* TI PPC NOTES * V5N2, 1980.

NEWSLETTER OF THE TI PERSONAL PROGRAMMABLE CALCULATOR CLUB.
9213 Lanham Severn Road, LANHAM MD, 20801.

At the request of several members who pointed out the advantage, we are calling this issue V5N2, rather than V1N2. The year 1979 was for 52-Notes V4. So, for continuity's sake, we are calling our year 1980 issues V5. Thus, in quoting passages in future issues we don't have to distinguish between 52-Notes and TI PPC NOTES. Please rename issue 1 of TI PCC NOTES accordingly: V5N1.

The greatest number of complaints was about the lack of suitable margins. So, to satisfy the hole punchers, the large majority of the members, this issue has at least one inch (2,5 cm to my European friends) margin on both sides. What to do about issue 1 ? You might copy the 12 pages on the office copier and file those pages. The original is going to become a collector's item anyway!

The Post Office was at its best again, playing games. I took a sample envelope of issue one and presented it to three different clerks in three separate offices. The unanimous answer was: The most advantageous mailing is 3d Class, Printed Matter, $ 0.20. I took the first 75 envelopes and gave them to a clerk, who assured me again that everything was peaches and cream. Exactly six days later my home mail box was overstuffed with 75 large envelopes. An accompanying form letter said that "oversize" mail gets fined by $ 0.07. "Oversize" is defined as"larger than 6.5 by 11 inches in 3d class mail"! After licking 75 seven cents stamps, I was informed that the same envelopes could be mailed "first class" at $ 0.28. I managed to keep my blood pressure within prescribed and safe limits.

Because we are going to make an honest effort to dissiminate good programming practices, rather than just simply distribute programs per se, we would like to establish some minimum standards of getting a program published. I don't care how you list your program, as long as I can read it. I will re-list it anyway, to satisfy the neatness impulses of the printer. But each program should be accompanied by a reasonable explanation as to HOW IT WORKS. If you want to write some prose about it, as George Vogel does in the Newcomers Corner, I like it even better. And so will all the members. In one sentence, make it as easy as possible to others to understand what is going on. We thank you for it. I realize that it is not always possible to enforce this rule at all times.Sometimes I will rush a neat program into print and then "badger" the author for more info, which will be printed later, together with comments received. A case in point is the program ALPHA SORT in this issue. Yes, you saw correctly, it alphabethically sorts print code in registers.(You can fool it, if you know how: use extended print code, i.e. print code with digits above 7. The thing goes haywire!) But it was such a nice program I couldn't resist publishing it without documentation, which, I am sure, we will get later.(Sorry Robert. I couldn't resist that either!)

One other highlight in this issue is our TELECOMMUNICATIONS EXPERIMENT. I had to wait for this issue to publish it, to give our patent lawyer time to finish his applications. Now he has given us the go-ahead. I hope I can count on your cooperation to make this scientific endeavor a resounding success. Again we thank you!

One member, among a lot of other recommendations, writes something to the extent that, if the HP PPC JOURNAL is able to make it on an equivalent contributions basis, I'd better too! I'll try my utmost best, rest assured! But bear in mind that 1) the HP club has been around for many years and is well known by now. Ergo, they can dispense with spending a large percentage of their income on making propaganda for itself. So, please tell your friends about the club. Word of mouth is very effective and doesn't cost that much.
2) The HP club has a few more members than we have. It is a well known fact from the business world that per capita costs decrease when the number of costumers increases. So, please tell your friends........

Some of you have paid more than their dues, "to defray start-up costs," as one member puts it. I thank those thoughtful ones for their generosity.

Maurice E.T. Swinnen

TELECOMMUNICATIONS EXPERIMENT.- During last summer we, at the Washington DC Area club, had several meetings at an annex of the Walter Reed Army Institute of Research in Forest Glen, Maryland. The area is dotted with commercial TV and radio stations. One of the most powerful stations in the Washington area, WWDC, is situated about 500 yards from from our meeting place. Very often we noticed strange interference, either while executing a program or while listing one. The printer would suddenly produce gibberish. Bill Skillman, who is a professional radar engineer with the Westinghouse corporation in nearby Baltimore finally researched the phenomenon a little closer and by means of some field strength measurements and loop antennas pinpointed the source of interference as WWDC. He and Panos Galidas convinced the transmitter technician on call that Saturday afternoon to interrupt transmissions for a few seconds. By a brilliant deduction method it was found that the so-called interlace signals, signals transmitted in between TV frames, were the ones producing the gibberish on our PC100's.

We also reasoned that, if these signals can travel 500 yards, what impedes them to travel even further? A call to an absent member, Burton Anrews, who was painting his house in nearby Potomac, confirmed our suspicision. Burt received our signals as clear as if his PC100 had been sitting in our usual meeting place.

Next we convinced the transmitter technician to put a few patterns of our own design on his carrier frequency, in between the interlace signals. It didn't take us long to come up with a recognizable code. In fact, it proved to be a rather efficient way of communicating.

We also found that the signal strength could be enormously increased if a radio receiver or TV set was brought close to the PC100. It even increased the signals more if the receiver was tuned to certain frequencies.

As with all radio transmissions, the higher the transmitting power, the farther away your signals will be received. Thus, I convinced my boss to let us use the enormous power supply used for our gigantic 2.45 Gigahertz pulsed micro-wave transmitter. It is a rather huge affair, this oil-filled transmitter and capacitor multiplier, too large to transport 500 yards away to the site of WWDC. But the power stored in the large 2500 μfarad capacitor can be conducted by means of a cable wherever you wish. This job was right up Frank Blachly's alley. He is a civil engineer and he does these assignments professionally every day of his life. The capacitor can theoretically charge to 8000 volts, but we limit this to about 750 volts for operator safety. Even at that modest voltage we arrive at a rather respectable power. Taking in account the Beta or multiplication factor of our capacitor-multiplier of 10,000 we have $E = CV^2 / 2 = 25 X 750^2 /2 = 7.03123$ Megajoules. At this power output we would still have a signal strength of -32 dBM on Mars! Obviously, we can easily illuminate any point on earth with sufficient signal strength.

As the system is obviously patentable, we would like to do first some long distance reception experiments. And here is were we would like to ask for your cooperation. Enclosed find a recorded mag card. For reason of patent disclosure, we had to protect the contents by recording it with -1. We ask your indulgence for this. On the 31 of March, at 2300 hours your local time (we will repeat the pulsed transmissions on the hour for 5 consecutive periods) insert the card in the TI-59 mounted on a PC100. (A, B, C or D) Side 1 only is recorded. Next, mark the exact time you press A to start. That is all. You can go to bed now. Some time during the night, if all goes well,signals will be received and the result will be printed on the PC100.You can increase signal strength considerably, and thereby the chance of good reception, by putting a TV set or radio receiver close by, preferrably with the PC100-TI-59 inside a wire loop, formed by knotting the two ends of exactly 3.14 meters of any type of lamp card, TV lead-in or just simply bare wire. No need to turn the receiver on, just tune it to any of the frequencies of WWV, i.e. 5, 10, 15, 20 or 25 Mhz. TV sets can be tuned to either channel 4, 8, 16, 32 or 64.

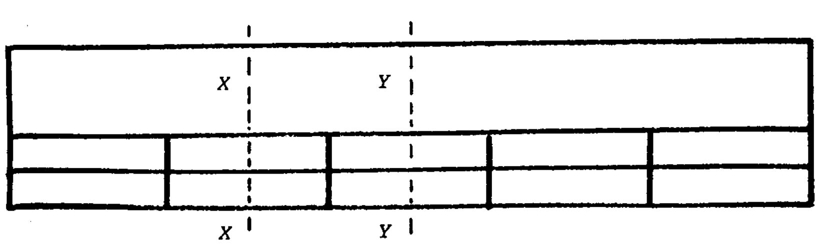
After successful reception of the signals, the mag card in turn will now contain in-

formation valuable to our club. Please keep the card away from magnets, small motors and other magnetized objects. We will collect the mag cards at a later date. We will send you special mu-metal-lined envelopes, to protect the card from magnetism in the sorting machines of the post office.
We would also like to ask you not to attempt a premature execution of the program, "just to have a peek what this guys are up to." It would disturb a timing loop recorded on the card. That timing loop should ideally only start when you press A. So, please mark the exact time you press A on the card itself. That will allow us later to compare that time to our own transmitting periods and the counted down time on the mag card. From it we will be able to deduce a lot of charateristics of our novel transmission system.
We would be very grateful to you if you would be so kind as to assist us in this scientific endeavor. The Washington DC Area TI PPC club thanks you in advance.

HARDWARE.- Peter Poloczek tells in TI-Software, a German-language newsletter, how he has successfully adjusted the reading speed of mag cards on the TI-59. This way it becomes possible to read a mag card recorded on an incompatible TI-59. His recipe is as follows:

1. *Take out the module and the battery pack.*
2. *Take out the two screws in the back and separate the two plastic halves of the calculator. To do this, slide the top half down with respect to the bottom half. There are two plastic tabs holding the two halves together, just below the key board.*
3. *On the right side of the bottom half, on the side opposite the battery charger connector, there is a small, about 1/4 inch diameter, potentiometer, near the edge of the printed circuit board. It is sitting on its side. Inserting a small screwdriver in its slot, on your left hand, and turning the potentiometer clockwise will increase the reading speed, and, of coarse, vice versa.*
   But that adjustment should be done when the calculator is reassembled. So,
4. *drill a small hole in the top half of the calculator case, such that later, after reassembly, you will be able to insert a small screwdriver in it and adjust the reading speed. Unfortunately, the hole will be situated on the left side of the calculator, such that it will be difficult to adjust the speed when the calculator is mounted on the printer. The printer housing will be in the way. Thus, either adjust the speed with a short screwdriver, of maximum 1 3/4 inch length or adjust the speed with the calculator not mounted on the cradle-printer.*
5. *Reassemble the calculator housing.*
6. *When you are faced with a problem card, tweek the potentiometer just a trifle. That is, turn it not more than a few degrees at a time. Try both directions and watch the results.*
   Peter also says that new calculators come adjusted, such that, after reading, the mag card always stops between X and Y. See drawing below.

   I modified one of the two calculators I own and was able to finally make it compatible with the other one. It now also reads all the cards from friends it formerly rejected. Peter claims that a cure is possible in 80 % of the cases.



NEAREST STANDARD VALUE ROUTINES.- *In programming it often occurs that you need the nearest standard value to an entered one, either lower or higher. For example in electronics, resistors come in standard values of 20 %, 10 %, 5 %, 2 %, 1 % and if you have the money even smaller percentages. This means that, by allowing a manufacturing tolerance of ± 10 % one can cover a complete decade with only 12 different values. Thus, if you entered, for example, an odd value of 553 ohms, your routine should produce either the next higher value of 560 ohms or the next lower value of 470 ohms. This problem is not only limited to electronics, Many other professions have tabulated standard values which are used routinely.*

In the case of standard resistor values, as we said, one decade is covered by 12 values in the 10 % series. As there are close to eight full decades in use, that would mean 96 values to be stored. Luckily, if one stored only the basic values between 1 and 10, the next decade values could be obtained by multiplying by 10, and so on. Information as to the exact multiplier to be used can be obtained by the well-known trick of taking the log (base 10) of the odd entered number, taking the integer value of it and then the anti-log value. The process is rather slow, and slower if the list is longer. But its redeeming factor is that it is rather economic in program steps.

To demonstrate the technique I have printed on the next page two routines I and II. In both routines LBL E stores the standard values in registers 1 through N. This can of course also be done by means of an extra mag card, but this scheme saves you the trouble. Just press E to initialize. Routine I returns with the next higher value if you enter an odd value and press A.

Routine II gives you two possiblities. Press A for the next higher value or press B for the next lower one. The values stored here are for the 10 % standard resistors. The "10" at the end of the series, which at first glance seems to be redundant, is necessary if the entered value is 8.2, such that the program can choose 10 as the next higher value. The "13" in program steps 017-018 of I or 021-022 of II corresponds to the highest register called in the standard list. If you want to make your list longer, change that number accordingly.

Working registers have been chosen as 54 and 55 here, to allow the data registers list to grow without having to reassign the working register numbers.

One could, of course, choose a low number for both working registers. But that would require a lot of programming gymnastics: you would have to subtract a fixed number each time from your DSZ register, prior to doing the DSZ test. Subsequently you would have to add that same number again. The fixed number I am talking about is equal to the lowest register in the standard values list. (not its contents, but the #) Remember to synthesize DSZ 55 by DSZ STO 55 BST BST DEL SST or by DSZ DIV

HIDDEN DIGITS VIEWER.- *It is well known that any number in the display might still have an 11th, a 12th and a 13th "hidden" or "guard" digit in the so-called x-register. Several schemes have been divised to make those digits visible. With uncomplicated numbers, such as π it is rather simple to make the hidden digits visible. Enter π and press* X 1000 = INV INT *to see 0.59265359*

Thus it was shown that the last three digits (359) were rounded up in the display of π to 4 (3.141592654)

But try to make this simple routine work with, for example π X 1 EE 22 = and it will not work. 52-Notes, V3N10P4 published such a routine that would work with any number. Richard Blayney has shortened that routine by 9 steps. Put it in user memory, enter your number in the display and press A. You might add a PRT command if you want to see the result printed.

LBL A X (EE DIV EE 0 0) FIX 0 EE +/- X 1 EE 3) INV FIX INV INT INV EE RTN

What Dick is doing to make it work with any number can be generalized in this visual routine as X 1 EE n = INV INT INV EE

where n is equal to -Exponent + 3

For example, if the exponent is 36, then n is equal to -36 + 3 = -33

if the exponent is -41, then n is equal to -(-41) + 3 = 44

NEAREST STANDARD VALUES ROUTINES.

Instructions: Initialize, press E.

Enter odd value and press A. Program returns with next higher standard value

Enter odd value and press B. Program returns with next lower standard value.

For values lower than 1 display will flash 99999.99 99

ROUTINE II

```
000  76 LBL      031  22 INV
001  12  B       032  77  GE
002  86 STF      033  24 CE
003  00  00      034  97 DSZ
004  76 LBL      035  55  55
005  11  A       036  22 INV
006  29 CP       037  00  0
007  85  +       038  35 1/X
008  32 X:T      039  91 R/S
009  95  =       040  76 LBL
010  28 LOG      041  24 CE
011  59 INT      042  87 IFF
012  22 INV      043  00  00
013  28 LOG      044  33 X²
014  42 STO      045  01  1
015  54  54      046  44 SUM
016  35 1/X      047  55  55
017  65  ×       048  76 LBL
018  32 X:T      049  33 X²
019  95  =       050  73 RC*
020  32 X:T      051  55  55
021  01  1       052  65  ×
022  03  3       053  43 RCL
023  42 STO      054  54  54
024  55  55      055  95  =
025  76 LBL      056  22 INV
026  22 INV      057  86 STF
027  73 RC*      058  00  00
028  55  55      059  92 RTN
029  67  EQ      060  76 LBL
030  33 X²       061  15  E
```

Etc. etc., identical to LBL E of routine I.

ROUTINE I

```
000  76 LBL      035  91 R/S      075  02  2
001  11  A       036  76 LBL      076  93  .
002  29 CP       037  24 CE       077  07  7
003  85  +       038  01  1       078  42 STO
004  32 X:T      039  44 SUM      079  06  06
005  95  =       040  55  55      080  03  3
006  28 LOG      041  76 LBL      081  93  .
007  59 INT      042  33 X²       082  03  3
008  22 INV      043  73 RC*      083  42 STO
009  28 LOG      044  55  55      084  07  07
010  42 STO      045  65  ×       085  03  3
011  54  54      046  43 RCL      086  93  .
012  35 1/X      047  54  54      087  09  9
013  65  ×       048  95  =       088  42 STO
014  32 X:T      049  92 RTN      089  08  08
015  95  =       050  76 LBL      090  04  4
016  32 X:T      051  15  E       091  93  .
017  01  1       052  01  1       092  07  7
018  03  3       053  42 STO      093  42 STO
019  42 STO      054  01  01      094  09  09
020  55  55      055  01  1       095  05  5
021  76 LBL      056  93  .       096  93  .
022  22 INV      057  02  2       097  06  6
023  73 RC*      058  42 STO      098  42 STO
024  55  55      059  02  02      099  10  10
025  67  EQ      060  01  1       100  06  6
026  33 X²       061  93  .       101  93  .
027  22 INV      062  05  5       102  08  8
028  77  GE      063  42 STO      103  42 STO
029  24 CE       064  03  03      104  11  11
030  97 DSZ      065  01  1       105  08  8
031  55  55      066  93  .       106  93  .
032  22 INV      067  08  8       107  02  2
033  00  0       068  42 STO      108  42 STO
034  35 1/X      069  04  04      109  12  12
                 070  02  2       110  01  1
                 071  93  .       111  00  0
                 072  02  2       112  42 STO
                 073  42 STO      113  13  13
                 074  05  05      114  91 R/S
```

```
000  76 LBL     038  59 INT     076  77  GE     114  42 STO     152  U1   1
001  11   A     039  82 HIR     077  00  00     115  00  00     153  85   +
002  93   .     040  07  07     078  47  47     116  69  OP     154  42 STO
003  42 STO     041  82 HIR     079  82 HIR     117  30  30     155  00  00
004  00  00     042  55  55     080  15  15     118  73 RC*     156  32 X:T
005  69  OP     043  01   1     081  32 X:T     119  00  00     157  73 RC*
006  20  20     044  85   +     082  01   1     120  75   -     158  00  00
007  25 CLR     045  42 STO     083  85   +     121  52  EE     159  69  OP
008  43 RCL     046  00  00     084  22 INV     122  72 ST*     160  01  01
009  00  00     047  73 RC*     085  77  GE     123  00  00     161  82 HIR
010  82 HIR     048  00  00     086  00  00     124  95   =     162  18  18
011  08  08     049  32 X:T     087  45  45     125  52  EE     163  67  EQ
012  92 RTN     050  82 HIR     088  25 CLR     126  00   0     164  01  01
013  55   ÷     051  17  17     089  82 HIR     127  00   0     165  47  47
014  28 LOG     052  44 SUM     090  18  18     128  22 INV     166  77  GE
015  69  OP     053  00  00     091  82 HIR     129  28 LOG     167  01  01
016  03  03     054  73 RC*     092  05  05     130  52  EE     168  44  44
017  22 INV     055  00  00     093  82 HIR     131  64 PD*     169  81 RST
018  28 LOG     056  77  GE     094  17  17     132  00  00     170  76 LBL
019  52  EE     057  00  00     095  32 X:T     133  25 CLR     171  12   B
020  85   +     058  79  79     096  01   1     134  97 DSZ     172  85   +
021  82 HIR     059  32 X:T     097  03   3     135  00  00     173  01   1
022  17  17     060  72 ST*     098  22 INV     136  01  01     174  95   =
023  95   =     061  00  00     099  77  GE     137  18  18     175  32 X:T
024  72 ST*     062  82 HIR     100  00  00     138  02   2     176  11   A
025  00  00     063  17  17     101  39  39     139  82 HIR     177  73 RC*
026  61 GTO     064  94 +/-     102  04   4     140  68  68     178  00  00
027  00  00     065  44 SUM     103  22 INV     141  61 GTO     179  71 SBR
028  05  05     066  00  00     104  77  GE     142  01  01     180  00  00
029  76 LBL     067  32 X:T     105  00  00     143  52  52     181  13  13
030  13   C     068  72 ST*     106  39  39     144  44 SUM     182  77  GE
031  82 HIR     069  00  00     107  01   1     145  00  00     183  13   C
032  18  18     070  01   1     108  22 INV     146  73 RC*     184  61 GTO
033  82 HIR     071  32 X:T     109  67  EQ     147  00  00     185  01  01
034  05  05     072  44 SUM     110  00  00     148  69  OP     186  77  77
035  55   ÷     073  00  00     111  39  39     149  03  03
036  02   2     074  43 RCL     112  82 HIR     150  69  OP
037  54   )     075  00  00     113  18  18     151  05  05
```

ALPHABETHICAL SORT.

1. *Initialize, press A.*
2. *Enter print code, R/S.*
   Enter print code, R/S, etc.
3. *Sort, press C.*
4. *For print code already in registers, press B to sort.*

Author:

Robert Snow

```
APE       MA
DOG       ME
 HI     MINE
LOW       MY
        ZULU
```

SORT print-out

NUMERIC	ALPHA	REG
133317	APE	02
163222	DOG	03
2324	HI	04
273243	LOW	05
3013	MA	06
3017	ME	07
30243117	MINE	08
3045	MY	09
46412741	ZULU	10

Results in registers.

DIRECT ADDRESSES THE PAINLESS WAY.- My good friend Karl-Joseph Meusch in Frechen, Germany, recently sent me the fifth one in a series of books about programmable calculators now appearing in that country. This one had the promising title of PROGRAMMOPTIMIERUNG FUER TASCHENRECHNER (AOS)=Program Optimization for Pocket Calculators Using the AOS System. The author, Hans-Joachim Ludwig, has done a splendid job. With proverbial thoroughness usually ascribed to his country men he goes methodically through all the possible commands and its permutations on the SR-56, the SR-52, the TI-57 and the TI58/59. I especially enjoyed his thoughts on algorithms and flow diagrams. But then a curious thing happened: If you read a book about optimization of programs you expect a rather large chapter on direct addresses. Surprise! Only a couple of lines about them in a chapter on how to keep a program flexible. The author recommends to add copiously several series of NOPs at strategic places in your program! So, that is why I always find those unsightly left-over NOPs in programs. I even see them sometimes in label programs! Really now!

Why this uneasiness about direct addresses? It is such a great device to speed up a rather slow-executing common-labels-program. You don't have to be a genius to put them in. (An I.Q. of about 170 suffices)

Let's try one simple, fool proof method. But, whatever method you use, remember one ground rule: YOUR PROGRAM HAS TO RUN CORRECTLY WITH THE COMMON LABELS before you try to put direct addresses in. Don't try to alter your program once it is converted to direct addresses. That is just asking for trouble. It is always wise to keep a copy of the original common-labels-program handy, just in case somebody detects a bug in your program. It is much easier to modify that one and go through the labels-to-addresses conversion once again than to change all the addresses in a program. One mistake and goodbye program. You'll never find the bug!

The method presented here could be called "the tabulating routine." Buy yourself in an office supply store some columnar pads. The ones by Wilson Jones, #'s 7212, 7213 and 7214 are tops. Also the AMPAD, Efficiency Line # 6636 are very good.The pads must have at least 10 columns of 3/4 inch wide (2 cm to my European friends) and also have a left and a right hand margin.

Mark the headings of the columns as 0, 1, 2....9. Then mark the left hand margin with the program steps 000, 010, 020....going up by ten.

Now, fill in each step from your common-labels program. See example. When you arrive at a label you want to remove, just omit that label, but write it and the starting address in the top margin, such as, for example LBL STO = 136. When you arrive at a branch, either a conditional or an unconditional one, omit the branch label name but leave open two program steps instead. FRAME THOSE TWO STEPS. Then write in the right hand margin the name of the label you omitted. Later, once you know the address where that label starts you can fill in the framed two steps. Keep on writing down your entire program this way. Be careful with numbers: each digit is a step. On the other hand, register numbers are written as two digits per step. That is all there is to it!

Here follows a short common-labels program from which the labels have been replaced by direct addresses in the example on the next page.

000	76	LBL	011	91	R/S	021	35	1/X	032	43	RCL
001	11	A	012	76	LBL	022	61	GTO	033	76	LBL
002	42	STO	013	42	STO	023	42	STO	034	44	SUM
003	01	01	014	61	GTO	024	76	LBL	035	58	FIX
004	87	IFF	015	44	SUM	025	43	RCL	036	40	IND
005	05	05	016	76	LBL	026	42	STO	037	25	25
006	42	STO	017	13	C	027	13	13	038	91	R/S
007	43	RCL	018	43	RCL	028	97	DSZ	039	76	LBL
008	12	12	019	12	12	029	13	13	040	35	1/X
009	67	EQ	020	77	GE	030	44	SUM	041	91	R/S
010	43	RCL				031	61	GTO			

LBL STO = 014 , LBL RCL = 027 , LBL SUM = 036 , LBL 1/X = 039

	0	1	2	3	4	5	6	7	8	9		
000	LBL	A	STO	01	IFF	5	0	14	RCL	12	STO	
010	EQ	0	27	R/S	GTO	0	36	LBL	C	RCL	RCL	SUM
020	12	GE	0	39	GTO	0	14	STO	13	DSZ	1/X	STO
030	13	0	36	GTO	0	27	FIX	IND	25	R/S	SUM	RCL

Note that LBL 1/X would have meant in reality one more R/S past the last R/S at step o39. But if we refer the GE branch on step 021 to 039 as a branch address, it will stop there alright.

Note also that the addresses in the framed two-step blocks have been put there AFTER the entire program was written. Then, and only then, do we know the addresses of all the labels we removed.

The small numbers in the left-hand margin and in the column headings are the original numbers printed on the columnar pad. Just ignore them.

The pads have rows alternately shaded and unshaded. The light blue shading did not reproduce on the copy machine. It is wise to use only alternate rows. That leaves you a whole row in between to use in case of mistakes.

DIAGNOSTIC.- *Recently I wrote a diagnostic program and put it in our local club's newsletter. It received a lot of commentary, mostly from people saying it was a little bothersome to produce it on a mag card. But two members, Frank Blachly and Dick Blayney offered some positive suggestions how to improve the entry. So, the following is built upon those inputs:*
The idea is to put an SST, code 41, on each step from 000 through 959. Then if you press R/S SST, the program counter will move sequentially through each step, and finding 41 there will continue, until it arrives at step 959, where it halts with a zero flashing in the display. To check if in effect it has arrived at 959 you press LRN and you expect to see 959 41. If any other step contains any other code, the program counter will stop there, indicating a bad data register.
Now, in order to put a code 41 at each program step, you could write in LRN mode: STO 41 BST BST DEL SST STO 41, etc. But that will be kind of boring, because you will have to fill up at least one bank, meaning that you would have to do the above routine 240 times. So, I find Dick Blayney's routine here the easiest:
At Turn-on, key: LRN STO IND 00 DSZ 0 000 5 OP 17 R/S LRN
Now, back in calculator mode, press the following:
RST 93 STO 00 OP 17 41 41 41 41 EE 7 + 41414 EE 2 = R/S
The program is running now. Allow about 30 sec for execution. Then press GTO 240 LRN . *Now delete each"40"step you find. You will find it at step 240 and at each subsequent 8th step, 30 steps in total. Go beyond step 480.*
You now have a full bank, bank 2, with nothing but 41 steps.
Press CLR OP 17 *and see 959 in the display. Record bank 2 on any side of a mag card by pressing* 2 2ND WRITE *and inserting the card side.*
Mark the card with : DIAGNOSTIC CLR OP 17.
If you want to check the program memory on any calculator, you simply force the recorded card side into all 4 banks: (use partition 959.00 by pressing CLR OP 17)
Enter 1 +/- and insert card side.
Enter 2 +/- and insert card side.
Enter 3 +/- and insert card side.
Enter 4 +/- and insert card side.
Then press RST SST (once only) and wait a few seconds until display flashes a zero.
Then press LRN to see if program counter has indeed arrived at step 959. Your calculator is OK.
The main advantage of this method is, that you don't have to step SST 960 times or, as an alternative, waste a mile of printing paper.

Then, working in the same vain, Dick offers a diagnostic for 52 users. It puts a 71 step in each location.
At turn on, key: LRN IND STO 00 DSZ 00 0 LRN
And back in calculator mode, press RSET 97 STO 00 71 71 71 71 EE 10 + 71717 EE 5 = STO 01 DIV 0 = *(disregard flashing)* RCL 01 RUN
Wait for flashing display and press CLR
Record the card in the usual manner.
In order to use the program, to check any SR-52, :
Read-in both sides of the mag card and press RSET SST *(once only.)*

NICHOMACHUS' PUZZLE.- *I recently submitted to PPX a program by that name. It concerns a mathematical game in which you are asked to supply the REMAINDER of a number between 1 and 100 you supply,divided by 3, then by 5 and finally by 7. From these clues the program computes the original number you supplied but did not reveal. The algorithm is about 2000 years old, devised by a mathematician from Arabia Petreae:*
[N] mod 3 → A , [N] mod 5 → B, [N] mod 7 → C.
*70*A + 21*B + 15*C → D, N = [D] mod 105*
In which the square brackets denote "the positive integer of..."
As can be seen, you multiply the remainder of the first division by 70, the second remainder by 21 and the third one by 15, then sum all the products and divide as many times by 105 until you have a number lower than 105. That is the original number supplied.
To which Richard Snow said "the repetetive subtraction of 105 is one method of determining a remainder. Another method is – (CE DIV 105) INT X 105 = "
I cannot publish a copy of Nichomachus' puzzle here, but you might try your own version.

TI-58/59, ML MODULE
PC100 OPTIONAL

MAKE UP YOUR MIND.

This game can be played on both the TI-59 and the TI-58, with or without the PC-100. The calculator produces a divisor between 2 and 10. Next it produces 15 numbers between 1 and 100 and displays each of them for about 1.5 sec. It is up to you to decide if the number is divisible by the divisor. But do it quickly. If you decide YES and press R/S and if you are right, the number is added to your total points. If you are wrong, the punishment is the subtraction of the number from your total accumulated points. If you just let it slip by, only one point is subtracted from your total.

Instructions:
Enter a seed between 1 and 10^5 and press A.
With the printer, the divisor is printed, otherwise it is displayed only for about 3 sec.
Next a number between 1 and 100 is displayed for about 1.5 sec. If you decide it is divisible by the divisor, press R/S quickly. (HOLD DOWN)
Then press E to continue the game.
With the printer, your accumulated total so far is printed. Otherwise, the calculator stops to display the total. Press R/S to continue.
After 15 trials the calculator displays and the printer prints .11111111 , followed by your final total points.
For a new game, enter a new seed and press A.

Once you have become an expert at this game, you might make it more challenging by deleting one, or even two PAU commands at steps 017 through 019 . This will reduce the allowable reaction time from 1.5 to 1.0 resp. to .5 sec.
Another way of increasing the difficulty factor is to place a larger number at steps 060 through 062 and/or at steps 078-079. (the divisor)

The idea for this program comes from Karl-Joseph Meusch, in Frechen, Germany.

```
000 69 OP     040 98 ADV    080 42 STO    120 22 INV
001 38  38    041 91 R/S    081 11  11    121 44 SUM
002 43 RCL    042 76 LBL    082 36 PGM    122 08  08
003 08  08    043 11  A     083 15  15    123 43 RCL
004 99 PRT    044 42 STO    084 13  C     124 08  08
005 76 LBL    045 09  09    085 58 FIX    125 99 PRT
006 14  D     046 02  2     086 00  00    126 87 IFF
007 36 PGM    047 00  0     087 52 EE     127 07  07
008 15  15    048 69 OP     088 22 INV    128 14  D
009 13  C     049 07  07    089 52 EE     129 91 R/S
010 58 FIX    050 69 OP     090 42 STO    130 14  D
011 00  00    051 19  19    091 14  14    131 76 LBL
012 52 EE     052 25 CLR    092 66 PAU    132 10 E'
013 22 INV    053 36 PGM    093 66 PAU    133 43 RCL
014 52 EE     054 15  15    094 66 PAU    134 12  12
015 42 STO    055 10 E'     095 66 PAU    135 44 SUM
016 12  12    056 12  B     096 66 PAU    136 08  08
017 66 PAU    057 01  1     097 66 PAU    137 43 RCL
018 66 PAU    058 42 STO    098 99 PRT    138 08  08
019 66 PAU    059 10  10    099 98 ADV    139 99 PRT
020 25 CLR    060 01  1     100 92 RTN    140 87 IFF
021 97 DSZ    061 00  0     101 76 LBL    141 07  07
022 00  00    062 00  0     102 15  E     142 14  D
023 00  00    063 42 STO    103 43 RCL    143 91 R/S
024 00  00    064 11  11    104 12  12    144 14  D
025 25 CLR    065 25 CLR    105 55  ÷
026 42 STO    066 42 STO    106 43 RCL
027 12  12    067 08  08    107 14  14
028 09  9     068 01  1     108 95  =
029 35 1/X    069 05  5     109 22 INV
030 58 FIX    070 42 STO    110 59 INT
031 09  09    071 00  00    111 42 STO
032 66 PAU    072 14  D     112 15  15
033 66 PAU    073 76 LBL    113 29 CP
034 99 PRT    074 12  B     114 43 RCL
035 58 FIX    075 02  2     115 15  15
036 00  00    076 42 STO    116 67  EQ
037 43 RCL    077 10  10    117 10 E'
038 08  08    078 01  1     118 43 RCL
039 99 PRT    079 00  0     119 12  12
```

NOTE: ML module required.

LABELS.

```
006 14 D
043 11 A
074 12 B
102 15 E
132 10 E'
```

NEWCOMER'S CORNER.- This time we would like to give you a practical example of the use of algorithms. Algorithms are simply sets of instructions we have to follow in order to achieve the successful completion of a certain computation we want to do in order to solve a problem. Here then are George Vogel's views on good and bad algorithms.

EFFICIENT AND INEFFICIENT ALGORITHMS by GEORGE VOGEL.- As with everything else in this world, there are good algorithms and there are bad ones. The difference becomes especially important where a loop is passed through repeatedly. The following offers a good illustration: The greatest common divisor (GCD) of two numbers a and b (a ≥ b) is found by the Euclidian algorithm, which is best explained via an example, say, for a = 10 and b = 4.

1) *b is subtracted from a until less than b remains:*
```
 10
- 4 = 6
- 2 = 2 (remainder)
```
2) *if remainder is not zero, a is replaced by b and b by the remainder; steps 1) and 2) are repeated until remainder is zero:*
```
 4
-2 = 2
-2 = 0
```
3) *when a zero remainder is obtained, the last number subtracted (here 2) is the GCD of a and b.*

This has been translated directly into routine B below (except that in practice it is simpler to continue subtraction until the remainder is zero or negative, and if negative, b is re-added once) Initialize by pressing B. Then enter 4 and 10 each followed by R/S. The order is immaterial. In about 3 seconds you should get 2 as the GCD and 20 as the Least Common Multiple. (LCM)

Now try 6 and 1005 in the same manner. You will correctly get 3 and 2010, but the calculation will take far longer. (about 1.5 min!) This is largely because 6 had to be subtracted from 1005 a total of 168 times (and re-added once), which takes time:
```
  1005                                    Then:   6
 -1008 = -3 (6 subtracted 168 times)             -3 = 3
 +   6 =  3 (6 re-added)                         -3 = 0
```
Therefore GCD = 3, LCM (= ab/GCD) = 2010

Evidently the problem lies in the large number of subtractions involved, and will of course get worse in proportion to the total number of subtractions in a given calculation. Now there is an easy solution, though is was not discovered by the authors of, for example, the GCD/LCM program for the SR-56 or the "Addition of Fractions" program in the November 1979 issue of PPX exchange: If one can determine directly how many times b would have to be subtracted (say, n times) to get a remainder < b, without actually doing it, one could simply calculate the remainder as (a - nb), applying this approach in each round instead of the slavisgly repeated subtractions. Each round would then involve a single pass through the short routine, regardless of the number of passes through the corresponding routine necessary in routine B. n is simply the integer part of a/b. In the case of 1005 and 6, n = [1005/6] = [167.5] = 167. (Square brackets always denote "integer part of" (editor)) Therefore, 1005 - 167·6 = 3 is the remainder in the first round (since it is of necessity smaller than b but not negative, the only test required is, whether is is zero; here it is not, so we continue.) 6 - [6/3]·3 = 0; the remainder in the second round is zero, and GCD = 3.

This is the basis for routine A. Try it on the two problems. This time initialize with A. Each problem should now take only about 2 seconds.

Routine A clearly has the greatest edge on routine B when it eliminates long rounds of subtractions in routine B, i.e. if at one or more points a number must be subtracted from a larger one. This situation clearly has no effect whatever on routine A, which will take a time proportional only to the number of rounds necessary. It is interesting to define the "worst scenario" for routine A: it is when a and b are consecutive Fibonacci numbers. Then each round will lead to the next lower Fibonacci number, until finally 1 - 1 = 0, so that any pair composed of an n-th and the (n+1)-th Fibonacci numbers will require n rounds. In this (admittedly rather unlikely) case the advantage of routine A over routine B will be the smallest, but A will still be

(over)

Algorithms..(continued)
faster since it will need only one pass per round vs. two in routine B. Try it on the 19th and 20th Fibonacci numbers (4181 and 6765) which should take ca. 16 and 22 seconds with the two routines.
Thus, if your program takes too long, look for unnecessary loops!

000	76	LBL	024	95	=	048	00	0	072	63	63
001	11	A	025	67	EQ	049	00	0	073	85	+
002	29	CP	026	00	00	050	76	LBL	074	43	RCL
003	91	R/S	027	33	33	051	12	B	075	01	01
004	99	PRT	028	48	EXC	052	29	CP	076	95	=
005	42	STO	029	01	01	053	91	R/S	077	48	EXC
006	01	01	030	61	GTO	054	99	PRT	078	01	01
007	42	STO	031	00	00	055	42	STO	079	61	GTO
008	02	02	032	13	13	056	01	01	080	00	00
009	91	R/S	033	43	RCL	057	42	STO	081	63	63
010	99	PRT	034	01	01	058	02	02	082	43	RCL
011	49	PRD	035	99	PRT	059	91	R/S	083	01	01
012	02	02	036	35	1/X	060	99	PRT	084	99	PRT
013	75	-	037	65	×	061	49	PRD	085	35	1/X
014	53	(	038	43	RCL	062	02	02	086	65	×
015	24	CE	039	02	02	063	75	-	087	43	RCL
016	55	÷	040	95	=	064	43	RCL	088	02	02
017	43	RCL	041	99	PRT	065	01	01	089	95	=
018	01	01	042	98	ADV	066	95	=	090	99	PRT
019	54	)	043	61	GTO	067	67	EQ	091	98	ADV
020	59	INT	044	00	00	068	00	00	092	61	GTO
021	65	×	045	03	03	069	82	82	093	00	00
022	43	RCL	046	00	0	070	77	GE	094	53	53
023	01	01	047	00	0	071	00	00			

It is almost superfluous to say that these routines can also be executed on the TI-58. If you don't use a printer, replace steps 035 and 084 by either a Pause or an R/S. In the latter case, the calculator will stop first with the GCD in the display. Press R/s to obtain the LCM. Do not omit the other PRT commands, however. You could replace them with a NOP if you so desire, but skipping them altogether would upset the direct addresses in the routines, unless you renumber those also.

I intend to bring each time a few short routines for TI-57 and SR-56 users. That is, if members will be so kind and supply them for publishing. I have only a very limited supply of them. I found a good one in TI-SOFTWARE, a German-language newslettter. The author, H.G. Seib, wrote it originally for the TI-58/59 but if you leave the complete LBL A and its definition off, you store 4 in Register 0 and press RST R/S, the program will work as well on the TI-57 and the SR-56. Of course, the PRT's should be either Pauses or R/S, and the OP 20 should be written as 1 SUM 00. The ADV are just paper advances which you can omit. DIV means "divide by".

Solutions to $a^2 + b^2 = c^2$. Find pairs of a and b that satisfy this equation. The program will generate these pairs, ad infinitum.

RCL 00 STO 01 PRT DIV 2 = X^2 - 1 + SUM 01 PRT 2 = SUM 01 PRT 2 ADV OP 20

RCL 00 PRT RCL 01 PRT + 1 = PRT ADV OP 20 RST LBL A 4 STO 00 RST

TI-58/59 users press A to start.

See you next time, Maurice