

record over the program.

Now begins the tricky process.

When you do get the words/phrases, you will find them located as follows:

Register 0: your own word/phrase  
Register 2: the owl: o.o  
Register 4: S. Barbara Ca.  
Register 6: Colorado

Press f W/DATA. "CrD" will appear in the display. Take your Work 2 card and enter it into the right hand side of the 67, while placing your other hand on the off-on switch. When between one-third and one-half of the card has moved past the insert point, turn the machine off. None of the card should yet be visible on the left hand side. Turn the machine back on and let the card finish passing through. Press any key to clear "Error". Press DSP 9. Read the card through the calculator. Press any key to clear the "Error" display this will produce. Then press RCL 6. If you are lucky, you'll see Colorado in the display, and all of your other words will be in their respective registers. If not, you most likely will see one of the following other 7 displays:

-C.olorodo 0 90  
-o.lorodo 00-85  
-d.EOE566E78 48  
-E. 072C373o 24  
- .783959C9E 12  
-9.999999999 99  
0.000000000

If one of these, or something else, appears when you check register 6, press f W/DATA and start the card through, switching-off when it is one-third to one-half way through as before, and try again. Repeat the process until you get it. Until Colorado is found in Register 6, none of the others will be right either. Have fun!

If you create any interesting prompts, displays, etc. that you want to add to the HP-67 Word/Phrase Library, please send them to Lee Skinner (256), Box 14944, Albuquerque, NM 87111.

#### THE "GERALD AND IDA PROGRAM"

Lou Cargile, Jr. (753)

This is the program I mentioned last month (V4N1P8) which some of you have already had an opportunity to play around with. Since it has always lacked a name, I've come to refer to it by the above. Lee Skinner is listing it in the 67 Word/Phrase Library program section as P4--Communications Dialog (Gerald, Ida, Ella) and I guess as long as Lee's card-reader holds up under your requests you can get it from him by sending blank cards. Let's hear it for Lee.

This current version, the Library version, has some loose ends tucked in (the primary and secondary registers are no longer subject to unintended reversals for one thing) but it is essentially what I briefly described last time.

In operating the program, you should consider all of the calculator's keys off-limits except for: the A key (which functions as a start or re-set button); the 12 white keys (numerals plus 0, ".", and R/S); and the CHS, CLx, and EXP keys. These 15 keys serve as the "terminal." I use an overlay that is essentially John Hausch (88)'s with the slight modification of designating the numeral "3" key as also Y,Z,7 (instead of Y,Z,8lk) and designating either the 0 or "." keys as an "Execute" key. In this usage, in effect, each of the numeral keys represents (with of course a four-way ambiguity) any one of four characters, three of which are letters (in one case a "?") and one of which is a numeral. In other words, to key in an instruction such as REPEAT you would press the white keys in succession corresponding to the letters. In this example doing so would cause 686871. to appear in the display. Try to ignore it there.

There are (currently) three data cards intended for use with the program. (I hope some of you will come up with some worthwhile additions to the list--send them to me or to Lee.) Each card corresponds to a calculator persona--Old Gerald, Easy Ida, or Dear Ella. (If you create a new one you'll need to know this: the primary and secondary registers are reset correctly each cycle by comparing the equivalent numerical values of the displays stored in R6, RA, and R56. You should arrange things so that  $R6 \leq R_A$  and  $R56 > R_A$ .)

To operate the program, simply read one of the three

data cards Gerald, Ida, or Ella into the calculator, read the program card, and press A. The calculator goes into a looping 1-sec.-pause cycle. During the pause the display shows o o o o which indicates readiness for a communication from you.

Such keyed-in communications fall into two categories: 1) the serious business of numerical computations accomplished through mnemonic instructions; and 2) all other commands, comments, exhortations, questions, laments, etc. which you may care to address to Gerald, Ida, or Ella as the case may be.

The correctly processable computation instructions are strictly limited to a set of 13 and there are no error traps so you're on your honor. Keying them in differs from keying in all other instructions in that the first key pressed is the decimal point key followed by the three, four, or six "letter" keys corresponding to the instruction. The set, which is self-explanatory, is:

.ADD (i.e. .788)	.ARCSIN
.SUB	.ARCCOS
.MULT	.ARCTAN
.DIV	.LOG
.SIN	.ANTLOG
.COS	.RCP (reciprocal)
.TAN	

To do a computation such as addition, key in .ADD during the pause (or press R/S, key in .ADD, and press R/S to resume.) The pause display will change to show, in Gerald's case, "I is ready" and this display will continue to loop until you key in the first number (or operand) of the two which you wish to add. Key it in during a pause (or using R/S as above) and the number keyed in will replace "I is ready" in the looping display. On the first pause, or any subsequent pause, key in the second number and this will then appear in the looping display. (If you continue this process, the last two numbers entered will always be the two operands on which, in this case, addition is performed so corrections can be made this way without starting over. Maybe we could call this a stack or something.)

With the two numbers thus entered press zero or the decimal point key as an "Execute" instruction (notice how I'm trying to avoid calling it an = key) and the result of the addition appears in the looping display and continues there until a new instruction is keyed in.

Monadic operators such as .SIN are handled in the same way except that of course it is necessary to key in only one operand or argument before giving the "Execute" instruction. Thus to compute  $\sin 30^\circ$  you would key in .SIN, see "I is ready" in the looping display, key in 30, see it in the looping display, press zero or the decimal point key and see the answer 0.500000000 in the looping display.

It is possible to do chain calculations. When "I is ready" appears in the looping display following a keyed-in computation instruction, you may recall the result of the last previous computation (even after intervening non-computational exchanges with Gerald or whoever) by pressing zero or the decimal point key. The previous result will appear in the looping display and now becomes the argument for the instructed operation (or the first of the two operands for a dyadic operation such as multiply) just as if you had keyed it in at this point. (It should be obvious by now that the one quantity which may not be entered as an operand or argument is zero. Just memorize  $\sin 0 = 0$ ,  $\cos 0 = 1$ ,  $\tan 0 = 0$ , and do the rest in your head.)

So much for serious business. Now for the fun. Obviously, all the above is only a show-off way of giving the pseudo-terminal something ostensibly purposeful to justify its ambiguous existence. The instruction "set" to which the calculator responds when you key in a letter sequence not preceded by a "." is not restricted. Although a considerable number of instructions were originally chosen and the program designed to give appropriate responses to these, there will be some kind of response to any input (i.e. a sequence consisting of one or more letters) and rather amazingly a large proportion of these unplanned for instructions elicit responses that can be taken as consistent with the input. (This is in no small part due to a careful ambiguity built into the responses. Take this necessity into account if you create a Sara or Eric or whoever.) A two-word instruction is keyed in as one word, e.g. Do Pi is keyed in as DOPI (8567). Gerald responds "Big deal" then shows you pi correct to nine decimals! (Remember?)

A list of some useful inputs appears at the end of this. Any of these are keyed in exactly as are the computational instructions--but remember, without any decimal point--during the looping pause, either when the display contains o o o o or when it contains the result of a completed computation.

For example, have Gerald add two numbers and when the result appears key in GOOD (9558). He responds "I agree." If you sarcastically key in MODEST he replies impatiently "O good God." You caution him--WATCHIT. He hedges with "Easy, Boss." Unmollified you key in GOOFOFF, get back "Boss yell", which you deny--DIDNOT. Gerald replies "Disagree". You shout WHAT? He insists "Is so Boss," you repeat DIDNOT firmly and he gives in respectfully--"Yes sir."

As this demonstrates there is some provision in the program for the response elicited being altered by what precedes it but this is true only to a very limited extent and is one direction in which improvements in this whole idea would be very welcome.

There are a few inputs which are simply dismissed without response, but there is another class which requires explanation. Gerald (or either of the others) can take offense and when he does he sulks or angrily refuses to communicate. Several words or phrases induce this behaviour. Some of them (YOUSOB, FASCIST) cause the reply "Goodbye" more in sorrow than anger, and all subsequent verbal inputs from you are simply ignored. Others (EGOTIST, DAMN, LOUSEDUP, SLUT) cause Gerald to lash back "Boss also" before going into a moody silence. In this mode, business is carried on as usual over at the computation window, although Gerald's cheerful "I is ready" is replaced by a blank string of zero's when you key in your calculation instruction. It will be carried out efficiently, but in silence as it were with no chit-chat. When you have had enough of this cold shoulder treatment, key in ISORRY. Gerald responds with a gracious "Also sorry" and everything is back to normal.

At any time during the o o o o o pause you can read one of the other data cards and switch to Ida or Ella. The display announces the name of the new party you are dealing with and then returns to the o o o o o as before, ready for you to carry on. Ida is a street-wise gal and I think she may moonlight as a hooker on the side. Her replies are disrespectful, inelegant if not bawdy, and whereas Gerald tries to calm your anger ("Easy, Boss"), Ida delightedly eggs you on ("Go, boss, go"). Ella on the other hand is overly fastidious, given to excessively enthusiastic if not downright fawning acceptance of instructions and probably frigid. You'll form your own impressions of these three if you try the program. Maybe they constitute a sort of Rorschach inkblot and what you perceive is what you project.

As I've mentioned to several of you, I think the use of John Rausch's keyboard designations and a program technique such as used here could produce some interesting game programs in which, taking football as an example, the choice of plays would be made by keying in PUNT, PASS, or etc. during a pause rather than by consulting a table for a numeral or a key combination to enter as was necessary with similar 65 programs.

As for this program--get involved with the heart and mind of the calculator. Brag on it (PERFECT). Argue with it (WRONG). Insist (ISSO). Deny (ISNOT). Curse (GODDAM). Apologize (SORRY). Take pity on it (TAKES). Caution it (CAREFUL). Hurl epithets (COMMIE). Counsel it (LOVEOTHERS). With Ida proposition it (MYPLACE?), (5DOLLARS), (8), (15?). Question it (YOUOK?). Commend it. (VERYGOOD).

Don't expect perfection, but prepare to be surprised. Have fun.

Maybe one last word is in order. You must always--remain in command. The calculator must in the final analysis remain subservient to you. Any reversal of roles could prove disastrous. At the first sign of any such occurrence switch the calculator to off and immediately burn the program card which you used. Only then should you telephone your report to Richard. With this borne in mind the program can safely be used by any sufficiently paranoid adult. Hang in there, Ida, I'm coming.

#### SOME SUGGESTED INPUTS FOR THE "GERALD AND IDA PROGRAM"

##### General:

REPEAT (repeats last display)  
ANSWER (calls up result of last previous computation)

NAME  
DOFI (do pi)  
END  
READY?  
GERALD  
HELPER  
YES  
NO

##### Pejorative or expletive:

##### Complimentary:

GOOD  
PERFECT  
NIFTY  
WELLDONE  
PLEASE  
THICKY  
OKAY  
NEAT  
ACK  
NICEWORK

QUESTION  
SORRY  
TRY  
GOODBYE  
DOIT  
SOFTWARE  
SOMEPARTY  
YOU'RE FIRED  
TOWORK  
WHOCARES  
DONTGO  
BEGOOD  
AGAIN  
DIDNOT  
DIDSO  
COMEON  
ISSO  
ISNOT  
NOGOOD  
DONT  
DOTHIS  
CAUGHTYOU  
LEAVE  
ANGRY?  
GIVEIN  
MAYDAY

##### Ideological:

NAZI  
FASCIST  
RADICLIB  
AGNEW  
NIXON  
RADICAL  
LIBERAL  
COMMIE  
PINKO

DROPDEAD  
SLOPPY  
UPYOURLS  
STUPID  
NUMSKULL  
DAMPOOL  
SHUTUP  
DAMMIT  
LOUSY  
LIAR  
BUGOFF  
FOOL  
SLIPSHOD  
AWFUL  
GODDAMN  
YOUHAM  
YOUDOPE  
BALONEY  
IDAREYOU  
CLOWN  
DUMMY  
BREAK  
DIMWIT  
LAMEBRAIN  
SMARTASS  
ACIDHEAD  
DULLARD

##### Sympathetic:

CHINUP  
COURAGE  
TAKES  
RESTUP  
YOUOK?  
REST  
TIRED?

##### Cautionary or admonitory:

HURRYUP  
WATCHIT  
BE CALM  
THATS1  
THATS2  
THATS3  
CAREFUL  
WATCHOUT  
QUIET

##### Adverse:

ERROR  
WRONG  
SLOW  
POOR  
WHAT?

##### Religiously oriented:

YOUSINNER  
BORNAGAIN  
GODISLOVE  
BESAVED  
GODCARES  
SEABROTHER  
LOVEOTHERS  
LOVEPEOPLE  
REPENT  
PRAY  
BLESSYOU  
JESUSSAVES  
CONFESS

\* a number of possible four-letter words are omitted from this list Louis Cargile (753)

R/S

## DISPLAYS USING F LBL 1 BY LOUIS CARGILE JR. (753)

The kinds of synthetic brief pause displays which can be made by use of a string of 31 25 24 (f LBL 1) program steps (which I referred to briefly in the article in V3N1 on page 7) have turned out to be more interesting and useful than I realized at that time. Learning how this operates has made possible a variety of novelty displays--moving messages (HP Commercial), some animations, and, more significantly, ways of formatting program-generated data and displaying it--briefly-- with spaces, hyphens, decimal points, inserted as desired, as illustrated, somewhat shallowly, by the "cuckoo clock" program. A number of program sets in Lee Skinner's (256) expansion of the Word/Phrase Library illustrate what understanding of the principles can enable you to do.

The duration of the display which appears briefly, with a slight flicker but nevertheless steadily, while the program is traversing a continuous series of f LBL 1 steps in program memory depends of course on the number of such steps used. The visible display itself consists of the sign digit followed by the ten mantissa digits (or characters) of the contents of some internal register (or the corresponding first eleven digits of the fourteen in that register) as they are modified by the contents of the x-register as it exists at the beginning of the string of f LBL 1's. The particular internal register involved, and the nature of its contents, is dependent on the program instruction immediately preceding the first f LBL 1. (An intervening step such as CLX or RCL n which does not affect the internal register may occur.) I'll get to that in a moment.

The modification is on a digit-by-digit basis. If the x-register contents are zero (which means the first 11 digits of that register are a zero sign digit followed by 10 mantissa zero) there is no modification at all. In other words, wherever there is a zero in a particular position in the x-register, the corresponding digit in the internal register is displayed unaltered. At the other extreme, a 2 at a particular position in the x-register completely suppresses (blanks) a 0, 1, 7, or hex F (which is a blank space anyway.) Other digits interact to produce any of the possible 16 digits or characters with a decimal point superimposed, a - sign, or a - sign with a decimal point superimposed.

A complete description of this interaction is listed below:

## User instructions

## 1. Follow instructions, V4 N2 Pl6&amp;l7

001 f LBL 1	31 25 01	055 x	71
002 f GSB 0	31 22 00	056 DSP 0	23 00
003 RCL A	34 11	057 f RND	31 24
004 RCL 6	34 06	058 DSP 9	23 09
005 g $X \leq Y?$	32 71	059 h RTN	35 22
006 f $P \leftrightarrow S$	31 42	060 g LBL a	32 25 11
007 h R ↓	35 53	061 f LOG	31 53
008 h R ↓	35 53	062 h RTN	35 22
009 1	01	063 g LBL c	32 25 13
010 g $X > Y?$	32 81	064 g $\cos^{-1}$	32 63
011 GTO 2	22 02	065 h RTN	35 22
012 f GSB 3	31 22 03	066 f LBL 0	31 25 00
013 f $X = 0?$	31 51	067 h PAUSE	35 72
014 6	06	068 h F? 3	35 71 03
015 ENTER	41	069 h RTN	35 22
016 9	09	070 GTO 0	22 00
017 g $X = Y?$	32 51	071 RCL C	34 13
018 h CF 0	35 61 00	072 h RTN	35 22
019 f GSB 7	31 22 07	073 RCL 2	34 02
020 h F? 0	35 71 00	074 h RTN	35 22
021 GTO A	22 11	075 RCL B	34 12
022 6	06	076 h RTN	35 22
023 f GSB 7	31 22 07	077 RCL 1	34 01
024 7	07	078 h RTN	35 22
025 f GSB 7	31 22 07	079 RCL 2	34 02
026 8	08	080 h RTN	35 22
027 f GSB 7	31 22 07	081 RCL 0	34 00
028 2	02	082 h RTN	35 22
029 1	01	083 RCL $\Sigma^+$	34 21
030 f GSB 7	31 22 07	084 h RTN	35 22
031 2	02	085 RCL 5	34 05
032 x	71	086 h RTN	35 22
033 9	09	087 RCL 8	34 08
034 5	05	088 h RTN	35 22
035 +	61	089 h SPACE	35 84
036 CHS	42	090 f $P \leftrightarrow S$	31 42
037 h $X \leftrightarrow I$	35 24	091 RCL 6	34 06
038 RCL 2	34 02	092 h F? 2	35 71 02
039 f $P \leftrightarrow S$	31 42	093 g GSB d	32 22 14
040 f GSB (i)	31 22 24	094 h RTN	35 22
041 h STO I	35 33	095 h SF 2	35 51 02
042 f -x-	31 84	096 f $P \leftrightarrow S$	31 42
043 f LBL A	31 25 11	097 RCL 9	34 09
044 h CF 3	35 61 03	098 h RTN	35 22
045 RCL E	34 15	099 RCL A	34 11
046 GTO 1	22 01	100 h RTN	35 22
047 f LBL 3	31 25 03	101 h SF 0	35 51 00
048 x	71	102 h RTN	35 22
049 5	05	103 RCL 7	34 07
050 1	01	104 h F? 2	35 71 02
051 ÷	81	105 h RTN	35 22
052 g FRAC	32 83	106 GTO f D	22 31 14
053 5	05	107 h SPACE	35 84
054 1	01	108 h SF 2	35 51 02

(OVER)

109 RCL 6	34 06	171 GTO f a	22 31 11
110 h RTN	35 22	172 h 1/x	35 62
111 RCL 3	34 03	173 x	71
112 h RTN	35 22	174 h RTN	35 22
113 RCL 8	34 08	175 GTO A	22 11
114 h RTN	35 22	176 +	61
115 RCL B	34 12	177 h RTN	35 22
116 h RTN	35 22	178 f TAN	31 64
117 g LBL d	32 25 14	179 h RTN	35 22
118 f P↔S	31 42	180 f LBL 7	31 25 07
119 RCL 7	34 07	181 g X=Y?	32 51
120 h RTN	35 22	182 f P↔S	31 42
121 RCL 4	34 04	183 h R↓	35 53
122 h RTN	35 22	184 h RTN	35 22
123 RCL 7	34 07	185 g TAN-/	32 64
124 h RTN	35 22	186 h RTN	35 22
125 h SF 0	35 51 00	187 g SIN-/	32 62
126 h SPACE	35 84	188 h RTN	35 22
127 RCL 5	34 05	189 f SIN	31 62
128 h RTN	35 22	190 h RTN	35 22
129 RCL 1	34 01	191 GTO f c	22 31 13
130 h RTN	35 22	192 -	51
131 RCL 4	34 04	193 h RTN	35 22
132 h RTN	35 22	194 g 10×	32 53
133 RCL 2	34 02	195 h RTN	35 22
134 h RTN	35 22	196 h 1/x	35 62
135 RCL 5	34 05	197 h RTN	35 22
136 h RTN	35 22	198 h PAUSE	35 84
137 RCL 9	34 09	199 f COS	31 63
138 h RTN	35 22	200 h RTN	35 22
139 RCL 6	34 06	201 f LBL 2	31 25 02
140 h RTN	35 22	202 h R↓	35 53
141 RCL 0	34 00	203 EEX	43
142 h RTN	35 22	204 5	05
143 RCL 3	34 03	205 f GSB 3	31 22 03
144 h RTN	35 22	206 5	05
145 RCL 5	34 05	207 0	00
146 h RTN	35 22	208 -	51
147 h SF 2	35 51 02	209 h X↔I	35 24
148 f P↔S	31 42	210 RCL A	34 11
149 RCL 8	34 08	211 h F? 0	35 71 00
150 h RTN	35 22	212 0	00
151 RCL 3	34 03	213 f GSB 0	31 22 00
152 h RTN	35 22	214 f X=0?	31 51
153 RCL 2	34 02	215 RCL D	34 14
154 h RTN	35 22	216 f LBL 9	31 25 09
155 RCL 1	34 01	217 f GSB 0	31 22 00
156 h RTN	35 22	218 f X≠0?	31 61
157 RCL 0	34 00	219 GTO 9	22 09
158 h RTN	35 22	220 h R↓	35 53
159 RCL D	34 14	221 f GSB (1)	31 22 24
160 h RTN	35 22	222 STO D	33 14
161 RCL B	34 12	223 h STO I	35 33
162 h RTN	35 22	224 h SPACE	35 84
163 h X↔Y	35 52		
164 h RTN	35 22		
165 RCL 2	34 02		
166 h RTN	35 22		
167 RCL 8	34 08		
168 h PAUSE	35 72		
169 h π	35 73		
170 h RTN	35 22		

# Gerald data card

## Primary

0. DO OR DIE  
1. OLD GERALD  
2. EASY BOSS  
3. BOSS YELL  
4. BOSS CRAZY  
5. GOODBYE  
6. IS SO BOSS  
7. YES SIR  
8. BIG DEAL  
9. REAL COOL  
A. I IS READY  
B. O GOOD GOD  
C. YE GODS  
D. 7.00  
E. ☐ ☐ ☐ ☐ ☐  
I. OLD GERALD

## Secondary

0. YES BOSS  
1. LA DE DA  
2. BOSS ALSO  
3. ALSO SORRY  
4. GOD BLESS  
5. I AGREE  
6. I IS GOOD  
7. DISAGREE  
8. GOOD IDEA  
9. I DECLARE

ABC	DEF	GHI
7	8	9

JKL	MNO	PQR
4	5	6

STU	VWX	YZ?
1	2	3

Execute  
0 .

# Ida data card

## Primary

0. EASY BREAD  
1. EASY IDA  
2. GO BOSS GO  
3. CRAZY BABY  
4. BIG DADDY  
5. BYE BYE  
6. O YES BABY  
7. I DIG  
8. BIG DEAL  
9. COOL BABY  
A. READY BABY  
B. GOOD LORD  
C. BOSS SASSY  
D. 3.141592654  
E. ☐ ☐ ☐ ☐ ☐

## Secondary

0. YES BABY  
1. LA DE DA  
2. I A LADY  
3. GOOD BABY  
4. GOOD DEAL  
5. IS SO GOOD  
6. IDA GOOD  
7. BALLS BOSS  
8. GOLLY GEE  
9. DO SAY

.ADD	.788
.SUB	.117
.MULT	.5141
.DIV	.892
.SIN	.195
.COS	.751
.TAN	.175
.ARCSIN	.767195
.ARCCOS	.767751
.ARCTAN	.767175
.LOG	.459
.ANTLOG	.751459
.RCP	.676

# Elia data card

## Primary

0. SORRY DEAR  
1. DEAR ELLA  
2. O DEAR  
3. DO CEASE  
4. DO BE GOOD  
5. BYE DEAR  
6. O REALLY  
7. O YES YES  
8. YES DEAR  
9. SO GLAD  
A. READY DEAR  
B. O DEAR GOD  
C. GLORY BE  
D. 60.00  
E. ☐ ☐ ☐ ☐ ☐  
I. DEAR ELLA

## Secondary

0. YES O YES  
1. O ALAS  
2. ELLA CRY  
3. DEAR BOSS  
4. BLESS GOD  
5. I DO AGREE  
6. O BOSS  
7. I DISAGREE  
8. I OBEY SIR  
9. O I SAY

Zerox copy of  
overlay you need  
for "Gerald & Soda"  
program. Cut out  
& place over number  
keys.

