DECUS
PROGRAM LIBRARY

DECUS NO.	8-656
TITLE	SELFDRILL - THE SLOAN SELFDRILL PROGRAM
AUTHOR	Francis M. Wheeler
COMPANY	Beloit College Beloit, Wisconsin
DATE	September 1, 1973
SOURCE LANGUAGE	PAL III

ATTENTION

This is a USER program. Other than requiring that it conform to submittal and review standards, no quality control has been imposed upon this program by DECUS.

The DECUS Program Library is a clearing house only; it does not generate or test programs. No warranty, express or implied, is made by the contributor, Digital Equipment Computer Users Society or Digital Equipment Corporation as to the accuracy or functioning of the program or related material, and no responsibility is assumed by these parties in connection therewith.

SELFDRILL - THE SLOAN SELFDRILL PROGRAM

DECUS Program Library Write-up DECUS NO. 8-656

USER'S GUIDE

PURPOSE OF PROGRAM:

This program is designed to help you establish immediate recall of terms, dates, formulas, spellings of troublesome words, or other information which you would like to be able to master exactly. It is not intended as a means of evaluating your understanding of semantic relationships, nor even as a way of inculcating such understanding. However, it can help you to master any specific material which you are willing to learn in a fixed format.

So if you are learning Latin, for example, and wish to associate the English word austere with the Latin severus, you can use these two terms as cue and matching response in your drill, and quickly form a lasting connection between them. You will remain aware that severus and austere are not exact synonyms, and that sometimes other English words provide a better translation of the sense of this Latin word. But austere is a good place to begin. DRILL simply provides one technique by which you can begin to form the cluster of associations appropriate to a given term.

LOADING AND INITIALIZING:

To use the program, load it by the normal procedures into any available memory field of the PDP-8. It will not disturb the binary loader or RIM loader on the last page of the field. Start the program at location 200_8 (or 207_8 if you wish to avoid reinitializing). After identifying itself, the program will ask for your name. Be sure to use a carriage return after the name you wish the computer to use in addressing you.

The program will proceed to give you a report on the amount of information currently stored in its internal file, and the amount of storage (in characters) still available. If you know what items are stored, and wish to use them for drill, you may proceed to drill at once. If you do not know what the file contains, you may secure a printout of its contents by typing:

```
*PF
```

With the paper tape punch on, this command will give you both a tape of the file contents and a printout. To get the printout only, simply ignore the machine's request to turn on the punch. If you wish to start out with a clear file, use the command:

```
*I
```

This will reinitialize the file and produce the report:

```
Ø DRILL ITEMS
ROOM LEFT=1278 CHARACTERS
```

PREPARATION OF NEW DRILL ITEMS:

To enter new items, simply type the information you want to use as a cue, then a slash(/) followed immediately by whatever you want to use for the response to that cue. All printable characters are legal except the slash, asterisk, question mark, and an initial space at the beginning of the cue field (spaces elsewhere are perfectly acceptable). The program will accept the RUBOUT as a character delete signal. (This may be repeated to erase successive characters to the left. Each RUBOUT typed in is indicated by one reverse slash echoed by the program.) The program will also accept the CNTRL/U signal as a request to delete everything typed thus far on the line. These comments are identical in form and function to those implemented in the Digital library Text Editor. The carriage return signals completion of the drill item, which is then stored in the file.

Use care in the preparation of drill items, bearing in mind that the program will expect you to respond with exactly the sequence of characters found in the response field. Clearly you must not use the same cue for two or more different responses within the same drill. The more succinct and natural the material is in form, the easier it will be to master. One of the values of creating your own drill is that the very process of doing so will force you to think about the essential substance of what you are trying to master.

DISPLAY/DELETE FUNCTION:

If you wish to examine any of the records currently stored in your drill file, you may display the item in question on the teletype by using the command

```
*D n
```

where n is the number of any item in the drill file. Note that records are not labelled with any fixed number, but are designated by their relative position in the file. That is, record 1 is simply the first record in the file, record 2 the second, and so on. If record 1 is deleted from a file, for example, all the other records in the file will automatically be assigned a number one less than before the deletion. The display command enables you to check to make sure you are specifying the correct record before you delete it. You may then delete this record by typing the

```
*CNTRL/U
```

command. Alternatively, you may use this same command (CNTRL/U)

when any drill or review item is being presented. It will then delete the current record and adjust the file, the data tables, and the counters accordingly. If the item is a drill record, the program will produce a status report which reflects these adjustments.

USE OF TAPES:

The program will accept up to thirty-one (31) items for drill at any one time. You may punch out a tape of the file you have created when you have finished typing in the desired items. You may also use such a tape to create the drill file. Type the command:

*RD

Then follow the instructions printed out by the program. Note that records are separated by a short segment of blank tape so that their limits will be clearly distinguishable to the eye. Each record on the tape begins with an arrowhead to remind you of the direction of tape movement. It is possible to mount the tape so that it reads backwards, and the result is garbage. So please note the directional arrows, and be sure they point toward you as you mount the tape in the reader. IN MOUNTING THE PAPER TAPE FOR READ IN, BE SURE TO POSITION THE TAPE SO THAT THE BLANK SECTION OF THE TAPE IMMEDIATELY BEFORE THE FIRST ARROWHEAD (OR ANY DESIRED ARROWHEAD) IS OVER THE READER SENSE PINS. Each arrowhead marks the beginning of an item or "record". Reading a drill tape into the machine always reinitializes the file, so that any previous material in the file is automatically destroyed. Therefore, the reading in of any tape you wish to use for drill must precede your type in of any additional items you might want to add to the tape.

You may also read in your review tape at this point. Type the command:

*RV

Then follow instructions printed out by the program. Review items are used by the program to provide proper spacing between the successive presentations of items being learned. If you wish, you may begin with an empty drill file and simply read in a review tape. Items in the review which seem unfamiliar to you will automatically become part of your drill file. Thus, you may use the program simply as a means of comprehensive and thorough review of a file of materials you wish to master completely.

OPTIONS AVAILABLE DURING DRILL:

As explained on page 2, you may delete any item from the file while drilling by using the *CNTRL/U command. This

automatically deletes the item currently being presented. You may add additional drill items by typing them in as described above, at any time when the program is awaiting instructions. If an item needs to be changed for any reason, you may do so by deleting the item and typing it in afresh with suitable alterations.

You may also adjust the timing of the program to allow yourself more time to react to an error signal. Simply type in

*T

followed by a digit in the range 1 to 7. The digit represents the number of tenth-second intervals which will be added to the timing constant used by the program. To reduce the timing, use a negative number in the same range. This command may be repeated if necessary to change the timing drastically.

If you become bored or annoyed with the randomized comments, you may inhibit them completely by using the *K (Kill the comments) command. (They will automatically be restored next time the program is read in, unless you dump this version of the program for future use.)

You may also reverse the cue and response fields by using the command

*X

If you choose this option, be sure that the drill items are such as to make this alteration viable. It would not be workable to reverse the fields if two or more of the original response fields happened to be identical.

HOW TO START AND STOP DRILLING:

To begin drilling, press the space bar. The program will immediately begin presenting items in the proper sequence. To stop drilling, press the space bar again while the program is waiting for your response to any cue. If the program runs out of material it will type suitable notification of this fact and pause again. Without an adequate number of items for presentation, the program can not proceed to drill. Either type in additional items, or enter a review tape, using the *RV command.

DRILL PROCEDURE:

The procedure during a drill presentation is very simple: as soon as the cue item has been typed, try to recall the response.

If you have any idea what it is, begin typing immediately. Should you be wrong, the program will ring the teletype bell twice to notify you of the error. It then waits for a time to give you a chance to correct any purely mechanical error caused by inadvertently striking the wrong key. If you should make a typographical error, simply hit the RUBOUT key after the bell has rung, and type the correct character immediately. This will not count as an "error" in evaluating your response to the item (though it will have some effect on the timing of your response). For best results, do not try to correct your input if you do not know what letter is expected. Let the program make the correction for you.

At times, of course, you may be totally uncertain as to the response expected for a given cue. When this happens, you may ask for a character of the response to be typed out by typing the question mark (upper case shift/slash key). Instead of printing out a question mark, the program will print the expected character. If the initial letter does not remind you at once of the correct answer, you may repeat the ? to secure further typeout, or if you prefer you may simply wait. The program will then proceed to type the following characters with a short pause after each letter to see if you remember the rest after one or more characters are displayed. You may interrupt this typeout whenever you wish by typing in the next character. If you do not type anything at all, the program will eventually type the entire response for you. Of course, it will present the same cue again almost immediately if you ask for a hint character or allow timeout to occur, because this is obviously a response you do not know.

Note that the program will wait for a longer time on the first character, and a short time before typing the following characters automatically. When timeout is about to occur, the program will type a few blanks before typing the character. Do not try to type a character after the computer begins its own type out cycle.

It is of some importance that you understand that you will not be penalized for "failure to know." You should try to view the experience as a cooperative effort between yourself and the program to bring you as efficiently as possible to the place where you can readily and with assurance recall certain information. To that end it is to your advantage to indicate to the program as clearly as possible what you know and what you do not know. So by all means correct your typographical errors, so that the program will not interpret them as semantic errors.

If you are not sure of the answer at all, but something pops into your mind, start typing ! Type the first character even if you are afraid it may be wrong. The program will cheerfully and patiently correct you, and in the long run you will probably learn more by such inadvertent mistakes than by freezing until the program times you out and begins typing the right answer.

USE OF ALT MODE KEY TO SIGNAL "I AM SURE OF THE ANSWER:"

Each time you respond to a cue, the program will evaluate your speed of response (compared to your own average) as well as your accuracy, and will determine how soon to present the item again. Sooner or later--and perhaps as soon as the second presentation--you will remember the answer very well. When such is the case, you should press the ALT MODE key once before proceeding. If the computer responds to this signal by printing an exclamation mark and ringing the teletype bell, you should immediately type only the first character of the correct response. The computer will then finish the typeout for you. This saves time and effort, and serves as a kind of intrinsic reward of knowing. It also tells the program something of how you feel about the item. For the moment at least, you are sure you know this answer. This is an important factor in the program's evaluation of your progress in learning.

Of course, such a signal must really represent knowledge on your part. It is easy enough to type the ALT MODE signal automatically, just from force of habit, if you aren't careful. To guard against this possibility, the program requires you to type in the first character of the response correctly on each occasion you use the ALT MODE key; and it also spot-checks your ability to produce the correct answer in its entirety from time to time. In some cases it will respond to your ALT MODE signal with the request:

TYPE FULL ANSWER:

It then waits for you to type the answer in its entirety. As usual, you may correct occasional typographical errors; but if you are wrong, or if you allow the timeout function to operate, the program notifies you that it has lost confidence in your "sure" signal. To rebuild this confidence, it will ask for more frequent confidence checks for a time.

If you do not use the ALT MODE key on a given item, it will continue to be presented rather frequently, and will never achieve test status.

ITEMS READY FOR TEST:

When the program determines that your performance on a given item is consistent enough to make reasonable the assumption that you have learned the proper response, it notifies you that the item is now IN TEST STATUS. This message simply means that you will not see this particular item again for a while, and when next you see it you will be expected to "test out" on it by typing it correctly and at reasonable speed. If you do not "pass" this test, the item is returned to normal drill status for additional practice.

After you have tested out successfully on a number of items in the drill, you may wish to dump these items on tape so you will have room in the file for review items which may need further drill. The command

*PL

will dump these learned items on paper tape and will also delete them from your file. If you wish you may then use this tape of learned items for immediate review.

The choice of items for review is strictly up to you. By keeping track of the order in which you learned the materials, and by observing your own retention rate over a period of time, you should be able to figure out the review procedures most appropriate for you. It is likely that your need for review will depend on the nature of the subject matter and the ways in which you are involved in using the information you are learning. You will also find it of value to observe your own performance with DRILL, and to limit your drill sessions to the length of time during which you are at your best.

In some cases you will fill up the file in the process of reading in the review tape. You may mark the record on which read-in stopped, and begin with this record when you are ready to read in more items for review. Note that any addition to the drill file (other than the automatic conversion of review items to drill items described above) or the read-in of a new drill tape will automatically destroy any review items currently in the file. Punching out of either the learned items or the entire drill file will not destroy the review items, however. There is no provision for punching out review items. You may, however, ask for program status at any time. If review items are available in the file, the number of them will be printed out as part of the status report. Simply type:

*S

to get this report.

SPECIAL CHARACTER FORGIVENESS:

Certain characters in your response are assumed to be of less significance than the rest. As the program is initially set up, these include the space, hyphen, comma, semicolon, colon, period, parentheses, and double quotes. If your type in matches the stored record, the program operates normally. However, if you strike one of these characters in the middle of a word, it will be ignored. No error signal will result, nor will you be penalized. The program simply ignores this kind of error. If the error occurs at the boundary of a word, it will trigger the immediate printout of all consecutive special characters between the two words involved. The program will then look for your first following letter or number character, and match it

against the first letter or number of the upcoming word. This feature is intended to minimize your problems in matching the punctuation of the stored response exactly.

In some cases, you may wish the program to hold you to account for some or all of these special characters. The characters to be given this "forgiveness" treatment are located in a table beginning at location 7600. The contents of the table may be altered from the console if desired. Note that the table must be terminated with a blank word, and that the use of locations above 7611 will cause problems with the binary loader, if the loader is located in the same memory field.

In the special case where the first character(s) of the response happen to be non-significant for checkout purposes, the program will not recognize such characters as fulfilling the initial letter check required in the confidence mode. You must type in the first significant character to verify your knowledge of the response. Thus for example, if you are drilling yourself on Latin verb suffixes, each response in your drill might begin with a hyphen. In the confidence mode, you would then type in the first letter of the suffix rather than the hyphen to trigger the printout of the answer.

One final word before you begin to use DRILL: It is important that you realize its inherent limitations. Like all machines, the computer is an embodiment of organized human logic, i.e., it is a consistent method or system. Every system is inherently and necessarily limited by its particular assumptions, for by definition, no system can transcend its own assumptions. The instant it does so, it ceases to be systematic, and thereby loses the only power it really possesses. Since there is no such thing as a set of universal assumptions which are valid for the totality of our experience, there is no such thing as an infallible machine.

It is important to ensure that if the computer is to be used in the learning process, its limitations remain fully visible to the learner. For this very reason, DRILL does not pretend to inculcate ultimate understandings. It is simply a learner's tool to facilitate the recall of key (if arbitrart) connections, to habituate oneself to certain associations useful in building a total understanding of some area of knowledge. DRILL does not permit you the luxury of alternative correct answers. It is so designed deliberately, for three reasons: 1) it is psychologically unsound to attempt to reinforce two or more different responses to a cue; 2) it is impossible to provide for all the responses which may be considered semantically acceptable by various users; and 3), the half-solution of providing for the more-frequently-used semantic equivalents tends to mislead the learner into presuming that he is mastering final truth. It is more useful to form the habit of formulating one's understandings in a specific, consistent,

memorable way, in the full awareness that the truisms one accepts for functional convenience are oversimplifications. The conscious oversimplification is necessary for learning, and is much less dangerous than the total complex of unconscious oversimplification which has, for each of us, the force of truth.

SPECIAL NOTE ON TAPE READ-IN

At times you may find it useful to read in two or more tapes as part of the same *RD or *RV operation. You may do so, as long as there is room in the file, by stopping the first tape as soon as possible after the trailer code begins to be read. (If you wish, you may stop the reader at the blank space on the tape following any record.) Turn the reader off, dismount the tape, and mount the next tape to be read in at the beginning (arrowhead) of the desired record. If you wish to terminate read in before you have read all the records on a tape, position the tape at the beginning of the trailer area and allow to run until the end-of-read-in message is displayed.

The following chart summarizes the commands and options available in the DRILL program:

COMMAND MODE:

COMMAND	ACTION
*S	Report status
*RD	Reinitializes and reads in drill tape
*RV	Reads in review tape
*PF	Punches entire drill file (does not delete the file)
*PL	Punches and deletes learned items
*T(n)	Adjusts automatic printout timing
*X	Exchanges cue and response fields
*N	Changes name, initializes program for new user
*I	Reinitializes program, clears file
SPACE	Begins drill
*K	Kills randomized comments
*CNTRL/U	Deletes current record (i.e., the one last displayed)
*D(n)	Prints out record n of file, sets up for delete

RECORD EDIT MODE:

CNTRL/U	Deletes all input on current line
RUBOUT	Deletes one character (=backspace and erase), may be repeated
/	Field separator
CARRIAGE RETURN	End of record signal
Initial space illegal	Initial space interpreted as go-to-drill signal

DRILL MODE:

ALT MODE key	Assertion of confidence
*	Interruption of presentation to give command
RUBOUT (after bell)	Permits entry of correction
SPACE	Halts drill and returns to command mode
?	Displays next letter of response (may be repeated)

SOURCE TAPE NO.	NAME	OTHER SOURCE ROUTINES INCLUDED
1	ACCEPT	ECHO
2	ASMFK	
3	ASSERT	LOSSCF; CFAIL
4	CCPRI	
5	CHKOUT	
6	COMMENT SETS	
7	DELETE	
8	DELETR	
9	CHECKD	TSTATM; INZTAB; SLOW; TIMRPT
10	DIVR	XCHNG; DECR; GREET
11	DRILL	ASSESS; CMEND
12	EVRSP	CMPUTE
13	FNDFLD	FINDR
14	FNDPRI	
15	FUDGE	OUTPUT; RECON; CHK
16	GETNAM	
17	INCRTB	
18	INTLZ	PRTREC
19	IOR	XOR; ADD2; SUBT2; DIV2; PSTAR
20	(1) MESSAGE FILE (2) CONTINUED M.F.	
21	MONKBD	INZITM; MNKBD2; IDTFY; NULL
22	MTPLY	
23	NAMCHG	INZFIL; LOAD2; ERROR; PRTOCT; G1CHAR; ERR1
24	PNCHRC	
25	PRMPT	AJCFMK; DASH; HINTS; STEP
26	PRTCOM	
27	PRTDEC	
28	PRTFD	CPCOD; PRTMSG
29	PUNCH	
30	RANDOM	
31	RDKBD	
32	RETURN	REPEAT; SPACE
33	RVITM	CHKMOD
34	RVTAPE	REPORT
35	SEARCH	
36	SPEV	RVWRPT
37	STCHAR	
38	STOBLK	READEL; STOREL
39	TAKE5	
40	TESTL	MRKTIM; TIMER; INPUT; VARTIM
41	TIMER2	
42	TYPE	
43	UPTIME	
44	WRONG	
45	PAGE Ø (1) Ø - 77 (2) 1ØØ - 177	

CORE MAP FOR SELFDRill

PAGE	AREA	CONTENTS
0	0 - 77	(See listings for page 0)
	100 - 177	Subroutine linkage area (see listings)
1	200 - 336	DRILL
	345 - 376	ASSESS
2	400 - 541	SPEV
	544 - 552	INCRTB
	553 - 560	RECON
	561 - 576	RVWRPT
3	600 - 732	RVTAPE
	733 - 776	REPORT
4	1000 - 1176	CHKOUT
5	1200 - 1334	PUNCH
	1335 - 1373	PNCHRC
	1375 - 1377	(PUNCH)
6	1400 - 1441	WRONG
	1442 - 1534	DELETR
	1535 - 1567	INTLZ
	1570 - 1577	ECHO
7	1600 - 1774	CCPRI
10	2000 - 2012	TAKE5
	2013 - 2050	MRKTIM
	2051 - 2104	TIMER
	2105 - 2111	Command code table for RDKBD
	2115 - 2127	INPUT
	2130 - 2141	VARTIM
	2142 - 2171	GETNAM
	2172	GATE Controls time before type out begins
	2174 - 2175	RANDTB (See ASSERT routine)
	2176 - 2177	NAMUSE
11	2200 - 2224	MTPLY
	2225 - 2240	IOR
	2241 - 2254	XOR
	2255 - 2314	ADD2
	2315 - 2336	SUBT2
	2337 - 2365	DIV2
	2366 - 2377	PSTAR
12	2400 - 2427	STOBLK
	2430 - 2437	READEL
	2440 - 2451	STOREL

PAGE	AREA	CONTENTS
12	2452 - 2472	NAMCHG
	2473 - 2511	SEARCH
	2512 - 2574	FUDGE
13	2600 - 2631	PRTFD
	2621	EP OF CPCODE
	2632 - 2723	PRTMSG
	2726 - 2741	DIVR
	2742 - 2754	XCHNG
	2755 - 2765	DECR
	2766 - 2776	GREET
14	3000 - 3052	PRTDEC
	3053 - 3074	INZFIL
	3075 - 3104	ERROR
	3105 - 3127	PRTOCT
	3130 - 3155	G1CHAR
	3156 - 3170	ERR1
	3172 - 3177	SPACE
15	3200 - 3242	FNDFLD
	3243 - 3272	FINDR
	3273 - 3337	TIMER2
	3340 - 3370	RVITM
	3371 - 3376	CHKMOD
16	3400 - 3426	STCHAR
	3430 - 3511	ASMFK
	3512 - 3520	ECHO
	3522 - 3562	RANDOM
	3563 - 3577	TESTL
17	3600 - 3607	TYPE
	3610 - 3621	RETURN
	3622 - 3634	RPEAT
	3635 - 3777	ACCEPT
20	4000 - 4033	FNDPRI
	4035 - 4037	NULL
	4040 - 4077	DELETE
	4100 - 4137	PRITAB
	4140 - 4177	HISTAB
21	4200 - 4237	LSTPRI
	4240 - 4247	RTCTB
	4300 - 4337	TIMTAB
	4340 - 4377	EXPTB
22	4400 - 4572	ASSERT
	4574 - 4577	MESSAGE
23	4600 - 4673	MONKBD
	4700 - 4721	INZITM
	4722 - 4727	MNKBD2

PAGE	AREA	CONTENTS
23	4730 - 4747	IDTFY
	4751 - 4777	MESSAGES
24	5000 - 5177	MESSAGE FILE
25	5200 - 5377	
26	5400 - 5575	RANDOM COMMENTS FILE
	5575 - 5577	MESSAGE
27	5600 - 5643	INPUT BUFFER
	5644 - 5735	RDKBD
	5737 - 5776	UPTIME
30	6000 - 6070	EVRSP
	6072 - 6111	MESSAGES
	6112 - 6132	PRMPT
	6133 - 6142	AJCFMK
	6143 - 6152	DASH
	6153 - 6166	HINTS
	6170 - 6177	STEP
31	6200 - 6251	PRTCOM
	6255 - 6314	CHECKD
	6315 - 6332	TSTATM
	6335 - 6345	INZTAB
	6346 - 6361	SLOW
	6362 - 6375	TIMRPT
	6376 - 6377	MESSAGE
32	6400 -	
33		
34		DRILL FILE
35		
36	7577	
37	7600 - 7611	SPECIAL CHARACTER TABLE
	7612 - 7777	LOADERS

Functional Description and Program Notes

SELFDRill is a general-purpose utility program to enable a learner to master any verbal materials of his choice to the point of instant recall. The program operates basically in two modes: 1) a file input mode, in which it accepts from the keyboard or reader a series of items in the form CUE/RESPONSE, and creates a drill file of up to 31 such items; and 2) a drill mode, in which it presents items from the in-core drill file and then monitors and evaluates the user's response to each such presentation. Detailed information on various elements of the user's response is saved in a series of tables. This information is updated each time an item is presented. On the basis of the learner's experience with each particular item, his response time on this particular occasion, the quality of his response, and his assurance, the program predicts how "long" (i.e., after how many intervening items) he will be able to remember the correct response successfully. It then presents the item again after the computed interval, evaluates the response again, and continues this cycle until the interval exceeds a certain arbitrary threshold. When this happens, the item achieves test status. The next time it is due for presentation, it is labelled as a test item. If the learner types the expected response quickly and accurately, the item is adjudged learned and is no longer presented.

Review items are used to help provide proper spacing between the drill items. If answered correctly, these items are discarded after the first presentation. If not, they become part of the drill file. Learned items may be punched out and deleted from the file so as to leave room for further review items to be added as necessary. Experience with the program has verified that it is virtually impossible to be actively drilling on as many as 31 items at any given time, so that there is little possibility of running out of space in the file. If there are very many items in the file -- say twenty-five or more -- some of the earlier ones will be learned and tested out before the later ones are encountered at all in the drill process. If the individual items are very long, of course, the file area will be filled before the maximum number of items has been stored. The user whose application requires consistently long drill items might well consider modifying the program to use a different core field for the drill file, or may wish to keep the file on a disk or DECTAPE if he encounters such a problem. Thus far at Beloit, the problem has not occurred.

It is anticipated that users with intermediate access-speed storage will find it useful to adapt SELFDRill to enable the use of larger on-line files. To that end, the listings have been liberally annotated. Listing documentation includes the core map, definitions of page zero usage, and a special designation of entry points referenced in off-page branch instructions.

Program architecture: SELFDRill consists of some 84 routines, most of which are standard closed subroutines called by a JMS I using a page zero pointer. Locations 100 - 177 of page $\emptyset$ are reserved for an address table for this purpose. The mainline program, DRILL, is located at page 1 with starting address 200_8. DRILL consists of two basic loops: a data mode loop, in which drill items from the keyboard are accepted for storage as records in the drill file, and a drill loop, in which items from the file are displayed for user

response, with evaluation of results.

From either loop the user may register commands to perform the various functions listed in table form in the User's Guide (See ACCEPT listing). MONKBD is the basic routine for the data mode loop, and CHKOUT the routine for drill mode.

Timing is accomplished by means of a short loop in the TAKE5 subroutine. As the name implies, this routine simply takes 5 milliseconds to execute. Because the instruction time varies with different models of the 8, a different time-out constant is required in some instances. As assembled, the constant is set up for the PDP8-E (Location 2011, MTIME, contains 5334_8). For the PDP8-L, this value should be changed to 5757_8, and for the PDP-8I, it should be 5651_8. Since instruction times are nominal ("All computer times are ± 20%", says the PDP8-I/L handbook, p. 33), these adjustments of the constant may be an over-refinement. The timing constant determines the interval (nominally one tenth of a second) which the program uses as its basic unit in measuring the relative response time of the learner and in timing its own type-out rate.

The FUDGE subroutine allows the user some latitude in reproducing the punctuation of the expected response. This feature is especially important for the learner-designed drill, because it permits the user to be less than totally consistent in his use of punctuation without holding him rigorously to the exact reproduction of his inconsistent usage. The table of ASCII codes which are defined by FUDGE as non-significant is located beginning at 7600. This table originally includes the space, hyphen, comma, colon, semicolon, period, parentheses, and double quote. It may be changed simply by storing the desired series of ASCII codes beginning at 7600. The list must be terminated with a location containing zero. (This list is assembled with the Continued Message File.)

The priority assigned each drill item after each presentation is determined by the routine WRONG or by CCPRI if the answer was correct. The priority scheme consists of a table of negative numbers, each of which is incremented when an item is being selected for presentation (See FNDPRI). When the priority of a given item reaches the value -1, it is held at that value until it has been presented. In case of a pile-up of several items due for presentation, the "youngest" (i.e., the item with which the learner has had the least experience thus far in the drill) is presented first. This tends to keep the spacing closer to the norm for those items where the likelihood of poor recall is greatest, and where the interval between presentations is therefore the most sensitive.

When an item is adjudged wrong, the priorities of all items in the file are advanced by one count in recognition of the additional interference produced by such difficulties.

The RANDOM subroutine generates a pseudo-random number in the range 0 - 7. The subroutine simply adds the contents of successive memory locations within the field until the accumulator overflows, and then takes the low-order 3 bits of the accumulator as the number. The routine compares this number with the contents of a two-word table (supplied by the user for each application) to ensure that the number chosen is different from the ones used on either of the

last two usages involving the specified table. Therefore the number is not truly random. A test program designed to determine relative frequency of each of the eight digits indicated almost equal probability for each of the eight digits.

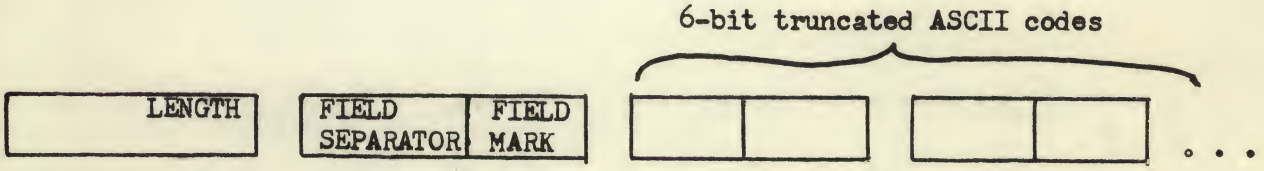
The frequency with which a complete check is made of the learner's ability to type the answer (in the confidence mode of response) is determined by a variable mask generated by the ASSERT subroutine whenever confidence is asserted, and updated when the user's response is evaluated. If he succeeds. the mask is modified by AJCFMK to reduce the probability of further checks; if he fails, the ASSERT routine increases the probability to 100% on the next use of ALT MODE, and the user must reestablish a high confidence level as he did at the beginning of the drill. Probability of a check never decreases below 1 chance in 16 (6.25%). This threshold was determined by experience in using the program. Maximum confidence is attained by three successive correct performances after TYPE FULL ANSWER challenges.

Evaluation of the user's performance (see EVRSP listing) is determined in part by the amount of time required for his response. Both the initial time lag and any delays incurred during type in are taken into account, and the learner's response is compared, for evaluation purposes, to his own "average" at the moment. This "average" is actually a model figure, derived by incrementing a nominal average each time the measured response time exceeds it, and decrementing each time the response time falls below it (see UPTIME listing). Thus, extreme variations have no more effect on the "average" than modest ones, and yet the figure represents a midway time which will change to reflect individual typing speed, the subject's relative alertness at the moment, and the relative difficulty of the items being encountered. "Fast" and "Slow" responses are identified in relation to this statistical mode.

A "comments" area in core contains 5 records of 8 fields each. If the record format is honored, these may be replaced by other comments of the user's choice. The five records are used by the program to cover five different cases:

1. General commendation
2. Wrong answer (confidence mode)
3. Exceptionally prompt
4. Exceptionally slow response
5. Recognition of successful response to a confidence check

The format of each record is as follows:



Fields must terminate with the field separator $\emptyset\emptyset_8$. The record length in the first word of the record must include all words up to the next record length indicator at the beginning of the next record.

Drill records are assembled by MONKBD (or MNKBD2, if read into the file from tape) in this same format. Review records are identical in format with drill records, and are distinguished from the latter only by virtue of being stored

in the file area above the drill file limit set by indicator DRFILX. When review items are present, DRFILX always holds the address of the next review item available for presentation. After a review item has been presented, the program determines whether to discard it from the file or include it in the drill. If the former, the remaining review items are moved down in the file to overlay and destroy the item; thus a new review item is available at the same address; if the item is to be transferred into the drill, DRFILX is simply changed to address the next item in the review file, and the tables and counters are adjusted to reflect the new status of the drill file (see subroutine SPEV).

SPECIAL NOTE ON ALPHANUMERIC FIELD ADDRESSES:

This program employs the convention of a character address (CHARAD) formed by shifting one place to the left the beginning address of a character string. The high order bit (bit Ø of the memory address) is always assumed to be 1, so that all character strings must be stored in the area above 3777_8. The low order bit (bit 11) contains Ø if the character occupies the left half of the memory location, and 1 if the character is in the right half. Character codes are stored in truncated 6-bit ASCII codes. A blank code ($ØØ_8$) is interpreted as a field terminator.

Drill and review records are identical in format: the first location in each record contains the length of the record (in machine words); each record contains two fields, and each field consists of a blank, a field mark (01_8 or 02_8), and a character string. Records are identified simply by their relative position within the file.

ADDENDUM TO DECUS NO. 8-656

I have several refinements of the current version of SELFDRill which I plan to submit in a revision, shortly after January 1 of next year. The changes are minor but important. They include the capability of using the asterisk in a drill record (in any position other than the initial character), simplification of the delete record command, and correction of the CR/line feed insertion on certain long items on punch-out of the file, which can ruin a drill item inadvertently when the tape is made. I would also like to include a special comment when the user is too lax about hitting the confidence key on items he obviously knows. I also have a separate version of the program for use with the VT05.

I would be more than happy to send a tape of the new version, plus preliminary documentation of the changes, to anyone who would like to write me for it directly prior to the official revision.

Sincerely,

F. M. Wheeler

F. M. Wheeler
Associate Professor
of Classics

Beloit College
Beloit, Wisconsin 53511

DECUS 8-656 - SLFDRL, The Sloan Self Drill Program

This program is currently being revised by the author. Future versions will allow for larger core storage and for use of the VTØ5. Evaluation forms on which the user may record his results using the SLFDRL program are available from the author. He will be happy to send them to anyone, to correspond with anyone concerning this program and the impending revision.

Please call or write to:

Francis M. Wheeler, Ph.D.
540 Kenwood Avenue
Beloit, Wisconsin 53511
Tel: 608-365-6841