

## Computer-Controlled Drills for First-Year Russian

---

*Edward T. Purcell, University of Southern California*

In a recent article, George Kalbouss broadly described a variety of ways in which the computer can be used in the teaching of Russian.<sup>1</sup> The purpose of the present article is to elaborate on Kalbouss' contribution by describing in detail one use of the computer in Russian language instruction: the system which is in operation in Los Angeles at the University of Southern California. The system to be described is an automated, interactive drill routine for first-year Russian which allows students to do homework at a typewriter terminal under the supervision of a computer. It does not aim at replacing the teacher, only at lessening his burden of paper correcting while making possible more creative use of class time. By interactive is meant allowing an interplay between the student and the computer: the student and the computer take turns typing messages to each other. I will first present a sample session with a student using the computer-aided instruction (CAI) package for first-year Russian, and I will then offer some commentary.

The student sits down at the terminal, turns it on, picks up a telephone which links the terminal with the computer, dials the computer's telephone number, and places the receiver on an acoustic coupler which serves as the interface between the telephone and the terminal. Next he signs on. This is a procedure which tells the computer the student is entitled to use the Russian CAI package. It may thus be compared to opening a combination lock: one must know the numbers in the correct sequence or the lock will remain closed. The student types "3A3A," hits the return key, types ")" (a right parenthesis), and hits the return key again. The computer responds by printing out "SIGN-ON PLEASE: ████████████████████," which tells the student to type his account number in the sixteen spaces provided. (They are blanked out so that others will not learn the student's account number.) After the student types his account number and signals he is finished by hitting the return key, the machine responds with a standard message giving the date and time, indicating that the machine has recognized the student's account number, for example:

001) 12.03.59 06/28/74 PQHA02AB  
A P L \ 3 6 0

The lock has opened. The computer next prints out "*THE FOLLOWING LESSONS ARE AVAILABLE*" and a column of lesson names such as "*RUSSIANL6*," "*RUSSIANL7*," "*RUSSIANL8*," "*RUSSIANL9*," and so forth. Directions are then given for calling for a particular lesson:

*TO LOAD ANY LESSON TYPE:*  
                   )*LOAD 300008 LESSONNAME*  
*FOR EXAMPLE: )LOAD 300008 RUSSIANL8*

The student then requests a lesson, and the machine responds with a standard message, such as "*SAVED 12.15.54 12/17/73.*"

The sign-on and initial entry into the system have been conducted in English. At this point the student removes the English type element from the terminal and replaces it with a Russian element. Henceforth everything is in Russian until the very end of the session. To locate the Russian letters the student must now watch the keyboard (see below, figure 2). To start the Russian program, the student types *davaj* (in cyrillic letters), and the computer responds by asking for the student's first and last names:

Имя? \_\_\_\_\_  
 Фамилия? \_\_\_\_\_

Students will have already chosen Russian first names and learned the cyrillic spelling of their last names. Thus a student named Theodore Jones might respond by typing "Fedja" and "Džons" in cyrillic letters in the spaces provided. The machine next types out a greeting using the student's first name, a list of the exercises available in the lesson, and a request for the student to choose one:

Здравствуй Федя! Очень приятно тебя видеть!

В этом уроке следующие упражнения:

б /стр. 235/  
 в /стр. 235/  
 г /стр. 236/  
 з /стр. 237/  
 й /стр. 238/  
 х /квиз/

Выбери одно упражнение:

? \_\_\_\_\_

Note that the list of exercises contains page references to the textbook, which allows the student to consult it in choosing an exercise. Although the textbook orders exercises according to the latin alphabet, the student has a cyrillic-only type element, and we are forced to order the exercises accordingly: a, b, v, g, etc. Should the student forget the cyrillic order, the page references help him locate the exercise.

Fedja chooses exercise B, and the computer shifts to a short subroutine

which types out a title for the exercise followed by directions, a sample question and answer, and a request whether Fedja wants to start at the beginning:<sup>2</sup>

```

//////
//////
Урок 14, упражнение Б /стр. 235/
//////
//////
Дай правильную форму слова в скобках.
Следуй данному образцу:
    Образец:
    Они говорят об /этот гид/.
    ?___этом гиде
    Хочешь начать с начала? /да или нет/
    ?___

```

If the answer is *da*, the computer will start with the first question; if it is *net*, the student is told to type the number of the question with which he wants to start. This option is included in the program because sometimes there are long exercises which the student does not complete during one session. This option permits him to re-enter the exercise at the point where he stopped. The four rows of slashes in the exercise title serve to mark the beginning of the exercise on the student's print-out, so that it can be easily located on the paper at a later time. The sample shows the form in which the question will be presented and the form in which the answers are expected. Thus the student must not type in a full sentence, but only modifications of the words presented in slashes.

Assuming that the student opts to start the exercise at the beginning, the first few exchanges between student and computer might proceed thus:

1. Они говорят об /этот гид/.  
? \_\_\_ этом гиде  
Да.
2. Они говорят об /этот профессор/.  
? \_\_\_ этом профессоре  
Неплохо.
3. Они говорят об /этот мужчина/.  
? \_\_\_ этом мужчине  
Верно.
4. Они говорят об /этот учитель/.  
? \_\_\_ этом учителе  
Правильно.
5. Они говорят об /Исторический музей/.  
? \_\_\_ Историческом музее  
Хорошо.

Note that the first question in the exercise is always the sample question given earlier; if the student is paying attention to the sample, he knows the answer to the first question. After typing each question the computer drops to the next line, types a question mark, skips a space, and types two underscores. This is the standard way of indicating that the computer is waiting for an answer to be typed in.

In the above, the student has given a series of correct answers. Note that the computer's positive replies get slightly more positive with each successive correct answer. The computer's positive replies range from neutral to enthusiastic: 1. *Da*, 2. *Neploxo*, 3. *Verno*, 4. *Pravil'no*, 5. *Xorošo*, 6. *Prekrasno!* 7. *Čudesno!* 8. *Otlično!!* 9. *Molodec!!!* In effect the student is on a ladder; each successive correct answer moves him up one rung.

But sometimes the student makes a mistake:

6. Он часто думал об /эта девушка/.  
 ? \_\_\_\_ этом девушке  
     это! девушке  
 Нет, неправильно.
6. Он часто думал об /эта девушка/.  
 ? \_\_\_\_ этое девушке  
     это! девушке  
 Нет! Федя, не так!
6. Он часто думал об /эта девушка/.  
 ? \_\_\_\_ этой девушке  
     это! девушке  
 Товарищ Джонс, что с тобой?  
 Правильный ответ — этой девушке
6. Он часто думал об /эта девушка/.  
 ? \_\_\_\_ этой девушке  
 Это лучше.
7. Он часто думал об /эта немка/.  
 ? \_\_\_\_ этой немке  
 Неплохо.

After Fedja hits the return key to indicate that he has finished typing, the computer compares his answer letter by letter with the expected answer. In this instance there is a mismatch in the fourth letter. The computer skips four spaces and retypes the answer, inserting an exclamation mark for the wrong letter. Next it drops to the next line and types a negative comment, then drops one more line and retypes the question. As can be seen, negative comments are also graded: the computer gets testier with each successive mistake. After three mistakes on one question, the computer gives the student the correct answer, and types out the question once again. The above shows the student answering correctly on the fourth attempt. If he should make a fourth mistake (not shown), the computer types fifteen

question marks and proceeds to the next question. The machine's positive response to the first correct answer after several incorrect answers is *Èto lučše*. Thereafter it resumes the graded positive responses, except that the student is dropped one rung on the ladder for every mistake he makes. In the sample above, Fedja starts three rungs lower. If he had only made one mistake, i.e., gotten the correct answer on the second try, he would have been dropped only one rung, and the response to the correct answer for question 7 would have been *Pravil'no*. Gradations in the responses for correct and incorrect answers are included in order to simulate the reactions of human teachers, who are often pleased when students do well and displeased when they do not. For instances when the student simply does not know the answer and would rather avoid the computer's negative comments, the program allows him to respond to the computer's request for the answer with *požalujsta*, whereupon the computer supplies the answer (e.g., "Pravil'nyj otvet—ètoj ploščadi"), drops the student one rung on the positive-response ladder, and repeats the question.

In this way the computer guides the student through an exercise. After the completion of the last question, the computer types out

/////  
 Конец упражнения Б  
 ////

Should a student want to get out of an exercise before answering all the questions, he can, in response to any question, type *dostatočno*, which the computer recognizes as a command to skip to the end of the exercise. Immediately after the end-of-exercise statement shown above, the computer asks

Еще одно упражнение? /да или нет/  
 ? \_\_\_\_

If the answer is *da*, the computer again types out the list of available exercises and asks the student to choose one. If the student answers *net*, the computer bids him farewell (as shown below).

The list of exercises for every lesson includes a quiz consisting of ten questions drawn from those exercises. The teacher has the option to specify how many questions are to come from each exercise (e.g., two from B, one from C, three from D, etc.), but the questions themselves are randomly selected by the computer, a new random selection being made every time a student requests a quiz. Thus a student could request three successive quizzes and get three different selections of questions. In response to a request for a quiz, the computer might type out



Урок 14, проверочная работа

Образец: В чем лежало письмо? /словарь/  
? \_\_\_\_ В словаре.

1. На какой лекции? /интересная лекция в университете/  
? \_\_\_\_

As before, the student follows the format of the sample question and answer when typing in his answers. When he hits the return key, the computer skips a line and types the next question. Rather than commenting on the student's answer the computer analyzes it and stores its analysis pending completion of the exercise. When the student finishes the tenth question and hits the return key, the computer provides feedback on his performance:

10. Завтра мы идем в /этот хороший новый театр/.  
? \_\_\_\_ этом хорошем новом театре

////

Конец проверочной работы

////

Правильные ответы: 1 2 3 4 7 8 9

Неправильные ответы: 5 6 10

Твой счет = 70%

Правильные ответы:

5. вас

6. мне

10. В этот хороший новый театр

Еще одно упражнение?/да или нет/

? \_\_\_\_

The listing shows the numbers of the correct and incorrect answers, followed by a percentage score. Next the computer prints out a list of the correct answers for the questions incorrectly answered. Thus the student can look above on his print-out and compare the answers which the computer expected with the answers he typed in. Immediately after the listing of correct answers shown above, the computer presents the standard option to quit or continue. If the student answers *da*, the computer again types out the list of available exercises and the student makes a new choice. If he answers *net*, the computer bids him farewell:

До свидания Федя.

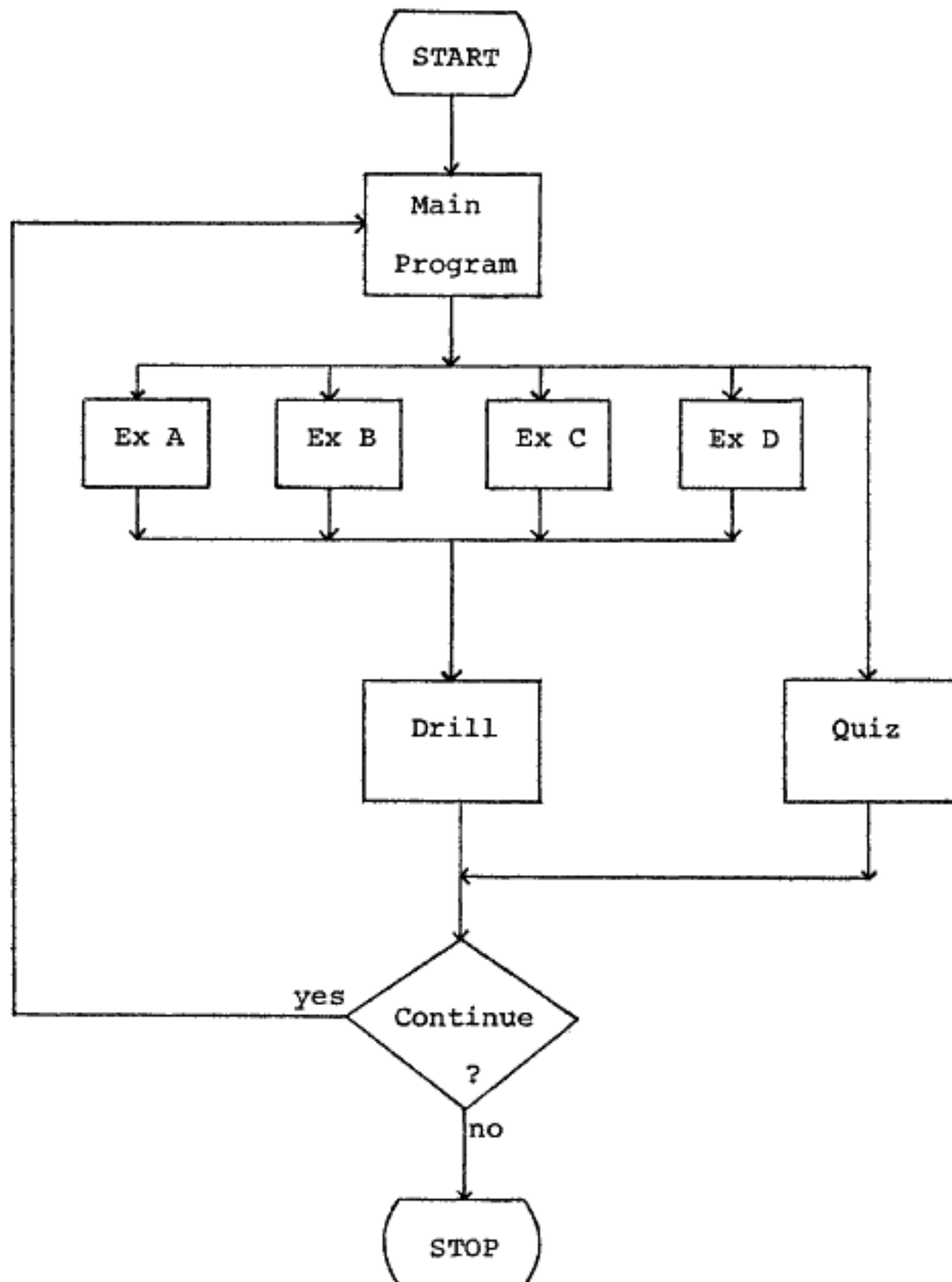
Было очень приятно!

The student has exited from the Russian segment of the package. It remains for him to replace the Russian type element with an English one and sign off in English: ")OFF." The machine responds with a statement of time at sign-off ("004 20.32.25 06/25/74 PQH"), a statement of the amount of time the student sat at the terminal ("CONNECTED 0.24.09 TO DATE

1.50.55”), and a statement of the amount of time spent by the computer in handling the student (“*CPU TIME 0.00.04 TO DATE 0.00.09*”). Finally the student turns off the terminal and hangs up the telephone mentioned at the outset.

Figure 1 presents a generalized flow chart of the sequence of programs involved in the system described above. The main program identifies the student, greets him, offers him a choice of exercises, and depending on the student’s choice activates the appropriate set of questions and answers. Next a small program is called which prints out the title of the exercise with

FIG. 1. Flow chart showing sequence of programs in a typical lesson of the CAI system for first-year Russian.



a sample question and answer. Next the main drill program prints out the questions sequentially and evaluates the answers typed in by the student. At the end of each exercise the student is given the option to exit from the system or continue with another exercise.

An important feature of interactive programs such as the one being described is that they can be handled through **REMOTE TERMINALS**: the student does not have to be physically present at the computer but can be across campus, across town, or across the country. There are other, noninteractive programs which are handled under what is termed **BATCH PROCESSING**: large batches of punched cards or other media are submitted to the computer to be processed. Batch processing is in my view less interesting for the language teacher and will not be further considered here. It is pertinent to note, however, that under what is termed **TIME-SHARING**, computers can handle both interactive programs and batch processing simultaneously. Much of the time involved in an interactive program is taken up by the typing in and typing out of messages; the student may sit working on drills at a remote terminal for several hours while the actual computer time expended is measured in seconds; e.g., Fedja's twenty-four minutes at the terminal in the above drill accounted for only four seconds of computer time. As a result the computer can handle several interactive operations at remote terminals at the same time that it is processing batch programs. This has implications for the cost of operating CAI programs.

A variety of terminals is available for interactive programming. A CRT or screen terminal consists of a keyboard linked to a televisionlike screen; whatever is typed by the student or the computer is displayed on the screen. A teletypewriter terminal types out questions and answers on paper; this the student can tear off and take home at the end of the session. In the choice between these two types of terminal at least one pedagogical issue is involved: some educators question the advantage of supplying the student with paper print-out ("hard copy") on the grounds that the student should be encouraged to master the concepts directly on the basis of the screen display ("soft copy"). A disadvantage of the teletypewriter is that it is slow and noisy;<sup>3</sup> CRT terminals are more expensive but very quiet in operation. Some teletypewriter terminals rent for approximately \$100 per month and can be purchased new for \$3000-\$4000. Rental includes service; purchase does not, but service contracts are available. Flexible CRT terminals cost over \$5000 new. By flexible I mean able to handle Russian programs and the programs of various other users (it is easier to obtain funding for equipment available to a variety of users; for example, when several modern-language programs submit a proposal for a CAI package, the large number of students potentially involved makes the proposal more attractive). In any case



one should seek the advice of specialists at the local computing center before choosing a particular model or make.

A major cost in developing a CAI system is programing. Someone must write a set of precise directions specifying the conduct of the exercises: the questions to be presented, the answers expected, explanatory remarks at the start of the drill, remarks to follow incorrect answers, etc. This program must specify everything that the machine must know in order to guide the student through the drill. It is sometimes possible to obtain previously developed programs from elsewhere, but this leaves one helpless in the event of malfunction unless one understands the programs themselves. Moreover, programs are written in special programing languages, of which there are many: FORTRAN, COBOL, BASIC, and others. The program described above is written in APL,<sup>4</sup> an efficient and flexible programing language which looks like this:

```

FN ← 17 ↓ FN ← □, 0ρ □ ← 'FIRST NAME???'
LN ← 16 ↓ LN ← □, 0ρ □ ← 'LAST NAME???'
'HI '; FN; '. NICE TO SEE YOU.'
```

Line 1 tells the computer to type out a request for the student's first name, wait for the student to type in his first name, and store whatever the student types under the variable *FN*. Line 2 tells the computer to type out a request for the student's response and store it under *LN*. Line 3 tells the computer to type out a greeting using the student's first name in the middle; cf. "Zdravstvuj Fedja! Očen' prijatno tebja videt'!" (above). Interactive programing languages are not difficult to learn, and consultants at a nearby computer center are usually glad to help one learn. But they normally are not willing to do the actual programing without compensation. Student help can often be obtained for the task of typing in questions and answers. I recommend that the individual instructor not undertake to do the programing without release time or extra compensation.<sup>5</sup>

Since the student and the computer interact by typing messages to each other at a terminal, a consideration of overriding importance in choosing a terminal is the requirement that it be able to handle the cyrillic alphabet. The IBM 2741 terminal currently in use at the University of Southern California is similar to an IBM Selectric typewriter and features changeable type elements.<sup>6</sup> Since a variety of changeable type elements is available, both bilingual English/Russian and cyrillic-only, it is important to note that IBM manufactures terminals with two different cyrillic keyboard arrangements, one which more or less corresponds with the English keyboard and one which approximates the keyboard of Soviet typewriters.<sup>7</sup> Obviously one should be sure to select type elements and terminals having the same arrangement of letters, for typing at a terminal with the Soviet arrangement

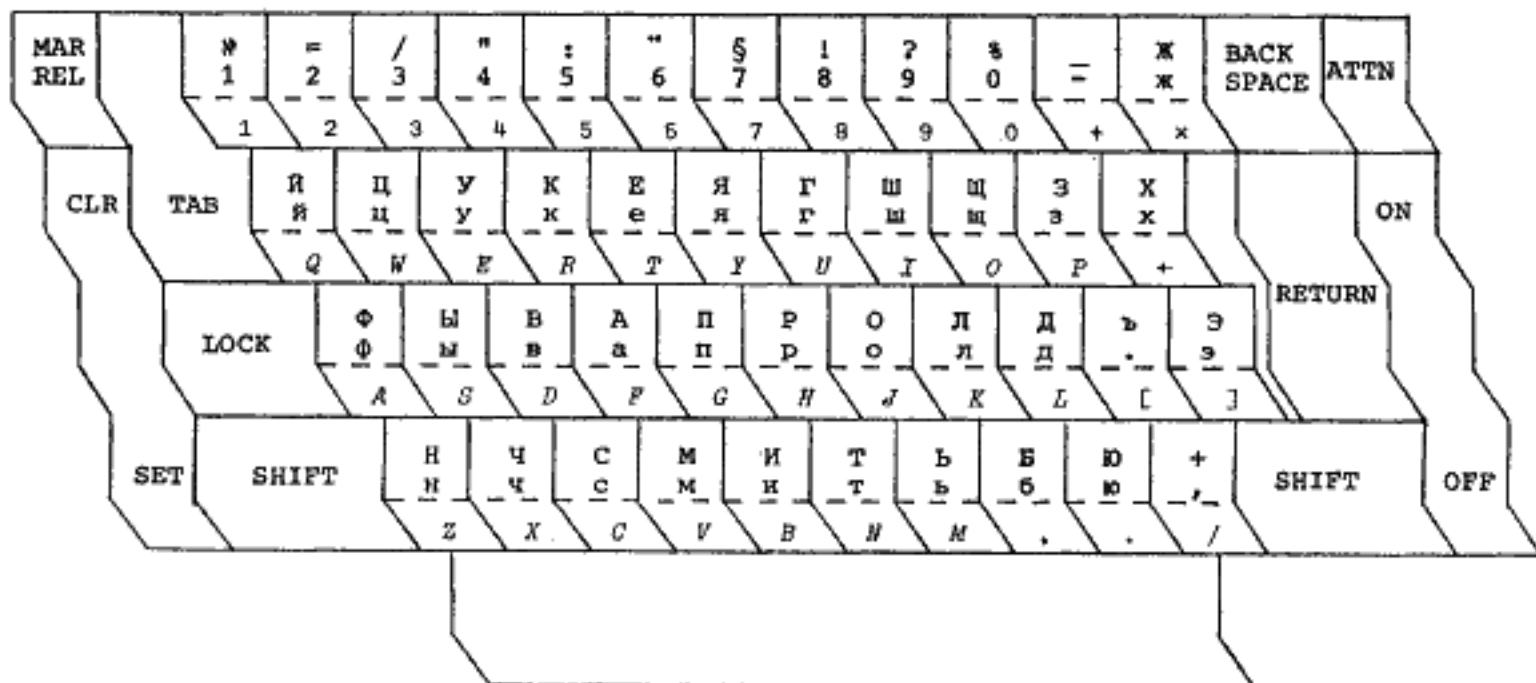
of letters using an element with the arrangement of letters corresponding to the English would result in a badly skewed print-out.

Which keyboard arrangement is best for CAI in first-year Russian, the one which more or less corresponds to the English keyboard, already supposedly known to the student, or the Soviet arrangement? Since a surprising number of our freshmen do not know the English keyboard, I favor the Soviet arrangement. I also favor cyrillic-only elements, since in my opinion it is better to conduct drills entirely in Russian. IBM presently manufactures five cyrillic-only type elements which can be used with the Soviet keyboard arrangement.<sup>8</sup> It also manufactures a bilingual element. Bilingual elements, such as described by Kalbouss, are useful in vocabulary or translation exercises: the computer can type out an item in one language and the student can answer in the other.<sup>9</sup>

Figure 2 shows the keyboard arrangement at the IBM 2741 installment in use at USC. The keyboard comes with the appropriate cyrillic symbols engraved on the key tops. On the front surface of each key we have attached a sticker showing which character will print when the APL/English type element is mounted. Thus the student looks at the key fronts when doing the sign-on and sign-off procedures during a typical drill session and looks at the key tops during the Russian portion of the session. The English stickers are also for the convenience of those using the terminal for purposes other than Russian.

The sample session described above assumes a certain familiarity on the part of the student with CAI. Simplified introductory materials have had to be developed to familiarize students with the system. A first session, entirely in English and with simple questions, gives students a taste of the

FIG. 2. Keyboard layout of IBM 2741 terminal for use with either Courier Russian element or APL/English element. Key tops show Russian, stickers attached to key fronts show APL/English.



system in the first week of classes. A videotaped segment is being developed to present the sign-on procedures. A program is also being developed to introduce students to the Russian keyboard; the first exercises will involve little more than copying letters and short words. Once students have a basic idea of how the computer operates and can locate the letters on the keyboard, they are ready to start rudimentary drills. To help the student with the computer's comments and directions, we give him a list of English equivalents for the computer's Russian messages.

The CAI Russian programs described above are clearly not the best that could be done from the point of view of programming. My decision was to go for a less elaborate model which could be rapidly developed for an entire first-year course, rather than to work on a more complex model which would not be so broad in application. In the near future I hope to add an accounting system which will keep track of the work done by each student on each exercise of a lesson. As a student works on a lesson the computer will record which exercises are done and which questions are answered incorrectly, storing these data under the student's name. The teacher will at any time be able to request a print-out.

A second improvement would be a more sophisticated answer-checking routine. At present there is only one correct answer, which the student's answer must match letter by letter. A more flexible routine would search the student's response for the string of letters representing the expected answer or would allow several possible answers. Further, an optional timer might be included which would offer the student a choice of time limits—perhaps thirty, forty-five, or sixty seconds per answer—after which incomplete answers would be considered mistakes. (Some allowance could be made for automatically increasing the time limit whenever a long answer is expected.) Another small improvement would be to tack on at the end of each exercise any questions missed by the student, so that he does not complete an exercise until all questions are answered correctly.<sup>10</sup>

In summary, I have described a computer-aided instructional system for first-year Russian. The system is designed to complement one particular first year textbook, but could be adapted to almost any other text which utilizes written drills involving short, set answers.<sup>11</sup> The system allows a student to do homework under constant automated supervision. Its main advantage over traditional modes of homework correction is that it provides instant feedback, with no delay between answering a question and finding out whether that answer is correct. Although the system requires that students be present at the terminal to do homework, a wise choice of location (e.g., a language laboratory) can encourage students to come to the same place for a variety of language-related materials. It is also possible to make previous lessons available to students so that the system can be a means of nonscheduled review on their part. This latter attribute is par-

ticularly valuable as final examinations loom. The system can also be useful to students who fall behind during the course of the academic year. In addition, the system might be used as one component of a self-paced, individualized instructional package for first-year Russian. However, it does not aim at replacing the teacher. It may be possible to design more elaborate programs which more closely mimic the teaching function, but I am not interested in an expensive, automated classroom (nor I might add are most administrators).

I have also discussed some of the more technical aspects of the system such as programs, terminals, type elements, costs, and resources. Such information may be helpful to those interested in developing similar systems for use at their own institutions.

## NOTES

- 1 "Computer-Assisted Instruction in the Teaching of Russian," *SEEJ*, 17 (1973), 315-21. The present paper was first presented at the annual meeting of AATSEEL in Chicago in December 1973.
- 2 Based on Ben T. Clark, *Russian for Americans* (New York: Harper & Row, 1967), 235.
- 3 The Van San Corporation (1180 Centre Dr., City of Industry, Calif. 91748) manufactures a Sound Off Dampener, a large metal-and-plexiglass hood for the IBM 2741.
- 4 APL stands for A Programming Language. *APL / 360: An Interactive Approach* (New York: Wiley and Sons, 1970) offers an introduction to this computer language.
- 5 Kalbouss suggests (p. 321) that national CAI networks are not "too far in the future." From my vantage point on this side of the Rockies, regional networks seem much more feasible. In any case I urge interested parties to learn an interactive programming language and become involved.
- 6 Other manufacturers produce comparable terminals which use Selectric type elements. Among these are the DATEL 20-31, TST 707, and the NOVAR 5-50. One should consult the local computing center for advice and specific information on available terminals.
- 7 It is not at all clear to me what is the Soviet standard keyboard layout. While most Russian keyboards available in this country are similar, they often differ in the location of some of the letters. The IBM cyrillic-only layout is shown in figure 2.
- 8 A Courier 72 pica (10 pitch), no. 2428609 (also listed as 6522907); a light italic elite, no. 2428609 (also listed as 6522904); a large pica 72, no. 2428607 (also listed as 6522901), a Polyglo pica, no. 2428606 (also listed as 6522900); and an Artisan elite, no. 2428610. One should be aware that type elements are fragile, made from plastic with a metallic coating, and cost around \$18-\$20 each. They are easily broken when mismounted or dropped on the floor.
- 9 The bilingual type element mentioned by Kalbouss is no. 1167045. It has upper-case cyrillic letters located over upper-case latin letters and hence is suitable for



use on Russian keyboards with the English layout. For the Soviet keyboard arrangement the bilingual type element to be used is no. 1167625. Camwil, Inc. (835 Keeaumoku St., Honolulu, Hawaii 96814) manufactures a bilingual type element with lower-case cyrillic letters located over lower-case latin letters, arranged according to the Soviet keyboard (part no. 113-m, 12 pitch).

- 10 Two further improvements may interest colleagues who would like to use CAI for more advanced applications. First, as described by W. D. Hagamen, D. J. Linden, H. S. Long, and J. C. Weber in "Encoding Verbal Information as Unique Numbers," *IBM Systems Journal*, 11, no. 4 (1972), 278-315, character strings may be encoded as numeric data at great savings of storage. Using available APL functions, it is possible to store letters, words, sentences, and even entire paragraphs as a unique number. Second, it is possible, though presently expensive, to present visual materials under computer control. At Golden West College in Costa Mesa, California, microfiche readers have been linked to IBM 2741 terminals in such a way that the computer can present individual slides and type out questions to the student. Such a system would be very useful for vocabulary drills; it would enable one to do all vocabulary drills strictly in Russian. The installation at Golden West College has the capability of branching to different slides depending on the student's answers. For example, a picture of a chair is shown on the microfiche reader and the computer asks the student to identify the item shown by choosing one of four possible answers: "table," "chair," "lamp," or "window." If the answer is "table," the computer will flip the microfiche card to show him a slide of a table then type out on the terminal, "No, this is a table." Then the computer flips back to the original slide of the chair and types out the original question followed by the same four choices. Such a system would improve the speed and efficiency in acquiring vocabulary.
- 11 When first developed, the system was based on Clark's *Russian for Americans*. Since our department has recently decided to change to Kostomarov's *Russian for Everybody*, new questions and answers will be fed in.