are not good at reproducing the richness of expression of the human voice. Videotape was eliminated because of the reliability problems and videodisc was ruled out because of the expensive mastering process. And though high-quality tape and cassette players have been around for a long time, voice-based learning requires computer control of the player.

There are currently at least two devices on the market which I have found are suitable for voice-based learning: the Instavox RA-12A and the - Tandberg TAL 822. Both devices offer high-quality audio and computer control. The Instavox is a giant floppy disk drive which plays analog recordings but allows random positioning. The Instavox can position to any message on the disk in under half a second. This means the computer can prompt or respond to the student immediately throughout a lesson. The Instavox retails for under $5,000. The media costs $25 per diskette.

The Tandberg TAL 822 is a random access cassette player. It uses a highly accurate frame counter (0.5 second) to position at rewind speed to any place on the tape. The media is the standard audio cassette. The positioning is not as accurate as that of the Instavox, and the response time is slower. However, the TAL 822 retails for under $1,500, so for a slight degradation in performance a significant savings can be achieved.

A Sound Basis for Learning

The VBLS Authoring System was announced by Scott Instruments Corporation in 1982. It was designed to allow classroom teachers to create courseware which a student would study by voice. The System consists of an Apple-compatible microcomputer, a VET voice entry terminal, and an optional voice output device.

This system gives the following advantages over traditional CALI:

1. **Visual Pronunciation Feedback.** The System has a mode in which the student's pronunciation is checked against that of a native speaker. The System displays a bar graph which shows how closely the student's utterance matches the standard. This is an extremely powerful tool for progressively refining the student pronunciation skills.

2. **Auditory Feedback.** The System plays the standard phrases as spoken by a native speaker. The student can listen and emulate the native.

3. **Helps Eliminate Unlearning.** A major difficulty in learning a foreign language is that a student frequently learns incorrect pronunciation due to inadequate visual and auditory feedback. It is far more difficult to unlearn poor pronunciation, than to learn correctly from the outset.

4. **Privacy.** Students are free to experiment, make and correct pronunciation errors without fear of embarrassment. Only the System is listening and it offers only encouragement.

5. **Success Is Reproducible.** Voice-based courseware allows successful language instruction techniques to proliferate. Each student is presented with the same learning opportunity, whereas the quality of classroom instruction varies from instructor to instructor and even from day to day.

6. **Active Learning.** Many traditional instructional technologies are passive, e.g., books, filmstrips, videotape, and cassettes. Voice-based learning permits the student to interact with otherwise passive aids in an interactive process.

7. **Ideal Computer Interface.** The human voice is the ideal means of communicating with a computer, especially in foreign language instruction. While the ideal technology is not yet available, current technology is a dramatic improvement over the QWERTY keyboard. Students are introduced to the computer in an unobtrusive manner. Virtually no typing skills are needed.

8. **Multisensory Learning.** The System stimulates multiple senses which reinforce learning.

Preparing Voice-Based Courseware

To create a lesson, the author first creates the text frames and associated responses which will appear on the CRT. The author is assumed to be a native speaker. The author then creates the audio sequences which correspond to the lesson text. Finally, the author creates the standard pronunciations, or *templates,* against which the student's utterances will be compared.

The menu-based authoring system makes the process very simple, even for the novice computer user.

The VBLS Grant Program

While the VBLS system was designed for general education, many of its early users were foreign language educators. At the 1982 FLINT Conference, Dr. Brian Scott announced the VBLS Grant Program. Its purpose was to stimulate ideas on improving the VBLS system for use in teaching foreign language. We offered a free VET voice entry terminal and VBLS software in exchange for the completion of a project defined by the grantee. Typical project proposals were for subjective studies or for units of courseware.

We awarded 15 systems to various applicants. A fair proportion of the awards went to FLINT attendees. In addition, we placed a System at the C.A.C.I. Language Center in Arlington, Virginia. C.A.C.I. supplies language training to governmental agencies and proved to be a fountain of ideas and approaches to voice-based learning. At the time of this report, some grantees have yet to fulfill the terms of their grant. However, the preliminary results were very useful in defining a foreign langauge version of VBLS.

The Grant Program is now closed, but we hope to have an ongoing relationship with those grantees who demonstrated skill and imagination in executing their projects. For example, Dr. Robert Baker completed his Russian courseware despite the absence of a Russian character set in the original VBLS software. Dr. Harry Wohlert worked with us to implement a version which used an ordinary cassette recorder. Nick Stancioff went so far as to hire a programmer to try various voice-

based learning schemes.

The version of the VBLS system which was available at the time of the Grant Program did not have foreign character sets or voice output. The current version has both foreign characters and voice output via either the TAL 822 or the Instavox. As you might imagine, the two most frequent criticisms of the original version were that you could not hear the correct pronunciation and that you could not see the correct spelling. These features are now a part of the VBLS system.

When all the reports from the grantees have been filed, perhaps their studies can be made available in some form. Their work is valuable because the voice entry terminal's performance varies with the language in use. Each language has particular problems related to the sensitivities of the terminal.

VBLS System Today

VBLS Version 1 is in use in approximately 100 installations. About fifty percent of these systems were acquired for research in areas such as auditory rehabilitation, communication training, and many others. Approximately thirty percent were acquired for use in special education by slow learners, foreign students, hyperactive children, or physically handicapped students. Perhaps twenty percent were acquired for foreign language training.

VBLS Version 1.4 with foreign characters and voice output has just become available. It is our hope that this unique system will now begin to gain acceptance with foreign language educators. We have created only a tool. The quality and quantity of courseware for the System depends upon interest and commitment by the educational community. We work closely with prospective authors to introduce them to the new technology.

We are continuing the research and development of the VBLS system. In future versions, we aim to reduce the cost of the hardware involved. We are also developing new learning modes for the System. And research is continuous on voice recognition, independent of its educational applications.

Summary

Voice-based learning is a new type of computer-assisted instruction. It holds the promise of increasing both the availability and the quality of language instruction. The fulfillment of this promise depends on the participation of foreign language educators, both in determining the proper role of voice-based learning and in embracing the technology by producing courseware.

Definitions

Authoring System - A systematic procedure for generating CAI courseware that does not require the author to have computer programming skills.

Continuous Speech - Everyday natural speech like humans use. Voice recognition products do not yet understand continuous speech without artificial pauses between words and with large vocabularies.

Courseware - A computerized course, typically including text, workbook, and software for a specific computer system.

Frame - One in a series of visual displays on the monitor of a computer system.

Human Factors - A voice recognition device must account for the variations inherent in human speech and human environments, e.g., inconsistent pronunciantion, variations in tone, pitch, duration, and emphasis, and environmental noise variations. Humans find it impossible to say the same word twice in exactly the same way, especially if they are ill or under stress. And, of course, different speakers will pronounce the same word differently. A voice recognition device must, therefore, adjust itself to these variations.

Isolated Word Recognition - Current voice recognition products recognize only words or phrases which are in its vocabulary and which are spoken as short commands or answers. Affordable products cannot pick the desired word out of a stream of continuous speech.

Keyboard Terminal - An archaic typewriter-style keyboard for manually typing data into a computer.

Prompt Square - Symbol used to indicate whether the microphone is ON or OFF; located in the lower righthand corner of the monitor when the System is listening.

Speaker Adaptive - Refers to a system which overcomes the limitations of speaker dependence through software which adjusts itself to different speakers.

Speaker Dependent - Current products are speaker dependent, meaning they must be trained to recognize the voice of a particular speaker, and will respond reliably to no other speaker.

Speaker Independent - A product which could understand any speaker using any accent.

Speech Synthesizer - Electronic device capable of producing speech-like sounds.

Template - A pattern or guide; a computerized representation of a spoken word; speech patterns.

Training - The process of inputting several pronunciations of a word or phrase for future matching to one's own or another individual's voice; enrollment; creating a vocabulary.

Vocabulary - Group of words used by or known by a person or group of people; the words trained for a specific VBLS lesson, and therefore known to the System.

Voice-Based Learning System(s) - Computer systems that use vocal dialog between the user and the computer for instructional purposes.

Voice Entry Terminal - An input device enabling a human being to speak directly to the computer; VET; VET-2; ShadowVET.

Voice Recognition - The identification of spoken words by a device. A speech recognition device can respond to the human voice, instead of, or in addition to, a typewriter keyboard terminal.

WORD DRILL - A VBLS study session mode which allows the user to practice pronunciation and receive immediate visual feedback on the correctness of pronunciation.

BIBLIOGRAPHY: Books

The VBLS Authoring System Reference Manual, Scott Instruments Corporation Instructional Design Team, Austin, Texas. Sterling Swift Publishing Co., 1983.
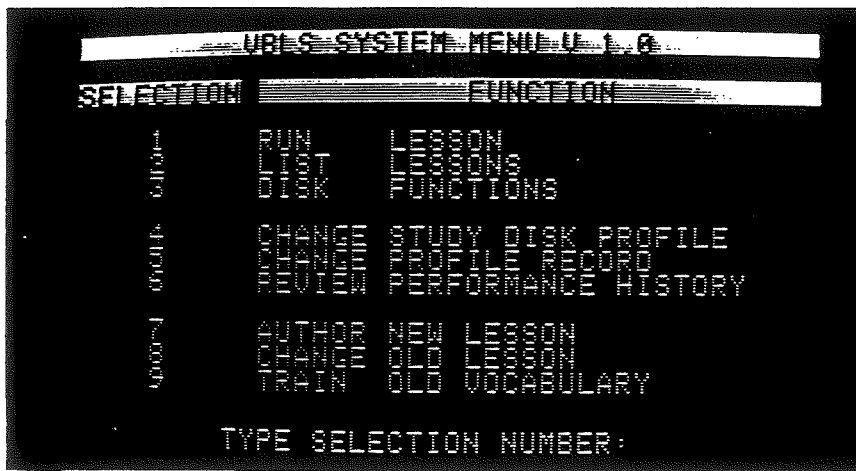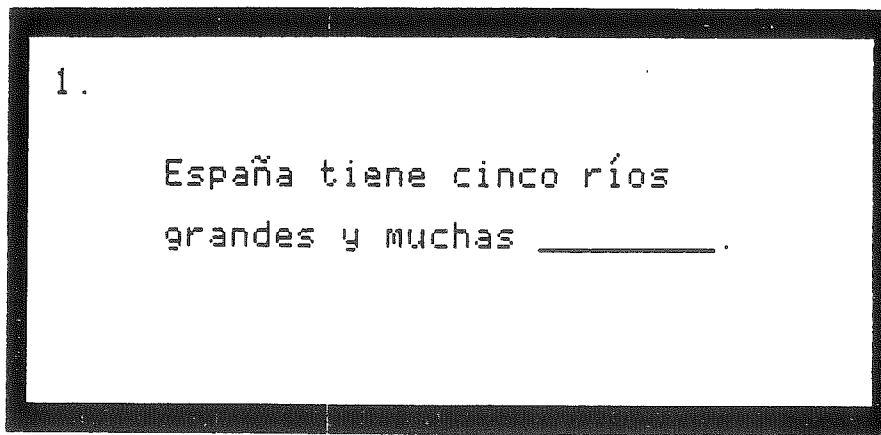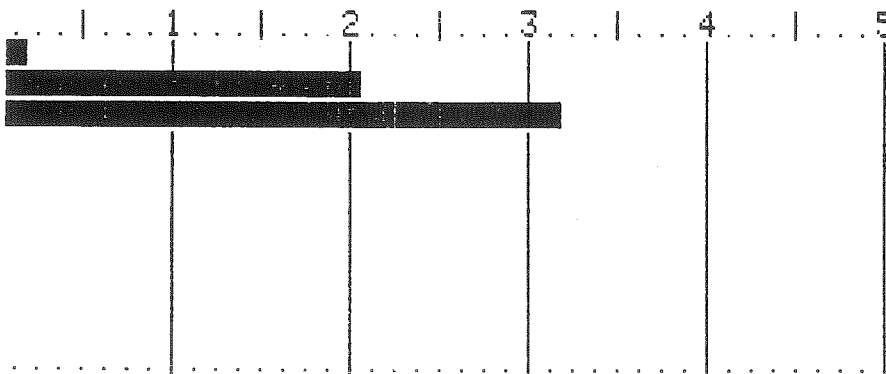
```
          VBLS SYSTEM MENU V 1.0

  SELECTION                 FUNCTION

       1      RUN    LESSON
       2      LIST   LESSONS
       3      DISK   FUNCTIONS

       4      CHANGE STUDY DISK PROFILE
       5      CHANGE PROFILE RECORD
       6      REVIEW PERFORMANCE HISTORY

       7      AUTHOR NEW LESSON
       8      CHANGE OLD LESSON
       9      TRAIN  OLD VOCABULARY

            TYPE SELECTION NUMBER:
```

Figure 1

```
  1.


    España tiene cinco ríos

    grandes y muchas _____ .




                                    ON
```

SAY: montañas

Figure 2

```
...|...1...|...2...|...3...|...4...|...5
```

LAST SCORE: 3.2   PHRASE LENGTH OKAY

BEST SCORE: 3.2

                                        ON

SAY: montañas

        (Press ESC for next word)

Figure 3

BIBLIOGRAPHY: Articles

Drake, Janice, "Authoring and Learning System Takes Spoken Response," *Speech Technology*, Jan.Feb., 1983, pp. 104-106.

Horn, Carin E., "The VBLS Voice-Based Learning System," *Interface Age*, May, 1983, pp. 28, 155-157.

Schadewald, Robert, "The Speech Gap," *Technology Illustrated*, June, 1983, pp. 55-59.

Scott, Dr. Brian L., "Voice Recognition Systems and Strategies," - *Computer Design*, January, 1983, pp. 67-70.

BIBLIOGRAPHY: Presentations

Horn, Carin E. and William D. Wagers, "Voice Controlled CAI, Presentation," *Proceedings of the National Educational Computing Conference '82*.

Horn, Carin E., "Voice Input/Output, National Educational Computing Conference, Presentation." *Proceedings of the National Educational Computing Conference '83*.

Horn, Carin E. and Brian L. Scott, "Micro-Based Speech Recognition: Instructional Innovation for Handicapped Learners," *Proceedings of the Council for Exceptional Children National Conference, 1983*.

Olsen, Solveig, "Voice-based learning systems: Some Strategies and Challenges," presentation, Modern Language Association Centennial Meeting, December, 1983.

Propst, Dr. Franklin, "Use of Random Access Audio in an Interactive Learning Environment," Presentation, E.T. and High Technology Conference sponsored by National Cryptologic School, January, 1983.

Scott, Dr. Brian L., "Today's 'Future' Technology for Language Study, Voice Recognition, and Word Verification," Presentation, Foreign Language Instructional Technology Conference sponsored by Goethe Institute, Defense Language Institute, September, 1982.

Scott, Dr. Brian L., "Speech Recognition Technology as it Pertains to the Teaching of Foreign Language," Presentation, E.T. and High Technology sponsored by National Cryptologic School, January, 1983.

Wohlert, Dr. Harry S., "Voice Input/Output in Foreign Language Learning, Presentation," Educational Computing Conference, Sponsored by the IEEE Computer Society, October, 1983.

BIBLIOGRAPHY: Unpublished Materials

Janot-Giorgetti, M.T. and M. Lamotte, "On-Line Word Recognition Using a Microprocessor System for Assistance in Learning a Foreign Language," Unpublished Paper, Centre de Recherche en Automatique de Nancy, Universite de Nancy, 1983.

AN EXPERIENCE WITH VOICE-BASED LEARNING

Robert L. Baker

ABSTRACT

Dr. Baker performed a detailed analysis of the capabilities of the VBLS system and preliminary testing of that system with students. This article presents his experience with the VBLS system, conclusions, and recommendations to VBLS courseware authors.

Before describing my experience working with the Scott Instruments' VBLS system, it would be well to set forth some of my own pedagogical principles and prejudices-- where I'm coming from.

I am known to be a fanatic for phonetics. I firmly believe that if a student is ever to speak a language without offence to the native ear, a firm phonetic foundation must be laid at the beginning. I am not an adherent of the gradual approximations theory, even though I am wholeheartedly in favor of increased emphasis and stress on communicative competence. If we are convinced of the value of foreign language study in furthering worldwide understanding, we owe it to the people whose language we are learning, whether friend or foe, to learn to speak it with some semblance of proper pronunciation and intonation. Many teachers claim that there is not enough time at the early levels to worry about such things, and most students won't ever use the language anyway (maybe we have cause and effect confused?!). But it is the rare student, usually the student who can learn a language just as well without a teacher at all, who can recoup a poor beginning and move on to not only fluent but correct speech.

I shall group my remarks under three categories: The use of the Scott VBLS system for (1) pronunciation and intonation; (2) for reading/speaking practice in the case of those languages which have a foreign script or have a poor phonetic fit between sound and symbol; (3) for use in grammar drills, etc.

PRONUNCIATION AND INTONATION

It should be remembered from the beginning that Scott Instruments does not make claims for the VET product as a phonetic analyzer. And yet I have found that *if the materials are carefully produced and if there is proper understanding on the part of teachers and students about the System's capabilities and liabilities, it can have great potential.* It is extremely important that both teachers and students realize the capabilities and shortcomings of the System and be able to capitalize on what the System does well. It seems to me that VBLS will be particularly valuable in cases where a foreign character set is involved, or where there is not a good phonetic fit between orthography and the sound system (as in French).

There are certain things which the VBLS product does very well with respect to phonetics, others which it does less well. Actually, I have the impression that even some of the things it seems to do not terribly well, it does

Dr. Robert Baker is Professor of Russian at Middlebury College, which has an international reputation for language instruction. Dr. Baker currently serves as Special Assistant to the Provost with the charge of investigating the state of the art in CAI and making recommendations to the administration. He is co-author of *Russian for Everybody* which first appeared in 1972. His co-author is a Soviet citizen. Due for publication in 1984 is *Russian for Everybody: Version for English-Speaking Students.* Dr. Baker has developed language lab materials for Russian for the past 20 years.

well on a sporadic, inconsistent basis, (due no doubt to the sampling procedure). And yet it is surprising how sensitive the System is to certain things, particularly to global analysis—it's surprising the extent to which the lesson author must be extremely careful on all matters regarding suprasegmentals, intonation, tone of voice, etc. It is essential that voicing be done in as neutral a manner as possible unless emotional overtones are specifically called for.

Otherwise a student may find him/herself pronouncing the segmental phonemes correctly, but being told he is wrong.

It is very possible that courseware prepared with the VBLS system may not be for *all* students. Students who tend to get very up-tight and who are unable to relax in front of the machine may find it a frustrating experience, particularly since repeating an utterance over and over in an attempt to improve on one's production can lead to having the frustration and irritation creep into the voice, which will affect the way the machine hears the student, and result in a vicious circle.

But those students whom I have observed working with Russian lessons prepared with the VBLS system seem to have thoroughly enjoyed the experience, to have been delighted with the capabilties of the machine, and to agree with me that VBLS has great potential in the area of making the necessary tie in the student's mind with the foreign script and the foreign sounds. The weak spots did not seem to bother them. Maybe the excitement of working on such materials on a computer is in itself such a motivational factor that this in itself has value. (We must yet determine, however, how long this effect will last with all but the most motivated students.)

It seems clear to me that use of the VBLS product will help impress upon students the need for precision in foreign language learning (something they don't seem to value much nowadays in any field, although we can expect this to improve as more of them become computer literate).

It does seem clear to me that if the system is to be used for pronunciation analysis and foreign character recognition, it is vital that the lessons be very carefully planned so as to add only one new element in each lesson, which will be mastered before moving on to another point. More care will have to be used than we are accustomed to applying, knowing (in the past) there would always be a teacher present to make up for the defects in the drills. It is possible (this will require research) that the System, if it is to be used for phonetics drills, must be used in its en-
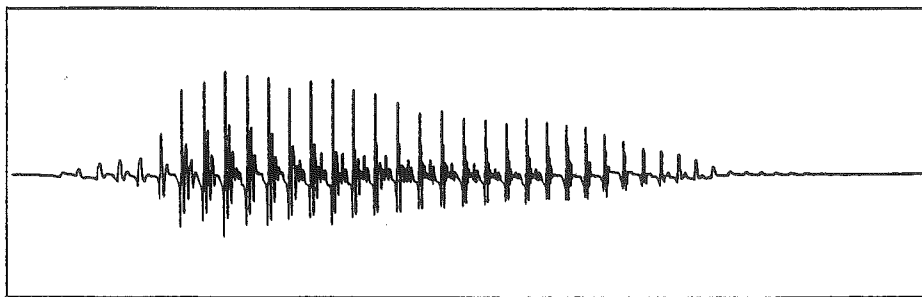
ESL student's template of the word learn
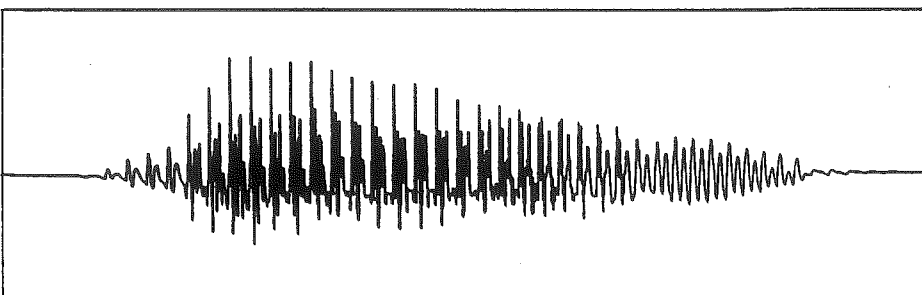
Native speaker's template of the word learn

tirety, from the beginning, working on each phonetic problem in an orderly sequence based on an analysis of the phonetic difficulties of the given languages vis-a-vis the native language; it may be that it will not be as effective if individual drills are to be extracted from the totality for remedial work on individual problems.
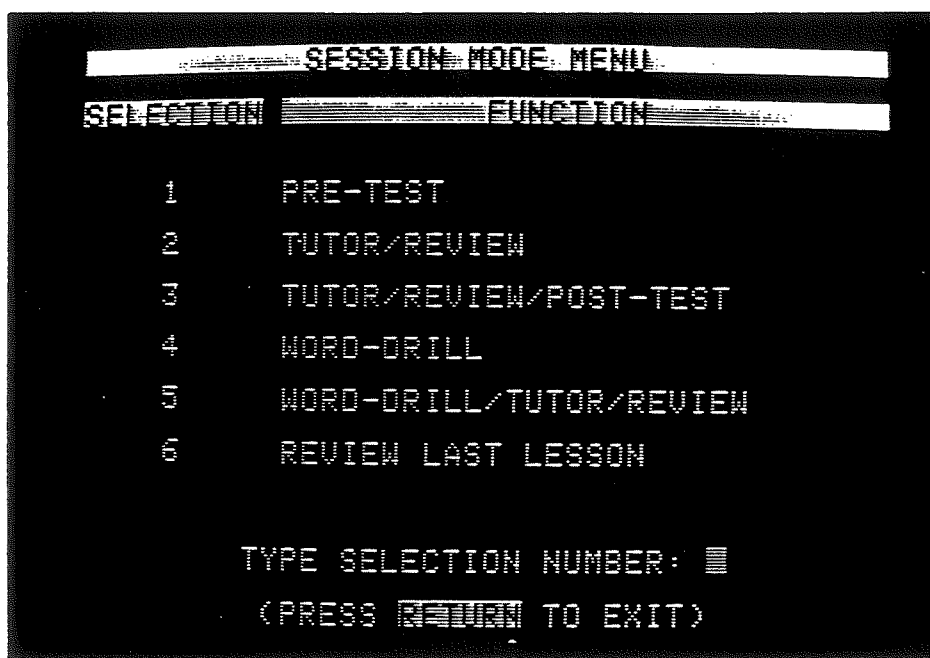
The teacher may find it advisable to use one or another of the study modes provided when dealing with phonetics matters. This writer finds the WORD DRILL mode extremely demanding (maybe too much so for most people), while the REVIEW and POST-TEST modes (after the machine has been trained to the student's voice) may not be demanding enough for strict teachers.

The following are my observations concerning the use of the VBLS system in phonetics drills:

1. The VET terminal is surprisingly good at distinguishing accented pure vowels and detecting diphthongization (at least if at all extreme).
2. Not good at distinguishing initial consonants (Scott Instruments realizes this). It is rather good at detecting extreme aspiration, but less good at discriminating voiced vs. voiceless initial consonants. On drilling these distinctions, it would be wise to place the consonants in

question in medial position in the word.
3. Although not outstanding on initial consonants, the system is still surprisingly sensitive to fully-voiced consonants vs. American English semi-voiced consonants (an essential point in good Russian pronunciation).
4. Not terribly good at discriminating released vs. non-released final stop consonants (important in Russian).
5. Really very sensitive in most cases of intonation, tone of voice, location of phrase accent, relative syllable strength, etc. (Perhaps too much so, at least unless voicings are done by more than one person.) Excellent on requiring smoothness in reading, without excess pauses (particularly since the microphone will turn off if there is too long a pause).
6. Not terribly good at differentiating between apical and blade production of dental consonants.
7. Not good at distinguishing [m] and [n].
8. Good on accented/unaccented vowels.
9. Surprisingly good at distinguishing between a very vigorous Russian [y] ([j] for many linguists) and the much weaker English off-glide which the letter y usually represents in English.

18

```
============ SESSION MODE MENU ============

SELECTION ============== FUNCTION ==========

        1       PRE-TEST

        2       TUTOR/REVIEW

        3       TUTOR/REVIEW/POST-TEST

        4       WORD-DRILL

        5       WORD-DRILL/TUTOR/REVIEW

        6       REVIEW LAST LESSON


            TYPE SELECTION NUMBER:  ▧
            (PRESS RETURN TO EXIT)
```

Screen showing VBLS options

10. Surprisingly good on distinguishing various varieties of central vowel (such as $[\wedge]$, schwa, and the Russian high central vowel).

11. Quite good on distinguishing between a true dental [n] and the English back sound pronounced in such words as bank, and sing.

12. Quite good on distinguishing a true dental [t] from the flap sound in the English word matter.

13. Variable on discrimination of palatalized vs. unpalatalized consonants, depending on position in word (and, apparently, particularly on how much the palatalization affects neighboring vowel sound). Discrimination seems better in final position than in some other positions. But very good at distinguishing between a soft consonant followed by a vowel and a soft consonant followed by an independent [y] before the vowel sound.

FOREIGN SYMBOL RECOGNITION

The current version of the VBLS system allows for foreign character sets or accent marks, but my experience to date is with version 1, which did not provide for foreign character sets.

One of my own greatest interests in the capabilities of the VET terminal and VBLS system is for use with students who take longer than their peers to make the connection between the foreign symbol and its related sound(s). In spite of the fact that I have had to do all of my experimentation to date using a transliteration (in the questions) and a specially devised transcription (in the answers), I am convinced that, in the long run, this is one of the areas in which the VBLS System has the greatest potential in foreign language study. Students who have worked with the lessons agree with me on this (although they were all at a more advanced stage of study than the lessons were intended for). The ability of the student to work at his own pace, repeating the material over and over (and in random order if the author has set the parameter values properly) should provide an excellent, low-pressure sort of drill for such students.

GRAMMAR DRILLS

It appears to me that the following are the only limitations which the System has with respect to grammar drills:

1. The answer must be no longer than 3-seconds.

2. The typed answer must require no more than 40 characters.

3. There must be only *one correct answer* for any one item—ambituity will lead only to student frustration.

4. In writing grammar drills, the teacher must keep in mind the limitations in discrimination of the system. For example, a drill in Russian contrasting the *unaccented* endings of the third-person singular and third-person plural verbs of the first conjugation ([-ait] vs. [-ayut] is unlikely to be productive.

Other than these limitations, the courseware author's own ingenuity is the limit. Any sort of substitution, transformation, fill-in-the-blanks, short-answer, translation (ugh!), or vocabulary (double ugh!) drill items, the answers to which fit the given limitations should prove to be effective. (Please forgive this author's prejudices about certain drill types expressed in that paragraph!) At least this author has found them so.

CONCLUSION

It seems clear to me that use of the System will help impress upon students the need for precision in foreign language learning. In spite of my awareness of certain problems, I am certainly sold on the capabilities of the System and hope to be able to use it actively in my teaching.

It does seem clear to me that if the System is to be used for pronunciation analysis and foreign character recognition, it is vital that the lessons be very carefully planned so as to add only one new element in each lesson, which will be mastered before moving on to another point. It will also be important that both teachers and students realize the capabilities and shortcomings of the system and be able to capitalize on what the system does well. It seems to me that it will be particularly valuable in cases where a foreign character set is involved, or where there is not a good phonetic fit between orthography and the sound system (as in French).

I also wonder whether the System may not have tremendous potential for working with dyslexic and learning-disabled students, allowing them to work intensively in a non-threatening atmosphere. ▣

Voice Input/Output Speech Technologies for German Language Learning

Harry S. Wohlert
Oklahoma State University

Direct interaction between a teacher and a student is generally considered necessary for learning a foreign language under formal conditions, and immediate feedback from the teacher is probably the most attractive feature of such an ideal situation. During a normal fifty-minute class session with twenty-five students, however, each student participates at most for only about two minutes and usually for much less time.

A limited solution to this obvious shortcoming is the traditional language laboratory. Here the students can listen and repeat foreign words and sentences without feedback; this means that the students themselves have to decide whether their responses are right or wrong. This constant and frustrating decision-making has been taken from the student, at least for reading and writing, with the introduction of computer-assisted instruction (CAI). While the effectiveness of CAI in terms of motivation and achievement is fairly well documented by now,[1] so far it has not involved the other two aspects of language learning—speaking and listening. To recognize speech, computers need special devices to be able to analyze a student's spoken words, or "voice input." Speech recognition devices use a computerized process that analyzes sounds and stores them as patterns of electronic signals, so-called templates, in the computer's memory. When the computer can match the template with the pattern of a student's vocal input through a microphone, speech recognition occurs.

Until recently, however, the cost of speech recognition systems, with prices ranging to over $65,000, made their use in CAI prohibitively expensive.[2] During the last few years, however, a number of less expensive systems have become commercially available,[3] and thus more accessible for use in secondary schools and colleges. At the 1982 National Educational Computer Conference (NECC), for example, Scott Instruments introduced a recognition system for under $1,000 which the company lists now at $895. This voice-based learning system (VBLS™) expands CAI in a limited but effective way into the area of speaking and listening.[4]

The Scott Instruments Vet-2 uses the recognition technique described above, which compares the student's voice pattern to stored vocabulary. The recognition system limits the template for voice input to forty characters with a maximum duration of three seconds. In other words, rather than speaking continuously, the speaker builds a frame around the input by pausing before and after the word or sentence to be matched to the template.

The device itself is a small box (10 x 8 x 2 inches) that comes with an interface board for the Apple or Franklin computers, a headset with a highly

directional microphone, a software package, and documentation. The small, lightweight microphone is positioned not more than one inch from the speaker's mouth to ensure a response to normal speech volume. This intentionally low sensitivity of the microphone allows the speech recognition device to be used with a moderately noisy background. The program provides a step-by-step approach that allows anyone with a basic understanding of computers to program self-designed lessons. The teacher can store these lessons on a separate library disk, or create a study disk that holds both the main programs and the lessons. The manual is comprehensive and written with a minimum of computer jargon for the novice computer user.[5]

For voice output, the author purchased a cassette control device for $79.00 from Hartley Courseware and a cassette recorder with an external microphone input.[6] The control device plugs into the Apple's game paddle port and the external microphone input of the cassette. The software

Figure 1: Apple II with Vet-2 Speech Recognition Unit and Voice Output System
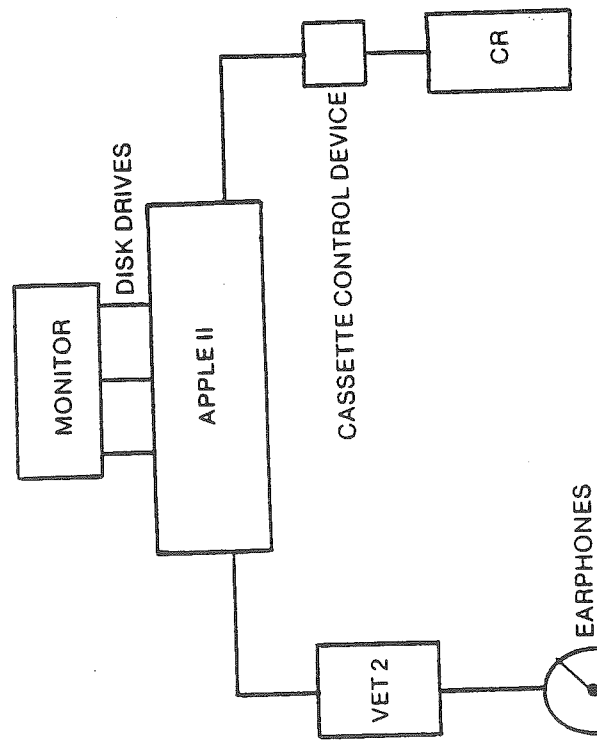
then controls the cassette recorder so that the computer can turn it on and off at appropriate intervals.

When looking at any new system, an educator will want to know how it can be used in the classroom, and how effective it is from a pedagogical point of view. To answer these questions, the author selected for use in

a pilot VBLS program one hundred sixty of the most commonly used German strong verbs. These verbs, according to an informal survey conducted by the author, were never mastered by even most of the advanced students. The verbs were divided into sixteen groups of ten verbs each:

Group 3
empfehlen   empfahl   empfohlen   to recommend
fahren,ä   fuhr   gefahren   to drive
etc.

For each group of ten verbs—actually forty forms with the infinitive, past tense, past participle, and translation—five lessons were created. The following representations of the monitor screen in Figures 2 to 6 show examples of each lesson in the review phase.

Lesson I—The student should drill all three principle forms:
Figure 2

Say the German equivalent:

recommend   recommended   recommended

Lesson II—The student should drill the past tense:
Figure 3

Say the German equivalent:

He recommended

Lesson III—The student should drill the present perfect:
Figure 4

Say the German equivalent:

He has recommended

Lesson IV—The student should use the past tense in context:
Figure 5

Say the complete sentence in German:

Er _____ dem Freund das Buch.
a. sprang
b. empfahl
c. sank

Lesson V—The student should use the present perfect:
Figure 6

Say the complete sentence in German:

Er _____ dem Freund das Buch _____.
a. ist—geschrieben
b. hat—gebeten
c. hat—empfohlen

For each verb the programmer, a native speaker, read the German text five times into the system which then averaged these sounds into one template. The author also recorded two of these verbal inputs simultaneously

onto a cassette. The simultaneous recording is of special significance because during playback, it presents the template's pronunciation to the students for imitation of pitch, intensity, speed, and pauses. While the recognition device will not analyze phonemes but only syllables, it does consider the above-mentioned parameters. In fact, the influence of the pitch on the pattern to be analyzed made it necessary to create templates with a female voice for women with a higher pitch.

The Language Laboratory Session

When students insert the disk into the disk drive, the cassette into the recorder, and turn on the system, they are presented with the following menu:

1. PRE-TEST
2. TUTOR/REVIEW
3. TUTOR/REVIEW/POST TEST
4. WORD DRILL
5. WORD DRILL/TUTOR/REVIEW
6. REVIEW LAST LESSON

Assuming the students select the standard session, the Tutor-Review mode, they will choose the desired lesson and do the kind of work illustrated below:

Tutoring Mode

```
HELLO MARY!                               (system greeting)
LESSON IS: Strong Verbs 3—Past Tense      (lesson description)
PRESS - PLAY - ON THE CASSETTE            (instruction for voice output)
                                          (response request)
SAY "Er empfahl"                          (computer plays tape)
    "Er empfahl"                          (student speaks)
    "Er empfahl"                          (second response request)
SAY "Er empfahl"                          (student speaks)
    "Er empfahl"                          (tutor step
SAY THE GERMAN EQUIVALENT:                    with
    He recommended                        correct answer displayed)
ANSWER: _____                        (computer plays tape)
    "Er empfahl"                          (student speaks)
    "Er empfahl"
```

When the student's response is not accepted, he or she is prompted to try again. Therefore, the computer will not proceed to the next question until the above sequence is completed.

Review Mode

```
SAY THE GERMAN EQUIVALENT:                (question 1
    He recommended                        without
SAY ANSWER: _____                    display of correct answer)
    "Er empfahl"                          (student speaks)
```

If the response is not accepted, the student will be prompted to try again. The number of attempts can be set by the instructor. The correct answer

will be displayed if all attempts fail. After all ten questions have been presented, the student's score is also displayed. At the end of the review mode, students rewind the cassette to a specific cassette footage location for the lesson. The computer-to-cassette interface used cannot rewind the tape.[7]

Another interesting use for the VET-2 is pronunciation practice (see Fig. 7), which is called Word Drill on the VBLS menu.

Figure 7. Real-Time Pronunciation Feedback

WORD DRILL

```
5  6  7  8  9  1
0  0  0  0  0  0
0  0  0  0  0  0
                 0
```

Score: 820

SAY 'Sie empfahl'

ON
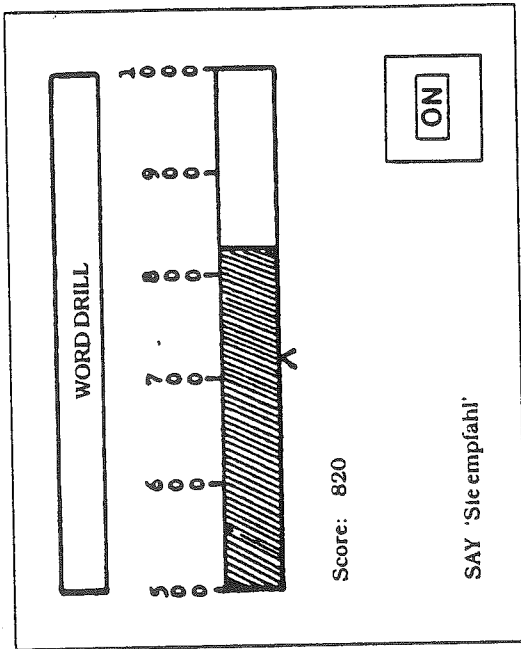
In this exercise, the student repeats a sentence into the microphone, and a solid column advances on the rating scale to indicate a score for the pronunciation. Since the perfect match of 1000 cannot even be obtained by the speaker who created the template, the teacher must select for the student a range representative of good pronunciation. The author used 750 to 1000 for most exercises.

The pointer (caret) below the scale indicates the student's previous high score, and is erased when the program advances to the next word of the sentence.

Evaluation

The VBLS voice-based learning system as used here integrates verbal tutoring from computer-controlled cassette recorders, visual messages, and oral responses with real-time pronunciation feedback, creating an instructional technique that has proven superior to learning in only one of the above modes.[8]

According to a survey of recent research by Wagers and Horn at Scott Instruments,

the amount of learning and recall of information that is associated with exercises in pronunciation exceeds that acquired in a

silent environment. Research has shown that pronunciation serves to decode written words into their oral forms and, thereby, elicit recognition (meaning) in response to the displayed word.[9]

A study of the effectiveness of the VBLS program conducted at this university during the summer of 1983 also revealed that students work longer with the VBLS system in the language laboratory than when they study in the traditional fashion at home.

The sixteen students, mostly graduate students with majors in the humanities, were asked to study all forms of twenty German strong verbs for ninety minutes with the VBLS program described earlier and another set of twenty verbs for an equivalent time at home. They were further informed that all forty verbs would later be part of a regular graded quiz. A nongraded quiz administered the day before the regular quiz showed that the group mastered 87% of the verbs studied on the computer and only 67% of those studied at home. The large difference immediately suggested that some of the students did not follow instructions. This impression was also confirmed with follow-up personal interviews. Seven of the students had indeed worked for ninety minutes (three sessions) on the computer but only fifteen to thirty minutes instead of the suggested fifty minutes at home. Because of time lost through checking out disks, loading the program, and other factors, the equivalent time for studying at home was calculated at fifty minutes.

Surprisingly, although fourteen of the students had never used a computer before and were on the whole less than enthusiastic about studying the verbs on a machine, they obviously came to prefer this approach. This attitude and its change is probably best described by quoting one of the evaluations which all students were required to write at the end of the study:

I am very impressed by the results, even though they may need corroboration. Undoubtedly they show the machine to be superior to my studying at home. In six sessions I almost overcame my personal aversion to that type of machine. I am absolutely convinced that it will be greatly appealing to young people and that the combined sensory method it uses is preferable.

The anonymously written evaluations ranged from slightly negative, like the above statement, to very positive:

I felt that the voice recognition unit was extremely helpful, especially in comparison with studying at home. I am basing this judgment on the results of my test which indicated that after approximately 5½ sessions with the computer I made one error on that material as compared to the homework where I made 11 errors after studying 90 minutes. I would be interested in studying with a computer in the future, for I feel this studying uses my time more efficiently and helps with retention. Thank you for the experience.

It should be added that the results of the study with the seven subjects eliminated still favored learning with the computer. A significant difference between the two groups was observed (t = 3.65, df = 8, p = .0065).

The major criticisms offered by the students dealt with the manual operation of the system and its inability to distinguish accurately certain sounds.

While the more technically adept students learned to operate the system in less than ten minutes, others needed up to three sessions of thirty minutes each. The difficulties centered almost exclusively on the use of the cassette recorder. When, for example, the students were presented with a response request in the Tutor mode, the computer turned on the recorder. The user was supposed to listen to the pronunciation and then press the spacebar on the keyboard. This turned the microphone on and the cassette off. Frequently, some of the users often waited to study the visual presentation first, and the recorder continued to play the next sentence or word. The audio then no longer matched the next visual presentation since it was out of synchronization. Most students quickly learned to recover from this situation by stopping the recorder until the video caught up with the sound. This problem rarely occurred during the later sessions.

The only problem associated with the operation of the standard VBLS system was the positioning of the microphone. As mentioned previously, the Shure unidirectional microphone ignores most sounds originating more than one inch away. Whenever students considered the program "hanging," i.e., it did not advance to the next step, they either had spoken too low or the microphone was positioned too far from the mouth.

The state of the art in speech recognition and the level of the student's language competency are the two main factors determining the acceptance of wrong or mispronounced words. The Vet-2 does not distinguish unstressed vowels very well, strongly observes the rhythm of a sentence, and similarly to the human ear confuses certain consonants or consonant clusters. However, the accuracy of recognition can be increased greatly, for example, by avoiding alternative answers in a multiple choice problem with sounds that are similar to those in the correct answer, especially when both have dominant letters of the alphabet containing the /e:/ sound (b,d,g,e,t,c,v,p,z). Another method of increasing the accuracy is a simple adjustment of the unit's threshold level. The menu on the authoring disk will guide the teachers to a routine that prompts them to type a number from zero to two hundred for the acceptance threshold. Since the higher number lowers the threshold, the author used 180 for beginning students but 100 to 130 for the most advanced users. It is highly recommended that the teacher take the lowest number possible because students react very negatively to a threshold that its too low. On the other hand, no complaints were registered about the Vet-2 rejecting pronunciations too frequently. After working with a tight threshold for a period of time, intermediate German students observed that some of the less advanced students in the class showed a "dramatic" improvement in pronunciation.

The limitations of speech recognition elicited some critical comments

from students, but did not seem to have diminished their enthusiasm for the Vet-2 nor did it keep them from obtaining a high level of achievement when the device was used with a cassette recorder and the programs described earlier. In the author's opinion, the VBLS system is probably the best one available for under $1,000 and can be considered a very useful pedagogical tool even with the limitations mentioned. Comments from teachers who have worked with the Vet-2 as well as evaluations from reviewers looking for commercial applications strongly support this view.[10]

The Scott Instruments voice recognition device can therefore be considered a desirable complement to computer-assisted instruction, especially for students in their first two years of German study. As the use of speech recognition devices grows, other imaginative programs will certainly become available to challenge even the most advanced students.

Notes

[1] James A. Kulik et al., "Effects of Computer-Based Teaching on Secondary School Students," Journal of Educational Psychology, 75, No. 1 (1983), pp. 19-26.

[2] Datapro Research Corp., 1805 Underwood Boulevard, Delran, NJ 08075, sells a twelve-page report "All About Speech Technology" for $19. It includes a succinct discussion about speech recognition and synthesis and lists 71 products with prices as well as technical data.

[3] Mike Rigsby, Verbal Control with Microcomputers (Blue Ridge Summit, PA: Tab Books Inc., 1982).

[4] The speech recognition unit is available from Scott Instruments, 1111 Willow Springs Dr., Denton, TX 76201. (817) 387-9514.

[5] The reference manual can be purchased from Scott Instruments for $15. It describes the use of the Vet-2 with the Dukane Cassette A-V Matic Filmstrip Projector Model 28A3B, the Kodak Carousel™ Projector Model 760H, and the Epson Dot Matrix Printer MX-80. Many other functions of the VBLS program, for example, record keeping or disk copying not discussed in this article are described in detail.

[6] The cassette device can also be used with many other CAI programs. It is available from various software vendors or Hartley Courseware, Inc., Box 431, Dimondale, MI 48821. (616) 942-8987.

[7] Tandberg sells a fast random-access recorder for approximately $1,300 and, if money is no object, one can purchase from INSTAVOX, Champaign, IL, a $5,000 unique record/playback system that uses a very large floppy disk to store the sound. The INSTAVOX is fully compatible with the Vet-2 and gives instant random access or playback through an external speaker.

[8] Will Wagers and Karin Horn, The VBLS™ Voice-Based Learning System Preliminary Manual (Denton: Scott Instruments, 1982) pp. 7-9; and Phillip M. Tell, "The Role of Certain Acoustic and Semantic Factors at Short and Long Retention Intervals," Journal of Verbal Learning and Verbal Behavior, 11 (1972), pp. 455-64.

[9] Donald A. Walter, "The Effect of Sentence Context on the Stability of Phonemic and Semantic Memory Dimensions," Journal of Verbal Learning and Verbal Behavior, 12 (1973), pp. 185-92.

[10] William Barden, Jr., "Speech Synthesis and Recognition," Popular Computing, 1, No. 11 (Sept. 1982), p. 129.

"Outline 1" – outline program
for creating BASIC programs
with Victor 9000 VOICE

```
10 REM : program name "OUTLINE1," bare-bones voice program outline
20 REM : Prepared August 1984
50 REM : initialize the CODEC strings, format XYZ$ = "XYZ"
200 REM : initialize the CODEC
210 PLAY.NOTE%  = 1
212 RECORD.SND% = 2
214 PLAY.SND%   = 3
220 LOAD.LIB%   = 4
230 SAY%        = 5
240 FREE.LIB%   = 6
250 PAUSE%      = 7
260 PAUSE.LEN%  = 500
300 DEF SEG = 0
310 LOWOFF  = PEEK(880)
320 HIOFF   = PEEK(881)
330 LOWSEG  = PEEK(882)
340 HISEG   = PEEK(883)
350 CODECSEG = (256 * HISEG) + LOWSEG
360 CODEC    = (256 * HIOFF) + LOWOFF
370 DEF SEG = CODECSEG
400 FILE.NAME$ = "
410 CALL CODEC(FILE.NAME$, LOAD.LIB%)
1790 CALL CODEC(FREE.LIB%)
2000 END
```

You do not need to open voice libraries to use the PLAY_NOTE function (as in line 830 through 850).

4.5 EXPLANATION OF THE CALC PROGRAM

Lines	Description
10 - 20	Print a banner at the beginning of the program.
50 - 150	Initialize the constants that will hold the names of the keys. The keys in the voice file and the words they contain are:

SAY0	"Zero"
SAY1	"One"
SAY2	"Two"
. .	. .
SAY9	"Nine"
PLEASE	"Please enter"
OPERAND	"Operand"
INVALID	"Invalid"
SUM	"The sum is"

SAYNUM$ is an array whose index corresponds to the key for that number. That is, SAYNUM$(0) = "SAY0", SAYNUM$(1) = "SAY1", etc.

Also, 130-150 combine some of the phrases with spaces between them, so that you can say a combination of the phrases with only one CALL.

You should keep the "header" information in the file current. To do this, press MENU (function key 1) to get to the DISK menu, then select FILES (function key 3), then select the HEADER option (function key 3), and fill in the areas specified by the Voice Editor.

4.4 APPLICATION PROGRAMS WITH THE VOICE KERNEL

The CALC program (listed in Appendix C) is a simple example of the power of the Voice Kernel. Even if you plan to use other programming languages with the Voice Kernel, this example is applicable because it shows the basic requirements. Due to some of the limitations of MS-BASIC, however, there are some steps (such as setting constants to the values for the CALL statements) that are not required in languages such as Pascal and PL/M.

Most applications you write will only use sounds previously stored on disk with the Voice Editor; and you will know the names of these sounds when you write the program. Thus, the functions you use could be limited to LOAD_LIBRARY, PLAY_PHRASE, and FREE_LIBRARY (functions 4, 5, and 6). The CALC program uses two more functions, PLAY_NOTE and PAUSE (functions 1 and 7) to make a little music, and to pause between speaking the prompt and speaking the numbers entered.

Before you use a sound library, you must open the library with the LOAD_LIBRARY function, as in line 370 of the CALC program. Once a library is open, you can send any of its sounds out the speaker with the PLAY_PHRASE function (such as lines 480 and 490). Be sure to close the library before returning to the operating system (such as on line

180 – 270   Initialize the constants that hold the
            function numbers (PLAY.NOTE%,
            LOAD.LIB%, etc.), and the numeric
            constants used in calling PAUSE and
            PLAY. Also specify the tones used with
            PLAY_NOTE.

280 – 350   Initialize the segment for the AUDIO
            routines. This is the same code that
            is given in Chapter 3.

360 – 370   Open the voice file ("CALC").

400 – 410   Initialize the interim variables and
            the sum.

450 – 510   Request the first operand by printing a
            message on the screen, and speaking
            "Please enter operand one." Be sure to
            pause for half a second (in case the
            user is typing ahead), or else the word
            "one" will run together with the first
            number typed. Call the subroutine to
            accept operand 0, and place the result
            in operand 1.

520 – 580   Repeat the previous steps for operand 2.

590 – 700   Find the sum, type and say "The sum
            is", pause for half a second, and type
            and say each number of the sum. Line
            660 starts at position 2 of SUM$ since
            the first character of the string is
            always a space. When all the numbers
            are printed and spoken, start the loop
            again.

C. THE CALC PROGRAM LISTING

730 - 930 This subroutine accepts characters
typed, and decides what the user wants.
The number given is placed in operand
0.

If a number is typed, operand 0 is
multiplied by 10 and the number is
added to it. The number is printed and
spoken, and the subroutine waits for
the next character. If a <RETURN> is
typed, indicating the end of the
number, a short set of tones (the notes
C, E, and G) is played, and the
subroutine exits. If an "X" or "x" is
typed, the CALC voice library is freed
and the program ends. If any other
character is typed, the words "Invalid
operand" are spoken and typed, the CALC
voice library is freed, and the program
exits.

```
10 PRINT : PRINT "Verbal Adding Machine"
20 PRINT: PRINT "Enter an X to exit"
30 '
40 '    Initialize the CODEC strings
50 DIM SAYNUM$(9)
60 FOR I% = 0 TO 9
70    SAYNUM$(I%) = "SAY" + MID$(STR$(I%), 2, 1)
80    NEXT I%
90 PLEASE$ = "PLEASE"            ' name of "Please enter"
100 OPERAND$ = "OPERAND"         ' name of "Operand"
110 INVALID$ = "INVALID"         ' name of "Invalid"
120 THE.SUM.IS$ = "SUM"          ' name of "The sum is"
130 PLEASE.1$ = PLEASE$ + " " + OPERAND$ + " " + SAYNUM$(1)
140 PLEASE.2$ = PLEASE$ + " " + OPERAND$ + " " + SAYNUM$(2)
150 INVAL.OPER$ = INVALID$ + " " + OPERAND$
160 '
170 '    Initialize the CODEC
180 PLAY.NOTE% = 1               ' Function 1
190 LOAD.LIB% = 4                ' Function 4
200 SAY%  = 5                    ' Function 5
210 FREE.LIB% = 6                ' Function 6
220 PAUSE% = 7                   ' Function 7
230 PAUSE.LEN% = 500             ' Duration of PAUSE
240 PLAY.DUR% = 50               ' Duration of PLAY NOTE
250 PLAY.C% = 262                ' C note (in Hertz)
260 PLAY.E% = 330                ' E note (in Hertz)
270 PLAY.G% = 392                ' G note (in Hertz)
280 DEF SEG = 0
290 LOWOFF = PEEK(880)
300 HIOFF = PEEK(881)
310 LOWSEG = PEEK(882)
320 HISEG = PEEK(883)
330 CODECSEG = (256 * HISEG) + LOWSEG
```

```
340 CODEC     = (256 * HIOFF) + LOWOFF
350 DEF SEG   = CODECSEG
360 FILE.NAMES = "CALC"
370 CALL CODEC(FILE.NAMES, LOAD.LIB%)
380 '
390 '      Initialize the calculator
400 DIM OPERAND%(2)
410 OPERAND%(1) = 0 : OPERAND%(2) = 0 : SUM% = 0
420 '
430 '      Main loop
440 '
450 PRINT : PRINT "Enter operand 1: ";
460 '
470 '      Say "Please enter operand 1"
480 CALL CODEC(PLEASE.1$, SAY%)
490 CALL CODEC(PAUSE.LEN%, PAUSE%)
500 GOSUB 710
510 OPERAND%(1) = OPERAND%(0)
520 PRINT : PRINT "Enter operand 2: ";
530 '
540 '      Say "Please enter operand 2"
550 CALL CODEC(PLEASE.2$, SAY%)
560 CALL CODEC(PAUSE.LEN%, PAUSE%)
570 GOSUB 710
580 OPERAND%(2) = OPERAND%(0)
590 SUM% = OPERAND%(1) + OPERAND%(2)
600 PRINT : PRINT "The sum is:    ";SUM%
610 '
620 '      Say each peice of the sum
630 SUM$ = STR$(SUM%)
640 CALL CODEC(THE.SUM.IS$, SAY%)
650 CALL CODEC(PAUSE.LEN%, PAUSE%)
660 FOR I% = 2 TO LEN(SUM$)
670 NUMBER% = VAL(MID$(SUM$, I%, 1))
680 CALL CODEC(SAYNUM$(NUMBER%), SAY%)
690 NEXT I%
700 GOTO 450        '   Go to main loop
710 '
720 '      Get the operand into OPERAND%(0)
```

```
730 OPERAND%(0) = 0
740 INPUT.CH$ = INPUT$(1)
750 '    Did they hit a number?
760 IF INPUT.CH$ < "0" OR INPUT.CH$ > "9" THEN GOTO 810
770 OPERAND%(0) = ( OPERAND%(0) * 10 ) + VAL(INPUT.CH$)
780 PRINT INPUT.CH$;
790 CALL CODEC(SAYNUM$(VAL(INPUT.CH$)), SAY%)
800 GOTO 740
810 '    Did they hit <RETURN>?
820 IF ASC(INPUT.CH$) <> 13 THEN GOTO 870
830 CALL CODEC(PLAY.C%, PLAY.DUR%, PLAY.NOTE%)
840 CALL CODEC(PLAY.E%, PLAY.DUR%, PLAY.NOTE%)
850 CALL CODEC(PLAY.G%, PLAY.DUR%, PLAY.NOTE%)
860 RETURN
870 '    Did they hit X or x?
880 IF ASC(INPUT.CH$) = 88 OR ASC(INPUT.CH$)
    = 120 THEN GOTO 920
890 '    Must have entered an invalid operand
900 PRINT " Invalid operand"
910 CALL CODEC(INVAL.OPER$, SAY%)
920 CALL CODEC(FREE.LIB%)
930 END
```

Fill In The Blanks

by Allen Patterson

A fill-in-the-blanks program for computer assisted instruction. This particular version of the program, which will run on any Commodore computer, helps students learn the correct forms of French verbs. But the program can be modified to accommodate many other applications. For any computer except the Commodore 64, delete line 98 in the program listing.

One of the most valuable assets that computers bring to education is their ability to supply immediate feedback in a non-threatening manner. However, if a new program has to be written (requiring valuable teacher's time) for every new skill that a student is expected to master, the value of the computer diminishes. In addition, in order for the computer to be truly effective in the classroom, it should present material consistent with other educational methods that have withstood the test of time. For example, many educators have relied upon a "fill in the blanks" type of exercise to reinforce learning, provide practice and review material. The computer can quite easily take this proven educational strategy and improve on it. Not only will the computer reward the student for correct responses but it will present the questions in a random order with the possibility that questions not answered correctly could be repeated. Alternately, these incorrectly answered questions could be recorded on paper for future reference.

The following program is set up so that the "fill in the blanks" sentences are located in data statements and can be changed at any time—by anyone. In the example that follows, the correct form of the French verbs etre, avoir, or aller are to be inserted. This

```
9  REM     ***********************
10 REM     ** FILL IN THE BLANKS **
15 REM     ***********************
20 :
30 :
50 REM        THIS PROGRAM WRITTEN BY
60 REM        ALLEN PATTERSON 83/3/24
61 :
62 REM BOX 178, BRAESIDE, ONTARIO
65 REM CANADA   K0A 1G0 (613)623-6867
70 :
75 REM   COPYRIGHT (C) 1983
78 :
80 REM TO ENTER DATA--FIRST RUN PROGRAM
81 REM AND PUSH STOP BUTTON SO THAT YOU
```

```
82 REM WILL HAVE UPPER AND LOWER CASE
83 REM LETTERS.
84 :
85 :
98 POKE 59468,14
99 NU=25
100 D$="[HOME,DOWN6]":DIM F(NU),F$(NU),Q$(NU),AN$(NU),AW$(NU)
110 FOR S=1 TO NU:READ Q$(S),AN$(S):NEXT S
145 TT$="ETRE, AVOIR, ET ALLER"
150 PRINT"[CLEAR]";D$;TAB(LEN(TT$)/2);TT$
160 PRINT"[DOWN2]ECRIVEZ LA FORME CORRECTE DU VERBE DANS LE TIRET."
165 GOSUB 600:REM IF STUDENT CHOOSES #  OF QUESTIONS THEN USE:GOTO550
170 :
200 J=J+1:A$="":IF J>NU THEN 1000
205 REM  IFJ>NE THEN 1000:REM USE THIS LINE IF STUDENT SELECTS # OF
    QUESTIONS
210 K=INT(RND(1)*NU+1):IF F(K)=1 THEN 210
220 F(K)=1:F$(J)=Q$(K):AW$(J)=AN$(K)
230 :
240 B=B+1:X$=MID$(F$(J),B,1):IF X$="*"THEN X=B-1:B=0:GOTO 260
250 GOTO 240
260 PR$=LEFT$(F$(J),X)+" --------- "+RIGHT$(F$(J),LEN(F$(J))-(X+1))
262 PRINT"[CLEAR]"
300 IF LEN(PR$)<40 THEN PRINT D$;PR$:GOTO 400
305 I=40
310 I=I-1:X$=MID$(PR$,I,1):IF X$<>" "THEN 310
320 Y=I
330 PRINT D$;LEFT$(PR$,Y):PRINT"[DOWN]";RIGHT$(PR$,LEN(PR$)-Y)
350 :
400 GET AN$:IF AN$<>""THEN 400
405 GET AN$:IF AN$=CHR$(13)THEN 500
410 IF AN$=""THEN 405
412 IF AN$=CHR$(20)OR AN$=" "THEN 420
413 IF AN$>CHR$(192)AND AN$<CHR$(219)THEN 420
415 IF AN$<CHR$(65)OR AN$>CHR$(90)THEN 405
420 A$=A$+AN$
425 IF LEN(A$)>10 THEN 500
426 IF AN$=CHR$(20)AND LEN(A$)=1 THEN A$="":GOTO 405
430 PRINT D$;TAB(X+1);"[RVS]";A$
435 IF AN$=CHR$(20)THEN A$=LEFT$(A$,LEN(A$)-2)
    :PRINT D$;TAB(X+1);"[RVS]";A$;CHR$(148)
440 GOTO 405
450 :
500 IF A$=AW$(J)THEN PRINT"[DOWN6,RVS]CORRECT![RVOFF]":R=R+1:
    GOSUB 600:GOTO 200
```

programmer's tips

```
510 PRINT"[DOWN3,RVS]INCORRECT[RVOFF,SPACE]-- THE ANSWER IS: ";AW$(J)
512 IF LEN(PR$)<40 THEN PRINT"[DOWN]";PR$:PRINT"[UP]";TAB(X+1);
    "[RVS]";AW$(J):GOTO 517
514 PRINT"[DOWN]";LEFT$(PR$,Y):PRINT"[DOWN]";RIGHT$(PR$,LEN(PR$)-Y)
516 PRINT"[UP3]";TAB(X+1);"[RVS]";AW$(J)
517 REM:    F(K)=0:J=J-1:REM USE THIS LINE TO HAVE INCORRECT
    QUESTIONS REPEATED
520 GOSUB 600:GOTO 200
600 PRINT"[DOWN4,RIGHT7]PUSH [RVS]SPACE BAR[RVOFF,SPACE]TO CONTINUE"
605 GET G$:IF G$<>""THEN 605
610 GET G$:IF G$<>" "THEN 610
615 PRINT"[CLEAR]"
620 RETURN
680 :
690 REM   DATA GOES HERE: PUT QUOTATION MARKS AROUND QUESTIONS WITH
    A COMMA
693 :
694 REM    PUT QUESTION THEN COMMA THEN ANSWER
695 :
696 :
700 DATA"TU*L'AMI DE GEORGES?",ES
710 DATA"LA FILLE*FAIM. OU SONT LES SANDWICHS?",A
720 DATA"MONSIEUR LEBLANC*DANS LE RESTAURANT.",EST
730 DATA"NOUS*DINER A MIDI.",ALLONS
740 DATA"J'*CINQ ANS. QUEL AGE AS-TU?",AI
750 DATA"OU EST-CE QUE VOUS*?  JE VAIS A L'ECOLE.",ALLEZ
760 DATA"LES GARCONS*TRES GENTILS.",SONT
770 DATA"MAMAN*DEVANT LA MAISON AVEC PAPA.",EST
780 DATA"JE*TRES CONTENT QUAND IL NEIGE.",SUIS
790 DATA"MADAME, VOUS*LA SOEUR DE MADAME LEBRUN.",ETES
800 DATA"TU*JOUER AU HOCKEY APRES LES CLASSES?",VAS
820 DATA"LE CHIEN*A COTE DE LA MAISON.",EST
830 DATA"LES STYLOS DE MONSIEUR*SUR SON BUREAU.",SONT
840 DATA"ELLE*AU PARC POUR NAGER.",VA
850 DATA"ILS*SOMMEIL PARCE QU'IL EST DEUX HEURES DU MATIN.",ONT
860 DATA"JACQUELINE ET MOI*VISITER LA VILLE DE MONTREAL.",ALLONS
870 DATA"PIERRE ET GEORGES*LES FRERES DE SUZANNE.",SONT
880 DATA"TOI, TU*MON CHANDAIL, N'EST-CE PAS?",AS
890 DATA"ELLES*CHANTER A LA SOIREE.",VONT
900 DATA"GEORGES N'*PAS DE SOEURS.",A
910 DATA"NOUS*DANS LA MEME CLASSE QUE MARIE.",SOMMES
920 DATA"JE*PARLER AU DOCTEUR.",VAIS
930 DATA"CHANTAL ET MOI, NOUS*DE TUQUES BLEUES.",AVONS
940 DATA"VOUS N'*PAS DE SOULIERS.",AVEZ
950 DATA"ELLE*RESTER A LA MAISON PARCE QU'ELLE EST MALADE.",VA
```
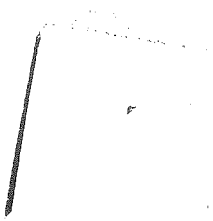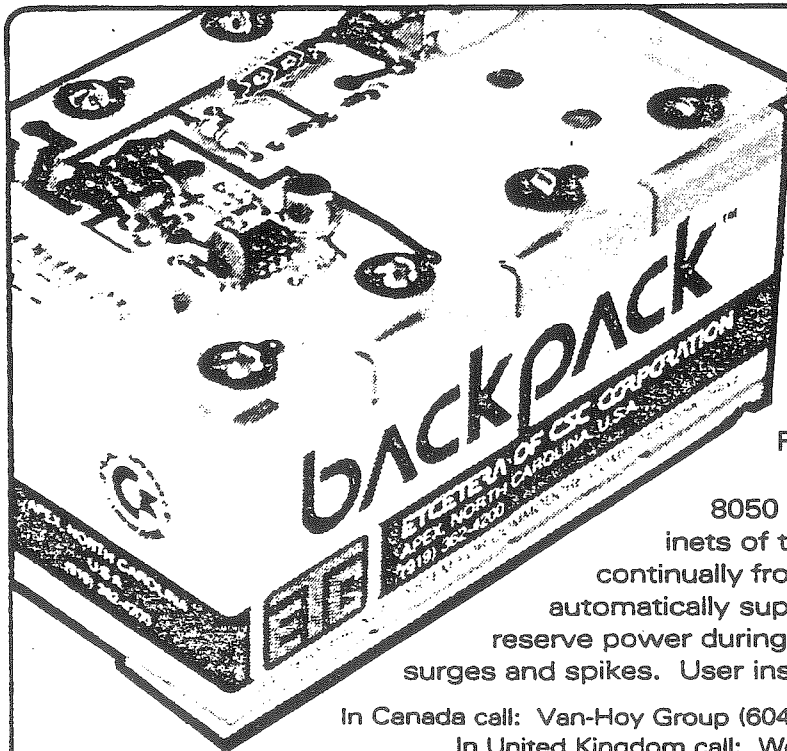
programmer's tips

```
165 GOTO 550
205 IF J>NE THEN 1000
550 PRINT"[DOWN3]HOW MANY QUESTIONS WOULD YOU LIKE TO TRY?";
560 GET K$:IF K$<>""THEN 560
565 H$=""
570 GET K$:IF K$=CHR$(13)THEN 580
571 IF K$=""THEN 570
572 H$=H$+K$:IF LEN(H$)>2 THEN 580
575 PRINT K$;:GOTO 570
580 NE=VAL(H$):IF NE>0 AND NE<26 THEN 590
585 PRINT:PRINT"CHOOSE A NUMBER BETWEEN 1 AND ";NU:FOR T=1 TO 1000:
    NEXT T :GOTO 150
590 GOTO 200
```

As you can see, this program would be very useful and very adaptable. In fact, many of the above subroutines would fit nicely into other programs of your own.   C

TWO COMPUTER PROGRAMS FOR THE LANGUAGE STUDENT

Robert L. Roseberry

ABSTRACT

Notebook computers, such as the TRS-80 Model 100 and the NEC 8201, provide both the students and the teacher with the unique opportunity of bringing into the classroom a truly portable, self-contained computer with integrated, non-volatile memory. Two programs that are useful to the language student and that take full advantage of the unique features of these computers are the Course Authoring Program, which allows the teacher to create computerized tutorial lessons without recourse to programming languages, and FLASH, a computerized flash card manipulator. Both programs utilize text files that are created by the computer's built-in editor and can, with minimal effort, be converted for implementation on any computer system for which a text editor and appropriate memory storage facility are available.

Perhaps more than most post-secondary disciplines, language study makes frequent use of drill-and-practice exercises. In this article I shall present two powerful, yet simple, drill-and-practice programs that are designed for implementation on both the TRS-80 Model 100 and the NEC 8201 computers. These programs, which I place in the public domain, can, with the aid of a modest knowledge of BASIC, be implemented on your own computer system. I have chosen the TRS-80 Model 100 (and the almost identical NEC 8201) for several reasons: they are small and light, having the shape, size, and weight of a medium-sized book; they have non-volatile memory, a built in text-editing program, and a high de-

gree of program integration, enabling several programs to reside in memory simultaneously and to interact with each other. As the prices of such notebook computers decrease, they will no doubt become as common in university classrooms as pocket calculators are now. If your computer system does not have the various facilities listed above, you will probably need the following hardware and software in order to make use of the programs described below: (1) a high-speed peripheral memory-storage device (such as a disk drive), and (2) a text-editing program that creates standard text files (such as a word-processing program).

THE COURSE-AUTHORING PROGRAM (CAP)

The first of the two programs, the Course Authoring Program (CAP), listed in Figure 1, is based on a simple algorithm (see program description) and is of value to language study, though by no means restricted to it. The lesson consists of a series of text files, each of which must contain a question and answer set. In addition, the file may (but need not) contain text. In this way a lesson can be constructed in which any desired amount of text may precede a question, or any desired number of questions may follow a given text. Each file name must be prefixed with the letter Q, and the files must be numbered consecutively. The text editor will automatically give the suffix .DO to each of these files. In addition, the first file (Q 1.DO) must contain, as its

Robert L. Roseberry received his A.B. in German from the University of California, Berkeley in 1968 and his M.A. (1970) and Ph.D. (1974) in German from the University of Toronto. After several years teaching German in Canada and Nigeria, he returned to university (British Columbia) to do an M.Ed. in English Education (1978), specializing in English as a second language, a subject that he taught at Vancouver Community College until 1981 when he took up his present job as Coordinator of the Remedial English Program at Okanagan College, Kelowna, British Columbia.

first item, the total number of Q files in the lesson.

In writing the lesson, the author must enter the computer's built-in text-editing program and create the necessary files. The text editor requires a file name before the typing can begin, so the author must type in Q 1 and press

ENTER. Note the essential space between the letter Q (for Question) and the number 1 (for section 1). In the first line of the first file (Q 1.DO) the total number of files in this lesson must appear, followed by ENTER. (The maximum number of files per lesson is 17.) On the second line three correct answers to the question that will be contained in that file must appear. Use commas (and no spaces) to separate the answers. If there is only one correct answer, enter it three times. If there are two correct answers, enter one of them twice. There must be three answers on this line, or the program will not operate properly. The first of the three answers should be the preferred one, since the computer will show this answer to a student who answers incorrectly. It is advisable to enter the answers in upper-case letters. Later, when the student runs the program, the CAPS LOCK key should be depressed to help facilitate an answer match. Next, you may type in as much or as little text as desired. (By omitting the text, you may line up several questions in succession following the previous text page.) If there is too much text to view at once, the CAP program will present the text seven lines at a time on the computer's eight-line LCD screen. When the program is running, the student will turn the text pages by pressing ENTER. Text is optional; whether or not there is any text in the file, the file must be concluded with the question whose answers have already been typed in. At the end of the file press ENTER twice. Subsequent files (Q 2.DO, Q 3.DO, etc.) are created in the same way except that these files begin with the three answers to the current question; the section count is given only at the top of Q 1.DO.

When the several Q files have been completed and stored in non-volatile RAM memory under their designated names, one further file must be written in order to complete the lesson. It is a remediation page that will be shown to a student who achieves a score of less that 75%. This file must be named REMED.DO and should contain a review of the most important information relevant to the questions contained in the lesson. There must be no ques-

tions or answers in this file.

When creating a lesson, the author must keep tabs on the number of files in the finished lesson. In the TRS-80 Model 100, for example, there is room to store at least 19 files in memory in addition to the five built-in ROM programs. Thus, there is room for the CAP program, 17 Q files, and REMED.DO. A lesson of this length could contain 17 questions together with 17 or fewer text pages. By omitting text in one or more Q files, the author can line up any desired number of questions after a page of text. The total word length of the lesson depends on the memory capacity of the computer. A 32K TRS-80 Model 100 could support a CAP lesson approximately equal in length to a type-written, double-spaced 15-page text. On the average, therefore, a single Q file could contain approximately one such page. If there are fewer than 17 Q files containing text, each could contain proportionately more text.

When a student uses a lesson created by CAP, he or she need only remember to press ENTER to turn text pages and to enter typed-in answers to given questions. In this way the CAP program will move through the lesson in the order in which it is to be presented, keeping a record of the right and wrong responses to questions. A response is judged correct if one of the three correct answers is embedded within it. If, at the end of a lesson, the student's score is lower than 75%, the score will be shown to the student, and the remediation text will appear, after which only those Q files whose questions had been incorrectly answered will be reviewed, giving the student a second opportunity to answer the missed question. If the score is 75% or higher, the lesson will end before proceeding to the remediation text. (To change the cut-off percentage to some other number it is necessary only to alter the decimal fraction .75 in lines 340 and 600.) All necessary comments to the student are provided by the program.

CAP ROUTINES

10-620 main program
1000-1080 reads the text files
3000-3150 evaluates student's response

5000-5020 causes program to pause until a key is pressed
6000 prints message; ends program

CAP VARIABLES

M—number correct
S—number of questions (Q file)
S1—current question (Q file)
R—flag indicating remediation routine
M$—wrong answer marker (—X)
F$—current file's name
T—number of questions missed on first pass
A$—dummy variable for reading one character from file
D$—student's response A1$ correct answer
A2$—correct answer
A3$—correct answer

THE FLASH CARD PROGRAM (FLASH)

The second program, FLASH, listed in Figure 2, is a flash card manipulator designed to aid in the memorization of lists of vocabulary. Construction of a traditional set of flash cards is usually a tedious process, requiring the preparation of numerous small cards, which are then difficult to organize or store. This program allows the student to create a text file named WORDS.DO in which each line contains a pair of words: the foreign word followed by a comma followed by the English word (no spaces). Unlike most other flash card programs, FLASH repeats only those words that have been translated incorrectly. The same word will not appear twice in a row unless it has been incorrectly translated more than once.

The words are presented in a random order, the student first having decided whether to translate from English to the target language or vice versa. When a word is translated correctly, this is so indicated, and the program unmarks the word so that it will not be repeated. When a word is missed, the correct translation is given. When only three or four words remain to be correctly identified, a delay of five to ten seconds between words may result as the program slows down somewhat, searching for the remaining words. When all words have been correctly translated, the program prints a message indicating the total number of words in the set (maximum: 50) and

the number of attempts required by the student to identify them all. If these two numbers differ significantly, further practice is probably required. Each set of words may be stored on tape for future reference.

Program Description: FLASH

FLASH ROUTINES
100-150 main program
1000-1050 reads text file
2000-2080 asks which side of flashcard to show
3000-3030 marks all words so that they will be shown
4000-4030 prints word; receives student's input; evaluates response (4020–K$—unmarks words correctly identified)
5000-5030 selects next unidentified word; ends program after last word
6000-6020 selects a random number R, (142-148 seed the random number generator.
These lines may be omitted or replaced by RANDOMIZE.)

FLASH VARIABLES
R—random number R1 the previous random number TR number of attempts
AA$—foreign (target) word
BB$—English (native) word
K$—marker identifying words to be reviewed (– X)
MX—number of words in file
AN$—student's response
J—dummy loop variable

Advantages of Notebook Computers
Editing or modifying existing files on computers such as the TRS-80 Model 100 or the NEC 8201 is a simple procedure, utilizing the powerful editing capabilities of the built-in text editor. For computers lacking built-in editors, the programs as listed here can be easily modified to be used on any computer for which a satisfactory editing mode is available. For example, word processing programs on larger computers could be used to create the necessary files, which could then be stored on disk and read successively into the computer's memory by a specially designed subroutine. But the integrated

memory of the two notebook computers mentioned in this article eliminates the need for expensive, noisy, and slow memory storage devices such as disk drives. Within integrated memory the program operates much more smoothly and rapidly than in conventional memory, for which peripheral storage devices such as disk drives are needed.

A further advantage of the two notebook computers referred to here is that elaborate graphic characters can be created within text mode. The built-in character generator of the TRS-80 Model 100, for example, can display up to 255 characters including symbols from mathematics and the sciences, as well as umlauts, accents, and other characters needed in the transcription of several European langauges. In addition, lines, boxes, blocks, and other figures may be constructed out of characters included in the graphics set and entered into text files as easily as text itself.

Once the necessary files and programs are residing in the non-volatile RAM memories of these notebook computers, the information will be retained, even when the computer's power is switched off. When it becomes necessary to remove them from memory in order for other lessons to be loaded or for the computer to be used for other purposes, the files can easily be saved on cassette tape.

Editor's Note: See page 23 for the listings for Mr. Roseberry's programs. ▣

CALI/CALL RESOURCES

Electronic University
At a press conference on September 12, 1983, in Washington DC several businessmen from Silicon Valley announced the beginning of the world's first Electronic University through TeleLearning Systems, Inc. Those in attendance included U.S. Secretary of Education, Terrell H. Bell; White House representative, James K. Coyne; Assistant Secretary of the Office of Educational Research & Improvement, Dr. Donald J. Senese; and other key

leaders in education and computer and telecommunications fields.

TeleLearning Systems, Inc. is a telecommunications system that connects personal computers of students with those of their instructors. Ultimately, any personal computer will be adequate. No experience on computers is needed and with the touch of a button logon is automatic. Secretary Bell stated that the thrilling thing about this concept is its flexibility and adaptability and capacity to reach all learners at all levels, and to teach them from where they are.

No longer will time and location be barriers to education as the classroom and teacher come to the student. All will be able to benefit from this one-on-one student-instructor connection, including the handicapped, homebound, and those confined to a penal institution. 170 courses are offered at present with hundreds more in the development process in connection with primary schools, universities, colleges, technical schools, trade associations, and businesses.

Using Electronic University a student can progress at his own rate, work at any time, and even set up a conference with the teacher through the computer. Using an electronic mailbox permits either party access to the stored assignments, questions, and instructions at any time.

The first to offer courses through TeleLearning Systems, Inc. in October, 1983, was the American Management Associations. TeleLearning courses are available through department stores as well as through licensed schools and corporations offering these programs. For more information, contact: Walter Rowan, The Orsborn Group, 505 Beach St., San Francisco, CA 94133 (415928-3600). ▣

```
COURSE AUTHORING PROGRAM
by
Robert L. Roseberry

10 CLEAR 500
15 M=0:K$="CORRECT!":W$="WRONG":S1=1:R=0

25 DIM M$(18)
50 S$=STR$(S1)
120 F$="Q"+S$+".DO"
130 GOSUB 1000
140 GOSUB 3000
300 S1=S1+1
305 S$=STR$(S1)
330 IF S1<=8 THEN GOTO 50
340 IF M/S=.75 THEN GOTO 6000
345 CLS
350 PRINT "HERE IS YOUR SCORE:"
360 PRINT
370 PRINT "YOU HAVE";M;" QUESTIONS RIGHT.
OUT OF A"
380 PRINT"TOTAL OF";S;" QUESTIONS."
390 PRINT:PRINT:PRINT"NOW FOR A BRIEF
REVIEW."
395 GOSUB 5000
400 F$="REMED.DO":R=1
410 GOSUB 1000
415 GOSUB 5000:CLS
420 M=0:T=0
430 S1=1
435 IF M$(S1)="X" THEN T=T+1:F$="Q"+STR$
(S1)+".DO":GOTO 462
440 S1=S1+1
450 IF S1>S THEN 600
460 GOTO 435
462 GOSUB 1000
465 GOSUB 3000
590 GOTO 440
600 IF M/T=.75 THEN CLS:GOTO 6000
610 PRINT@120,"MORE PRACTICE IS NEEDED."

620 END
1000 CLS
1010 OPEN F$ FOR INPUT AS 1
1015 IF LEFT$(F$,3)="Q.1" THEN INPUT#1,S

1017 IF LEFT$(F$,1)="Q" THEN
INPUT#1,A1$,A2$;A3$
1020 IF EOF(1) THEN 1070
1030 A$=INPUT$(1,1)
1040 IF POS(0)>30 THEN IF A$=CHR$(32)
THEN A$=CHR$(13)+CHR$(10)
1050 PRINT A$;
1055 IF CSRLIN=7 THEN GOSUB 5000
1060 GOTO 1020
1070 CLOSE
1080 RETURN
3000 LINE INPUT D$
3010 L1=LEN(A1$)
3020 L2=LEN(A2$)
3030 L3=LEN(A3$)
3040 LL=LEN(D$)
3050 FOR Z=1 TO LL
3060 IF MID$(D$,Z,L1)=A1$ THEN PRINT
K$:M=M+1:GOSUB 5000:IF R=1 THEN 440 ELSE
300
3070 NEXT Z
3080 FOR Z=1 TO LL
3090 IF MID$(D$,Z,L2)=A2$ THEN PRINT
K$:M=M+1:GOSUB 5000:IF R=1 THEN 440 ELSE
300
3100 NEXT Z
3110 FOR Z=1 TO LL
3120 IF MID$(D$,Z,L3)=A3$ THEN PRINT
K$:M=M+1: GOSUB 5000:IF R=1 THEN 440
```

```
ELSE 300
3130 NEXT Z
3140 PRINT W$:PRINT "SUGGESTED ANSW";
:PRINT A1$:S$=M$(S1)="X":GOSUB 5000
3150 RETURN
5000 R$=INKEY$:IF R$="" THEN 5000
5010 CLS
5020 RETURN
6000 CLS:PRINT@120,"WELL DONE. THIS
LESSON IS OVER.":END
```

```
FLASHCARD PROGRAM
by
Robert L. Roseberry

100 CLEAR 2000
105 R=0:TR=0
110 DIM AA$(50):DIM BB$(50):DIM K$(50)
120 GOSUB 1000
130 GOSUB 2000
140 GOSUB 3000
142 SEC=VAL(RIGHT$(TIME$,2))
144 FOR I=1 TO SEC
146 DUMMY=RND(1)
148 NEXT I
150 GOSUB 6000
160 IF K$(R)="X" THEN GOTO 4000 ELSE
GOTO 5000
1000 OPEN "RAM:WORDS.DO" FOR INPUT AS 1
1010 MX=0
1020 IF EOF(1) THEN RETURN
1025 MX=MX+1
1030 INPUT#1,AA$(MX),BB$(MX)
1050 GOTO 1020
2000 CLS
2010 PRINT@50,"GO FROM ---"
2020 PRINT@132,"1 - TARGET TO NATIVE"
2030 PRINT@172,"2 - NATIVE TO TARGET"
2040 PRINT@252,"(1 OR 2)";
2050 INPUT Q$
2060 Q=VAL(Q$)
2070 IF Q<1 OR Q>2 THEN 2000
2080 RETURN
3000 FOR J=1 TO MX
3010 K$(J)="X"
3020 NEXT J
3030 RETURN
4000 IF Q=1 THEN PRINT AA$(R) ELSE PRINT
BB$(R)
4003 TR=TR+1
4005 PRINT
4010 INPUT AN$
4014 LA=LEN(AA$(R))
4015 LB=LEN(BB$(R))
4016 L=LEN(AN$)
4017 J=1
4020 IF AA$(R)=MID$(AN$,J,LA) OR BB$(R)
=MID$(AN$,J,LB) THEN PRINT
"RIGHT":PRINT:K$(R)="":GOTO 150
4025 J=J+1:IF J<=L THEN GOTO 4020
4030 PRINT AA$(R);" MEANS ";BB$(R)
:PRINT:GOTO 150
5000 J=1
5010 IF K$(J)="X" THEN 150
5020 J=J+1:IF J<=MX THEN GOTO 5010
5025 PRINT"YOU REQUIRED";TR;"TRIES."
5027 PRINT"TO IDENTIFY";MX;"WORDS."
5030 END
6000 R1=R:R=INT(MX*RND(1)+1)
6010 IF R=R1 THEN 6020
6020 RETURN
```

Eliza

Description: ELIZA is a program that accepts natural English as input and carries on a reasonably coherent conversation based on the psychoanalytic techniques of Carl Rogers. You will have to forgive ELIZA for being a poor English student. You'll find that it is best not to use punctuation in your input, and you'll have to carry the conversation. But it does work!

How it works: In order to speak to you, ELIZA must: (1) get a string from the user, and prepare it for further processing; (2) find the keywords in the input string; (3) if a keyword is found, take the part of the string following the keyword and "translate" all the personal pronouns and verbs ("I" becomes "YOU", "ARE" becomes "AM", etc.); (4) finally, look up an appropriate reply based on the keyword which was found, print it and, if necessary, the "translated" string. ELIZA uses four types of program data to accomplish this:

(1) 36 keyword, such as "I AM", "WHY DONT YOU", and "COMPUTER". The keywords must be in order of priority, so ELIZA will key on "YOU ARE" before "YOU".

(2) 12 strings used for the translation or conjugation process. These are in pairs such that if one member of the pair is found, the other is substituted for it. Examples: "Y", "YOU", "AM", "ARE", etc.

(3) 112 reply strings. The strings are arranged in groups corresponding to the keywords. There is no fixed number of different replies for each keyword. Replies ending in a "*" are to be followed by the translated string, while the strings ending in normal punctuation are to be printed alone.

(4) Numerical data to determine which replies to print for each keyword. For each keyword there is a pair of numbers signifying (start of reply strings, number of reply strings). Thus the fifth pair of number, (10,4), means that the replies for the fifth keyword ("I DONT") start with the tenth reply string, and that there are four replies.

Detailed Explanation:
Lines 10-160: Initialization. Arrays and strings are dimensioned. N1, N2, and N3, which represent the number of keywords, number of translation strings, and number of replies respectively, are defined. Then the arrays are filled. S(keyword number) is the ordinal number of the start of the reply strings for a given keyword, R(keyword number)

is the actual reply to be used next, and N(keyword number) is the last reply for that keyword. Finally an introduction is printed. = Hi, I'm Eliza etc

Lines 170-255: User input section. This part of the program gets a string from the user, places a space at the start of the string and two at the end (to make it easier to correctly locate keywords and to prevent subscripting out of bounds), throws out all the apostrophes (so DONT and DON'T are equivalent), and stops if the word SHUT is found in the input string (which it takes to mean SHUT UP). ELIZA also checks for repetitive input by the user.

Lines 260-370: Keyword-finding section. ELIZA scans the input string for keywords and saves the keyword of highest priority temporarily in S, T, and F$. If no keyword is found, the keyword defaults to number 36, NOKEYFOUND (which causes ELIZA to say something noncommittal) and it skips the next section.

Lines 380-555: Translation or Conjugation section. The part of the input string following the keyword is saved. Then pairs of translation strings, as described above, are read and upon the occurence of one of these strings, the other is substituted for it. When this is done ELIZA makes sure there is only one leading space in the translated string.

Lines 560-640: Reply printing section. Using R(keyword number), S(keyword number), and N(keyword number), the correct reply is located. The pointer for the next reply is bumped and reset if it is too large. If the reply string ends in a "*" it is printed with the translated string, otherwise it is printed alone. The previously entered input string is saved to permit checking for repetetive input, and then ELIZA goes back for more input.

Limitations: Runs in 16K of memory.

Modifications: You can easily add, change, or delete any of the keywords, translation words, or replies. Remember, you will also have to change N1, N2, N3, and/or the numerical data. Just as a suggestion, if you decide to insert "ME" and "YOU" in the translation string list, put a nonprinting (control) character in YOU to prevent ELIZA from substituting I→YOU→ME. This means that YOU will always be assumed to be the subject of a verb, never the object, but resolving that difficulty is a whole different problem.

A Few Comments: The structures found in lines 120, 420, and 590 could be replaced by RESTORE NNNN statements if your BASIC has them. The use of an INSTR, SEARCH, or POS function to determine if one string is a substring of another would probably speed things up considerably (it takes ELIZA around 10 seconds to think of a reply).

What it all means: we'll leave to you. Although this program is an inferior imitation of the original, it does work. It is pretty farfetched to believe that a psycholanalyst is nothing but a sentence-input-keyword-finder-conjugator-reply-finder, but if you really think so, you can buy your computer a speech-recognition unit, a Computalker and a green couch, and charge $75/hr! My computer, the doctor!

Geneology: ELIZA was originally programmed in LISP by Joseph Weizenbaum at MIT. It was converted to BASIC by Jeff Shrager and further modified to work in Microsoft BASIC by Steve North, who also wrote these notes.

Name	Usage
R(X),S(X),N(X)	See Text = Keyword numbers
I$	Input string
K$	Keyword string
C$	Translated or conjugated string
F$	Reply string, also used to save K$ in scanning for keyword
R$,S$	Strings used in conjugation process
P$	Previous input string
Z$	Scratch (used for simulating RESTORE NNNN statement).
N1	Number of keywords
N2	Number of conjugation strings
N3	Number of replies
K	Keyword number
S,T	Used to save K and L when scanning for keyword
X,L	X,L Scratch. X is generally used for looping while L is used for scanning through strings
V	V Used for scanning for keyword string

56

ELIZA
CREATIVE COMPUTING
MORRISTOWN, NEW JERSEY

```
HI!  I'M ELIZA.  WHAT'S YOUR PROBLEM?
? YOU
WE WERE DISCUSSING YOU-- NOT ME.
? BUT YOU ATR THE PROBLEM
OH, I ATR THE PROBLEM
? SO I MISSPELLED A WORD
SAY, DO YOU HAVE ANY PSYCHOLOGICAL PROBLEMS?
? NO DO YOU
YOU'RE NOT REALLY TALKING ABOUT ME, ARE YOU?
? ONLY BEHIND YOUR BACK
WHY ARE YOU CONCERNED ABOUT MY BACK
? BECAUSE I'M AFRAID TO CON FRONT YOU
DID YOU COME TO ME BECAUSE YOU ARE AFRAID TO CON FRONT I
? YOU SPEAK WIERD ENGLISH
WE WERE DISCUSSING YOU-- NOT ME.
? AM I GETTING THE BEST OF YOU
OH, I
? WHAT IS THAT SUPPOSED TO MEAN
WHY DO YOU ASK?
? WHY NOT
WHY DO YOU ASK?
? I ASKED FIRST
WHAT DOES THAT SUGGEST TO YOU?
? THAT YOU ARE NOT AS SMART AS YOU THINK
WHAT MAKES YOU THINK I AM NOT AS SMART AS I THINK
? YOU ARE MAKING A FOOL OUT OF YOURSELF
DOES IT PLEASE YOU TO BELIEVE I AM MAKING A FOOL OUT OF YOURSELF
? DON'T TWIST THE WORDS ON ME BUDDY
I SEE.
? DO YOU REALLY
YOU'RE NOT REALLY TALKING ABOUT ME, ARE YOU?
? YES
ARE YOU SURE?
? YES I AM SURE
DID YOU COME TO ME BECAUSE YOU ARE SURE
? I CAME BECAUSE I AM
HOW LONG HAVE YOU BEEN
? AS LONG AS POSSIBLE
I'M NOT SURE I UNDERSTAND YOU FULLY.
? SO I AM NOT THE GREATEST CONVERSATIONALIST
DO YOU BELIEVE IT IS NORMAL TO BE NOT THE GREATEST CONVERSATIONALIST
? YES
I SEE.
? HOW CAN YOU SEE WITHOUT ANY EYES
DON'T YOU BELIEVE THAT I CAN SEE WITHOUT ANY EYES
? NO SO PROVE IT
ARE YOU SAYING NO JUST TO BE NEGATIVE?
? ABSOLUTELY NOT
YOU ARE BEING A BIT NEGATIVE.
? SO YOU WANT TO FIGHT DO YOU
WE WERE DISCUSSING YOU-- NOT ME.
? DON'T BACK DOWN FRIEND
WHY DO YOU BRING UP THE TOPIC OF FRIENDS?
? I DON'T HAVE ANY FRIENDS
DON'T YOU REALLY HAVE ANY FRIENDS
? NO NOT REALLY
WHY NOT?
? BECAUSE I AM YOU AND YOU ARE ME AND WE ARE ALL TOGETHER
PERHAPS YOU WOULD LIKE TO BE YOU AND WE AM ALL TOGETHER
? SAY WHAT
DOES THAT QUESTION INTEREST YOU?
? DEFINITELY
COME COME ELUCIDATE YOUR THOUGHTS.
? DO YOU LIKE THE BEATLES
OH, I LIKE THE BEATLES
? HOW ABOUT THE BEACH BOYS
WHY DO YOU ASK?
? DO YOU DREAM OFTEN
YOU'RE NOT REALLY TALKING ABOUT ME, ARE YOU?
? DREAM ON
WHAT DOES THAT DREAM SUGGEST TO YOU?
? WHAT DREAM
WHAT ANSWER WOULD PLEASE YOU THE MOST?
? DO YOU SUBSCRIBE TO CREATIVE COMPUTING
WE WERE DISCUSSING YOU-- NOT ME.
? I AM AN IBM 370/168
DO YOU ENJOY BEING AN IBM 370/168
? I AM WORRIED ABOUT THE COMPETITION
DID YOU COME TO ME BECAUSE YOU ARE WORRIED ABOUT THE COMPETITION
? DON'T GET PERSONAL
CAN YOU ELABORATE ON THAT?
? I'M GETTING TIRED OF LISTENING TO YOU TO SHUT UP
SHUT UP...
Ok
```
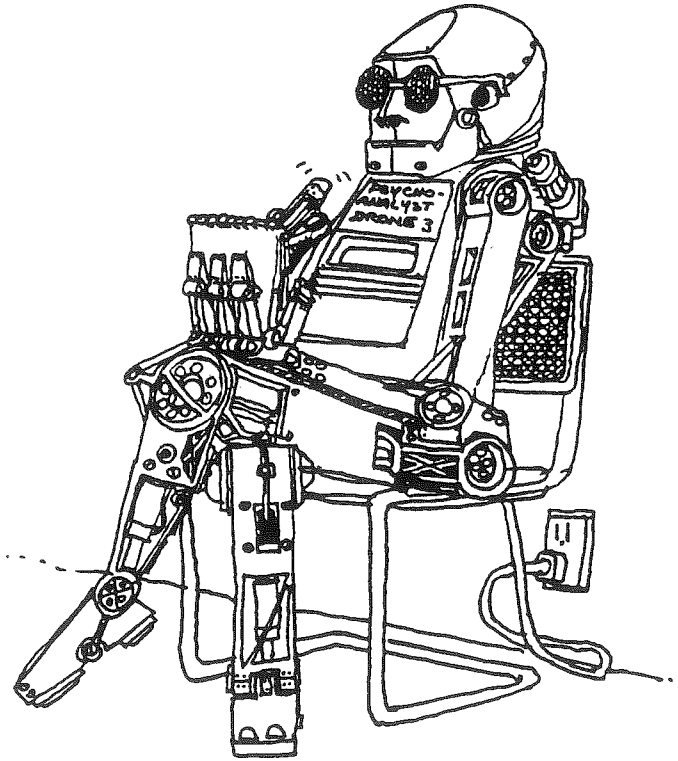
```
1 PRINT TAB(26);"ELIZA"
2 PRINT TAB(20);"CREATIVE COMPUTING"
3 PRINT TAB(18);"MORRISTOWN, NEW JERSEY"
4 PRINT:PRINT:PRINT
80 REM      -----INITIALIZATION-----
100 DIM S(36),R(36),N(36)
110 N1=36:N2=14:N3=112
120 FOR X=1 TO N1+N2+N3:READ Z$:NEXT X:REM SAME AS RESTORE
130 FORX=1 TO N1
140 READ S(X),L:R(X)=S(X):N(X)=S(X)+L-1
150 NEXT X
160 PRINT "HI!  I'M ELIZA.  WHAT'S YOUR PROBLEM?"
170 REM
180 REM      -----USER INPUT SECTION-----
190 REM
200 INPUT I$
201 I$=" "+I$+"  "
210 REM  GET RID OF APOSTROPHES
220 FOR L=1 TO LEN(I$)
230 IFMID$(I$,L,1)="'"THENI$=LEFT$(I$,L-1)+RIGHT$(I$,LEN(I$)-L):GOTO230
240 IFL+4<=LEN(I$)THENIFMID$(I$,L,4)="SHUT"THENPRINT"SHUT UP...":END
250 NEXT L
255 IF I$=P$ THEN PRINT "PLEASE DON'T REPEAT YOURSELF!":GOTO 170
260 REM
270 REM      -----FIND KEYWORD IN I$-----
```

57

```
280 REM
290 RESTORE
295 S=0
300 FOR K=1 TO N1
310 READ K$
315 IF S>0 THEN360
320 FOR L=1 TO LEN(I$)-LEN(K$)+1
340 IF MID$(I$,L,LEN(K$))=K$THENS=K:T=L:F$=K$
350 NEXT L
360 NEXT K
365 IF S>0 THEN K=S:L=T:GOTO390
370 K=36:GOTO570:REM  WE DIDN'T FIND ANY KEYWORDS
380 REM
390 REM    TAKE RIGHT PART OF STRING AND CONJUGATE IT
400 REM    USING THE LIST OF STRINGS TO BE SWAPPED
410 REM
420 RESTORE:FORX=1 TO N1:READ Z$:NEXT X:REM SKIP OVER KEYWORDS
430 C$=" "+RIGHT$(I$,LEN(I$)-LEN(F$)-L+1)+" "
440 FOR X=1 TO N2/2
450 READ S$,R$
460 FOR L= 1 TO LEN(C$)
470 IF L+LEN(S$)>LEN(C$) THEN 510
480 IF MID$(C$,L,LEN(S$))<>S$ THEN 510
490 C$=LEFT$(C$,L-1)+R$+RIGHT$(C$,LEN(C$)-L-LEN(S$)+1)
495 L=L+LEN(R$)
500 GOTO 540
510 IF L+LEN(R$)>LEN(C$)THEN540
520 IF MID$(C$,L,LEN(R$))<>R$ THEN 540
530 C$=LEFT$(C$,L-1)+S$+RIGHT$(C$,LEN(C$)-L-LEN(R$)+1)
535 L=L+LEN(S$)
540 NEXT L
550 NEXT X
555 IF MID$(C$,2,1)=" "THENC$=RIGHT$(C$,LEN(C$)-1):REM ONLY 1 SPACE
556 FOR L=1 TO LEN(C$)
557 IF MID$(C$,L,1)="!" THEN C$=LEFT$(C$,L-1)+RIGHT$(C$,LEN(C$)-L):GOTO557
558 NEXTL
560 REM
570 REM    NOW USING THE KEYWORD NUMBER (K) GET REPLY
580 REM
590 RESTORE:FOR X= 1 TO N1+N2:READ Z$:NEXT X
600 FORX=1TOR(K):READ F$:NEXT X:REM  READ RIGHT REPLY
610 R(K)=R(K)+1: IFR(K)>N(K) THEN R(K)=S(K)
620 IF RIGHT$(F$,1)<>"*" THEN PRINT F$:P$=I$:GOTO 170
630 PRINT LEFT$(F$,LEN(F$)-1);C$
640 P$=I$:GOTO 170
1000 REM
1010 REM        -----PROGRAM DATA FOLLOWS-----
1020 REM
1030 REM     KEYWORDS
1040 REM
1050 DATA "CAN YOU","CAN I","YOU ARE","YOURE","I DONT","I FEEL"
1060 DATA "WHY DONT YOU","WHY CANT I","ARE YOU","I CANT","I AM","IM "
1070 DATA "YOU ","I WANT","WHAT","HOW","WHO","WHERE","WHEN","WHY"
1080 DATA "NAME","CAUSE","SORRY","DREAM","HELLO","HI ","MAYBE"
1090 DATA " NO","YOUR","ALWAYS","THINK","ALIKE","YES","FRIEND"
1100 DATA "COMPUTER","NOKEYFOUND"
1200 REM
1210 REM     STRING DATA FOR CONJUGATIONS
1220 REM
1230 DATA " ARE "," AM ","WERE ","WAS "," YOU "," I ","YOUR ","MY "
1235 DATA " IVE "," YOUVE "," IM "," YOURE "
1240 DATA " ME "," !YOU "
1300 REM
1310 REM     REPLIES
1320 REM
1330 DATA "DON'T YOU BELIEVE THAT I CAN*"
1340 DATA "PERHAPS YOU WOULD LIKE TO BE ABLE TO*"
1350 DATA "YOU WANT ME TO BE ABLE TO*"
1360 DATA "PERHAPS YOU DON'T WANT TO*"
1365 DATA "DO YOU WANT TO BE ABLE TO*"
1370 DATA "WHAT MAKES YOU THINK I AM*"
1380 DATA "DOES IT PLEASE YOU TO BELIEVE I AM*"
1390 DATA "PERHAPS YOU WOULD LIKE TO BE*"
1400 DATA "DO YOU SOMETIMES WISH YOU WERE*"
1410 DATA "DON'T YOU REALLY*"
1420 DATA "WHY DON'T YOU*"
1430 DATA "DO YOU WISH TO BE ABLE TO*"
1440 DATA "DOES THAT TROUBLE YOU?"
1450 DATA "TELL ME MORE ABOUT SUCH FEELINGS."
1460 DATA "DO YOU OFTEN FEEL*"
1470 DATA "DO YOU ENJOY FEELING*"
1480 DATA "DO YOU REALLY BELIEVE I DON'T*"
1490 DATA "PERHAPS IN GOOD TIME I WILL*"
1500 DATA "DO YOU WANT ME TO*"
1510 DATA "DO YOU THINK YOU SHOULD BE ABLE TO*"
1520 DATA "WHY CAN'T YOU*"
1530 DATA "WHY ARE YOU INTERESTED IN WHETHER OR NOT I AM*"
1540 DATA "WOULD YOU PREFER IF I WERE NOT*"
1550 DATA "PERHAPS IN YOUR FANTASIES I AM*"
1560 DATA "HOW DO YOU KNOW YOU CAN'T*"
1570 DATA "HAVE YOU TRIED?"
1580 DATA "PERHAPS YOU CAN NOW*"
1590 DATA "DID YOU COME TO ME BECAUSE YOU ARE*"
1600 DATA "HOW LONG HAVE YOU BEEN*"
1610 DATA "DO YOU BELIEVE IT IS NORMAL TO BE*"
1620 DATA "DO YOU ENJOY BEING*"
1630 DATA "WE WERE DISCUSSING YOU-- NOT ME."
1640 DATA "OH, I*"
1650 DATA "YOU'RE NOT REALLY TALKING ABOUT ME, ARE YOU?"
1660 DATA "WHAT WOULD IT MEAN TO YOU IF YOU GOT*"
1670 DATA "WHY DO YOU WANT*"
1680 DATA "SUPPOSE YOU SOON GOT*"
1690 DATA "WHAT IF YOU NEVER GOT*"
1700 DATA "I SOMETIMES ALSO WANT*"
1710 DATA "WHY DO YOU ASK?"
1720 DATA "DOES THAT QUESTION INTEREST YOU?"
1730 DATA "WHAT ANSWER WOULD PLEASE YOU THE MOST?"
1740 DATA "WHAT DO YOU THINK?"
1750 DATA "ARE SUCH QUESTIONS ON YOUR MIND OFTEN?"
1760 DATA "WHAT IS IT THAT YOU REALLY WANT TO KNOW?"
1770 DATA "HAVE YOU ASKED ANYONE ELSE?"
1780 DATA "HAVE YOU ASKED SUCH QUESTIONS BEFORE?"
1790 DATA "WHAT ELSE COMES TO MIND WHEN YOU ASK THAT?"
1800 DATA "NAMES DON'T INTEREST ME."
1810 DATA "I DON'T CARE ABOUT NAMES-- PLEASE GO ON."
1820 DATA "IS THAT THE REAL REASON?"
1830 DATA "DON'T ANY OTHER REASONS COME TO MIND?"
1840 DATA "DOES THAT REASON EXPLAIN ANYTHING ELSE?"
1850 DATA "WHAT OTHER REASONS MIGHT THERE BE?"
1860 DATA "PLEASE DON'T APOLOGIZE!"
1870 DATA "APOLOGIES ARE NOT NECESSARY."
1880 DATA "WHAT FEELINGS DO YOU HAVE WHEN YOU APOLOGIZE."
1890 DATA "DON'T BE SO DEFENSIVE!"
1900 DATA "WHAT DOES THAT DREAM SUGGEST TO YOU?"
1910 DATA "DO YOU DREAM OFTEN?"
1920 DATA "WHAT PERSONS APPEAR IN YOUR DREAMS?"
1930 DATA "ARE YOU DISTURBED BY YOUR DREAMS?"
1940 DATA "HOW DO YOU DO ... PLEASE STATE YOUR PROBLEM."
1950 DATA "YOU DON'T SEEM QUITE CERTAIN."
1960 DATA "WHY THE UNCERTAIN TONE?"
1970 DATA "CAN'T YOU BE MORE POSITIVE?"
1980 DATA "YOU AREN'T SURE?"
1990 DATA "DON'T YOU KNOW?"
2000 DATA "ARE YOU SAYING NO JUST TO BE NEGATIVE?"
2010 DATA "YOU ARE BEING A BIT NEGATIVE."
2020 DATA "WHY NOT?"
2030 DATA "ARE YOU SURE?"
2040 DATA "WHY NO?"
2050 DATA "WHY ARE YOU CONCERNED ABOUT MY*"
2060 DATA "WHAT ABOUT YOUR OWN*"
2070 DATA "CAN YOU THINK OF A SPECIFIC EXAMPLE?"
2080 DATA "WHEN?"
2090 DATA "WHAT ARE YOU THINKING OF?"
2100 DATA "REALLY, ALWAYS?"
2110 DATA "DO YOU REALLY THINK SO?"
2120 DATA "BUT YOU ARE NOT SURE YOU*"
2130 DATA "DO YOU DOUBT YOU*"
2140 DATA "IN WHAT WAY?"
2150 DATA "WHAT RESEMBLANCE DO YOU SEE?"
2160 DATA "WHAT DOES THE SIMILARITY SUGGEST TO YOU?"
2170 DATA "WHAT OTHER CONNECTIONS DO YOU SEE?"
2180 DATA "COULD THERE REALLY BE SOME CONNECTION?"
2190 DATA "HOW?"
2200 DATA "YOU SEEM QUITE POSITIVE."
2210 DATA "ARE YOU SURE?"
2220 DATA "I SEE."
2230 DATA "I UNDERSTAND."
2240 DATA "WHY DO YOU BRING UP THE TOPIC OF FRIENDS?"
2250 DATA "DO YOUR FRIENDS WORRY YOU?"
2260 DATA "DO YOUR FRIENDS PICK ON YOU?"
2270 DATA "ARE YOU SURE YOU HAVE ANY FRIENDS?"
2280 DATA "DO YOU IMPOSE ON YOUR FRIENDS?"
2290 DATA "PERHAPS YOUR LOVE FOR FRIENDS WORRIES YOU."
2300 DATA "DO COMPUTERS WORRY YOU?"
2310 DATA "ARE YOU TALKING ABOUT ME IN PARTICULAR?"
2320 DATA "ARE YOU FRIGHTENED BY MACHINES?"
2330 DATA "WHY DO YOU MENTION COMPUTERS?"
2340 DATA "WHAT DO YOU THINK MACHINES HAVE TO DO WITH YOUR PROBLEM?"
2350 DATA "DON'T YOU THINK COMPUTERS CAN HELP PEOPLE?"
2360 DATA "WHAT IS IT ABOUT MACHINES THAT WORRIES YOU?"
2370 DATA "SAY, DO YOU HAVE ANY PSYCHOLOGICAL PROBLEMS?"
2380 DATA "WHAT DOES THAT SUGGEST TO YOU?"
2390 DATA "I SEE."
2400 DATA "I'M NOT SURE I UNDERSTAND YOU FULLY."
2410 DATA "COME COME ELUCIDATE YOUR THOUGHTS."
2420 DATA "CAN YOU ELABORATE ON THAT?"
2430 DATA "THAT IS QUITE INTERESTING."
2500 REM
2510 REM      DATA FOR FINDING RIGHT REPLIES
2520 REM
2530 DATA 1,3,4,2,6,4,6,4,10,4,14,3,17,3,20,2,22,3,25,3
2540 DATA 28,4,28,4,32,3,35,5,40,9,40,9,40,9,40,9,40,9,40,9
2550 DATA 49,2,51,4,55,4,59,4,63,1,63,1,64,5,69,5,74,2,76,4
2560 DATA 80,3,83,7,90,3,93,6,99,7,106,7
```