EDITORIAL

FIRST OF ALL WE HAVE TO ANNOUNCE YOU A CHANGE OF ADDRESS.
FROM NOW ON ALL SUBMITTED PROGRAMS AND CORRESPONDENCE ABOUT PRO-
GRAMMING SHOULD BE SENT TO THE FOLLOWING ADDRESS:
TISOFT
PB 63
2080 KAPELLEN
BELGIUM

ONE OF OUR MEMBERS DREW OUR ATTENTION TO A SMALL PROBLEM.
IT SEEMS THAT SOME OF THE PROGRAMS SUBMITTED TO US AND ACCEPTED
WERE COPIED FROM PROGRAMS PUBLISHED IN REVIEWS. WE ARE AWARE THAT
THOSE THINGS CAN HAPPEN, BUT IT IS VERY DIFFICULT TO DO SOMETHING
ABOUT IT. ON ONE OF OUR SUBMITTAL FORMS THERE IS A PLACE RESERVED
WHERE SUBMITTERS MENTION THE REFERENCES OF THEIR PROGRAMS. IF THEY
TAKE A PROGRAM FROM A REVIEW AND OPTIMIZE IT THEN IT IS OBVIOUS
THAT THEY SHOULD MENTION THE REVIEW AS REFERENCE. WE HOPE THAT THIS
MATTER IS SETTLED FOR NOW !!!
AND NOW ONCE AGAIN AN URGENT DEMAND.
MAY WE ASK YOU, IF YOU ORDER PROGRAMS FROM THE CATALOG, TO MENTION
CLEARLY THE CATALOG NUMBERS OF THE ORDERED PROGRAMS. FURTHERMORE,
MAY WE ASK YOU TO JOIN A LETTER TO YOUR COMMANDS. THE POSTAL ORDER
WE RECEIVE IS OFTEN UNREADABLE OR INCOMPLETE. THIS CAUSES A GREAT
DELAY IN DELIVERY OF THE ORDERED PROGRAMS. THANKS IN ADVANCE !!!!!
THAT'S ALL FOR NOW AND WE HOPE YOU'LL ENJOY THIS ISSUE OF YOUR
NEWSLETTER. HAPPY PROGRAMMING !!!

TISOFT
In one of our previous Newsletters we mentioned the existence of what could be called the Fast-Mode. At that time we promised you more information about this special discovery. It took us some time to find out a few facts. In finding these we were greatly helped by an article in "Display" (a German TI-users review) by M. Neef.

The first problem of course is how to get in this Fast-Mode. There are two possibilities of achieving this. Both methods require the presence of the Master Library Module (ML-1).

The first method goes like this: key in the following program and be sure to respect the given programming steps.

```
005  Pgm
006  02
007  SBR
008  2
009  40
010  9
```

Then press the following keys: LRN RST R/S

You are now in the Fast-mode with the calculator in Fix 0.

This means that an eventual program should start with INV Fix!!

The second method takes care of this little problem and goes like this:

```
000  2
001  4
002  0
003  STO
004  00
005  Pgm
006  02
007  SBR
008  Ind
009  00
010  9
```

Now push LRN RST R/S and you're in Fast-Mode again, but this time in INV Fix too.

From this moment on we suppose that you want to keep the calculator in this so-called Fast-Mode.

In that case a few things should be kept in mind.

It is impossible to use:
1) User defined keys A,B,...,E'
2) Subroutines
3) Module programs
4) RST
5) INV SBR (or RTN)

When you try to use one of these you will find yourself out of the Fast-Mode.

From the keyboard it is possible to use SBR nnn (to start a program for example) but not RST or CP.

Once a program has been entered most keys will start it running. When entering a program by using a magnetic card, it will start running immediately!! So every program on a magnetic card should begin with stopping it!!

TEXAS INSTRUMENTS
SOFTWARE EXCHANGE
To stop a running program R/S is not sufficient. First the display should be made "soft" (it means entering a digit doesn't cancel a number in the display). This can be done as follows:
1) EE INV EE R/S
2) CLR R/S
3) . R/S
4) any digit R/S

A last important fact is that flag 8 (the error flag) doesn't work any more.
That's all for now about the Fast-Mode. It is not very much yet, but with your help it can be much more.
So, find out some more information about the Fast-Mode and don't forget to write us about your findings.

STATUS !!

This article is in fact a continuation of the previous article on the Fast-Mode.
By using the following program:
Pgm 02 SBR 240
when the Master Library Module is installed, we can obtain much more than the Fast-Mode.
The result of this little program depends on the register in which it ends.
To make things easy let's agree on the following numbering:

<table>
<thead>
<tr>
<th>steps</th>
<th>register</th>
</tr>
</thead>
<tbody>
<tr>
<td>000-007</td>
<td>119</td>
</tr>
<tr>
<td>008-015</td>
<td>118</td>
</tr>
<tr>
<td>etc</td>
<td></td>
</tr>
<tr>
<td>944-951</td>
<td>01</td>
</tr>
<tr>
<td>952-959</td>
<td>00</td>
</tr>
</tbody>
</table>

So every register contains 8 programming steps.
When our little program above starts at step 005, it ends at step 009 and that means in register 118.
Now the register in which the program ENDS gives the status of our calculator. That means: flags, fix, angles, fast or not, ..!!
Let's call this register the STATUS-register from now on.
In our example register 118 was the status-register.
By entering the 16 digits of the status-register aforehand we can define the status of the calculator ourselves!!
Before continuing let's agree on the following definitions:
1 program step = 1 byte
1 byte = 2 nibbles
1 nibble = 4 bits
in a byte we can put 2 digits
in a nibble you can put 1 digit

And now we can start and discuss on what was discovered until now about the meaning of the contents of the status-register.
Significance of the status-register

First Byte: first nibble : unknown
    second nibble : this digit defines the Fix-mode
        0  INV Fix
        1  unknown but curious (operations?)
        2  Fix 0
        3  Fix 1
        etc
        9  Fix 7

    Remark: It is impossible to get Fix 8

Second Byte: first nibble : this digit gives the fast mode:
    second nibble : unknown
        0  normal mode
        2  fast mode
        4  fast mode
        6  fast mode
        8  you can read the firmware by pressing
        if this digit is odd then by pressing
        LRN you get a crash (the only thing
        LRN to do is use the ON-OFF switch)
        8  you can read the firmware by pressing
        LRN

Third Byte : first nibble : let's call this digit c
    second nibble : d

Fourth Byte : first nibble : a
    second nibble : b

    The four digits a,b,c and d give you
    the address at which the program pointer is pointing. This address has an
    upper limit 7999 and is calculated by
    the formula:
    address= 800a +80b +8c +d +1
    (when d=9 then d=-1 !)

Fifth Byte : first nibble : let's call this digit a
    second nibble : b

    All key codes are increased by the number 10a+b.
    For example, when 10a+b=23 then STO has code 65 instead of code 42.

Sixth, Seventh and Eighth Byte : these define flags, trace status
    of printer and error status following
    this scheme:

    | 8 | 4 | 2 | 1 |
    |---|---|---|---|
    | byte 6 nibble 1: flag 5 flag 0 |
    | nibble 2: ? ? |
    | byte 7 nibble 1: flag 7 flag 2 |
    | nibble 2: flag 6 flag 1 |
    | byte 8 nibble 1:TRACE TRACE flag 9 flag 4 |
    | nibble 2:ERROR ERROR flag 8 flag 3 |

    Remark: when all six nibbles are 8 then the calculator
    is in a sort of TRACE mode that only can be
    canceled by RST or CP.

Conclusion : there are still a lot of discoveries to make.
    There is still a lot of work for you in it!!!!
DISCOVERING THE PRINTER INTERFACE

This article continues our series of articles about the technical side of the calculator. This time we're concerned about the connection between calculator and printer. The description given here is meant for the PC-100A. The PC-100A printer has a 14-contact connector which allows calculator interface. First of all there are two large contacts which provide power for the calculator since the battery-pack is removed.

**Interface Overview**

Two clock signals are used by the printer in the same manner as the calculator. Internally these clocks are further grouped into 16 state times, S0 to S15. A full cycle of 16 state times is defined as an instruction cycle, approximately 80μs long (see figure 1). The various calculator chips and printer chips must be in synchronisation, or in the same state time in order to communicate. This function is provided by the IDLE signal, an output from the calculator on pin 8. IDLE is a logic signal that switches between V\textsubscript{ss} and V\textsubscript{dd}. The negative transition of IDLE (V\textsubscript{ss} to V\textsubscript{dd}) sets the leading edge of S0 as shown in figure 1. When the positive transition of IDLE occurs at S1 and remains high (V\textsubscript{ss}) until S0, the calculator is in the calculate mode. When IDLE occurs at the leading edge of S15 the calculator is in the display mode and is scanning the keyboard. The printer must monitor the negative transition of IDLE and set its state-time counter accordingly.

![Diagram of printer interface](image)

Figure 1

TEXAS INSTRUMENTS

SOFTWARE EXCHANGE
All calculator instructions are transmitted serially on the IRG-line. Each instruction is 16 bits long, one per state time. The instruction codes are presented in the following table.

### Instruction Codes

<table>
<thead>
<tr>
<th>State Time</th>
<th>S15</th>
<th>S14</th>
<th>S13</th>
<th>S12</th>
<th>S11</th>
<th>S10</th>
<th>S9</th>
<th>S8</th>
<th>S7</th>
<th>S6</th>
<th>S5</th>
<th>S4</th>
<th>S3</th>
<th>S2</th>
<th>S1</th>
<th>S0</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOAD</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>FUNCTION</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CLEAR</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>STEP</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PRINT</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PAPER ADVANCE</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Data is transmitted between calculator chips and from the calculator on the EXT line. EXT is a logic signal with one bit of data each state time. S3 through S9 are the state times when data is sent to the printer. The LSB (Least Significant Bit) is sent during S3 and MSB (Most Significant Bit) during S9, as shown in figure 1. Figure 2 gives the function and character codes.

![Figure 2: Character Codes](image)

**Texas Instruments**

**Software Exchange**
Some function codes are given hereafter:

S9 S3
1 0 1 0 1 1 1 SIN 1 1 0 0 1 1 0 STO
1 0 1 0 1 1 0 COS 1 1 0 1 0 0 0 RCL
1 1 0 0 0 1 0 SUM 0 1 0 0 0 1 0 INV

and so on.

**Detailed Timing**

The printer recognizes six distinct instructions on IRG as shown in the table of the instruction codes. These codes enable the print buffer to be cleared or loaded, a line of loaded characters to be printed and a paper advance to be actuated. The data on EXT is clocked into a shift register each instruction cycle and selectively loaded into a memory upon appropriate IRG command.

When the printer decodes a CLEAR command the print buffer is set to all zeros (blanks on printout) and the character load pointer is set to the right most position.

When the first LOAD command is decoded, the data from EXT is loaded into the right most print position of the memory and the character load pointer is moved one position to the left.

Each character load command received thereafter loads the current EXT into the memory and moves the character load pointer one position left. If this sequence is repeated for more than 20 character loads, the pointer wraps around to the initial position and the right most character is written over. A full twenty characters can be loaded by this character load sequence and subsequently printed. When the left most character in the desired output has been loaded no further character loads are required, the CLEAR instruction loaded the entire memory with the code for a blank.

The PRINT instruction causes the printer to initiate printing of the current contents of memory. The PAPER ADVANCE command will cause the paper to advance one half a line. The STEP command has two distinct functions. If STEP is decoded during the character load sequence it moves the pointer one position left and leaves the memory unchanged (if a CLEAR was used to begin the sequence a blank will be in the memory) regardless of the code on EXT.

The second function of the STEP command occurs when the printer is printing a line i.e. in a print cycle. During a print cycle the STEP command causes the printer to pull the BUSY status line (a status line to let the printer communicate various levels of activity to the calculator) to S3 at S3 of the instruction cycle following the STEP command. A loop is used by the processor controlling the printer to find when the print or paper advance sequence is finished. The FUNCTION command causes the EXT code to be converted by the printer into three character codes for easy loading of often used alpha strings like SIN, COS, INV and so on. There are 40 fixed three-character codes available. The character load pointer is moved over three places by the FUNCTION command. This instruction should only be used in the right most positions of the printout.

An example of the typical load-print sequence is shown below:

desired output

THE SIN OF 30 DEG IS 0.5

Note: each line represents one instruction cycle.

**Texas Instruments**

**Software Exchange**
So, this ends our technical article for this time. In the coming Newsletter we will continue this series of articles by giving some tests (programs and keystrokes) to control the good working order of your calculator.

TEXAS INSTRUMENTS
SOFTWARE EXCHANGE
This time we have only two problems for you, but they both need a rather long explanation.

**Problem 1**

This problem is both for TI-57 and TI-58/59 owners. The problem is rather easy and is taken from TI PCC Notes. The name is "Petals around a Rose". Suppose we make a roll of 5 dice and given the results on the dice we calculate a number called a. A few examples will make it clear where a comes from!!!

When the roll is 3, 4, 1, 6, 5 then a=6

\[
\begin{array}{cccc}
4, & 2, & 6, & 5, \\
3, & 3, & 3, & 6, \\
1, & 1, & 2, & 4, \\
\end{array}
\]

\[
\begin{array}{c}
a=4 \\
a=10 \\
a=0 \\
\end{array}
\]

A few more details will make it more clear. The name of the game is important, a is always even.

When you do not see the problem now, put some dice on the table!!

And now for the problem:

Given the roll of the 5 dice as a five digit number, make a program to calculate a.

Example: if the roll is 34165 then the result of the program should be 6.

As always all your programs are welcome. We are not looking specially for the most sophisticated but we also like the very clear programs.

So good luck with this one!!

**Problem 2**

And now for a difficult one!!!

We ask you to program a game that we found in the latest edition of the French review "Jeux et Stratégie". It is a deduction game that has some relation with the too well known "Mastermind". It is a game for two people, one is the coder the other the decoder. We ask you to make a program so the calculator is the coder.

The game is played on a 9x9 chess board with six colored pegs. These pegs are placed initially in a corner of the board by the coder in a certain order they'll keep during the game (see figure).

The coder chooses the order of the six different colours and places them on the points a,b,c,d,e,f. The colours have to follow each other as chosen, but never diagonally. For example, if the colours are 1,2,3,4,5,6 and the order is 6,3,1,5,4,2 then possible and not possible starting could be:

\[
\begin{array}{c}
y \rightarrow 2 \\
5 \rightarrow 4 \\
4 \rightarrow 5 \\
6 \rightarrow 3 \\
\end{array}
\]

\[
\begin{array}{c}
t \leftarrow 4 \\
t \leftarrow 2 \\
\end{array}
\]

not permitted

**TEXAS INSTRUMENTS**

**SOFTWARE EXCHANGE**
It will be clear from the following that the leading color (6 in our case) can never be placed in the positions a or b for starting the game. Now it's time for the decoder to enter the game. He gives two letters to the coder who represent the edge of the chess board from where he wants to look at the board. The coder then tells him which colors he sees from that edge.

In our example of a possible starting position if the question of the decoder had been AB then the answer of the coder would have been 4, 5, 6. For the question BA it would have been 6, 5, 4 and for AD 4, 2. Now the coder comes into action again. He will deplace all the colours by following some rules:

1. the colours should be moved in the chosen order
2. they move to an empty field next to the original position
3. all pegs should be moved
4. pegs may move diagonally but in the resultant position the order may not be diagonally
5. two following positions should be different (no translation)

Now the decoder enters in action again and so on. When the leading colour reaches the field E the decoder must have found the order of the colours, otherwise he loses the game.

The best solution of this problem will be taken in our catalogue and all good solutions will be rewarded with a free program out of the catalog.

At work and good luck!!!

TI-58 ..... PLUS ???

Want to change your TI-58 into a TI-59 (without magnetic cards, of course) ???
It is possible to do this by adding two memory chips.
Then you will have 120 memories (or rather 100 and 160 programming steps) or 960 programming steps.
Well, TI-Belgium can do it for you!!!
It will cost you 1500 FB and two weeks of patience.
If you're interested then contact the "Service Après-Vente" or "Naverkoopdienst" of TI-Belgium.

Texas Instruments Belgium
Raketstraat / Rue de la Fusée 100
Brussel / Bruxelles

P.S. This change is not possible for a TI-59C and newer types of the TI-58.

TEXAS INSTRUMENTS
SOFTWARE EXCHANGE
Once again, a lot of thanks for the many answers we receive every time for the problems. And now for the answers.

Problem 1

The problem was the classic ladder problem.
Our members had two points of view.
Some of them used a mathematical approach to reduce the problem to the solution of two second degree equations instead of the solution of a higher degree equation. The programs using this method are very quick ones, of course!!
Other members used approximation programs (sometimes after changing the problem a bit for better equations). These methods give slower programs.
We present two possible programs: one for TI-57 (approximation) and one for TI-59/59 (straightforward).
We also give you the mathematical background for both programs.

<table>
<thead>
<tr>
<th>TI-57</th>
<th>TI-59/59</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 31 l RCL 1</td>
<td>000 76 LBL 022 75 -</td>
</tr>
<tr>
<td>01 -56 INV Dsz</td>
<td>001 11 A 023 04 4</td>
</tr>
<tr>
<td>02 51 3 GTO 3</td>
<td>002 33 X² 024 95 =</td>
</tr>
<tr>
<td>03 28 sin</td>
<td>003 85 + 025 34 lX</td>
</tr>
<tr>
<td>04 25 1/x</td>
<td>004 01 1 026 95 =</td>
</tr>
<tr>
<td>05 32 2 STO 2</td>
<td>005 54 ÷ 027 22 INV</td>
</tr>
<tr>
<td>06 75 +</td>
<td>006 34 lX 028 44 SUM</td>
</tr>
<tr>
<td>07 33 1 RCL 1</td>
<td>007 85 + 029 00 00</td>
</tr>
<tr>
<td>08 29 cos</td>
<td>008 01 1 030 44 SUM</td>
</tr>
<tr>
<td>09 25 1/x</td>
<td>009 54 ÷ 031 01 01</td>
</tr>
<tr>
<td>10 32 3 STO 3</td>
<td>010 52 EE 032 93</td>
</tr>
<tr>
<td>11 85 =</td>
<td>011 22 INV 033 05 5</td>
</tr>
<tr>
<td>12 76 X &gt; Y</td>
<td>012 52 EE 034 49 PRD</td>
</tr>
<tr>
<td>13 51 1 GTO 1</td>
<td>013 42 STO 035 00 00</td>
</tr>
<tr>
<td>14 61 0 SBR 0</td>
<td>014 00 00 036 49 PRD</td>
</tr>
<tr>
<td>15 54 1 SUM 1</td>
<td>015 42 STO 037 01 01</td>
</tr>
<tr>
<td>31 81 R/S</td>
<td>016 01 01 038 43 RCL</td>
</tr>
<tr>
<td>25 STO 0</td>
<td>017 53 &lt; 039 00 00</td>
</tr>
<tr>
<td>4 X &gt; Y</td>
<td>018 24 CE 040 91 R/S</td>
</tr>
<tr>
<td>20 53 &lt;</td>
<td>019 65 X 041 43 RCL</td>
</tr>
<tr>
<td>R/S</td>
<td>020 53 &lt; 042 01 01</td>
</tr>
<tr>
<td></td>
<td>021 24 CE 043 92 RTN</td>
</tr>
</tbody>
</table>

Preparations for using the program are:

STO 1
STO 5
45
The number in reg. 0
is the number of iterations and can be chosen. The time of execution and the precision depend on it.

Mathematical background

TI-57 program:

The problem is reduced to one where we have to find an angle.
This reduction is done as follows:

TExas instruments
SOFTWARE EXCHANGE
\[
\cos \alpha = \frac{x}{4} \Rightarrow x = 4\cos \alpha \\
\sin \alpha = \frac{y}{4} \Rightarrow y = 4\sin \alpha \\
\sin \alpha = \frac{1}{a} \Rightarrow a = \frac{1}{\sin \alpha} \\
\cos \alpha = \frac{1}{b} \Rightarrow b = \frac{1}{\cos \alpha} \\
a + b = 4 \\
\frac{1}{\sin \alpha} + \frac{1}{\cos \alpha} = 4 \Rightarrow 1 + \tan \alpha = 4\sin \alpha
\]

So the problem is reduced to the following: find an angle between \(0^\circ\) and \(90^\circ\) which fulfills this condition.

TI-58/59 Program:

From the equation \(1/x + 1/y = 1\) given as a hint in the previous Newsletter we can easily deduce:

\[x + y = xy\]

The other equation can be written:

\[(x + y)^{2} - 2xy = k^{2}\] (k being 4)

This then leads to:

\[(x + y)^{2} - 2(x + y) = k^{2}\]

This is a quadratic equation with \(x+y\) as unknown. Then \(x+y\) can be calculated and both \(x+y\) and \(xy\) are known. Then \(x\) and \(y\) can be found from another equation of the second degree. Indeed, let the solution for \(x+y\) be successives then the equation to solve is:

\[z^{2} - sz + s = 0\]

So, that's it for problem 1.

We received good solutions from the following members: Vonhoff, Jani, Bleekemolen, Ristanovic, Deleau, Linet, Vanden Driessche and Ruelens. Thank you and congratulations!!!

Problem 2

It seems there were no difficulties with this one. We publish three possible solutions of the problem. One of the solutions takes advantage of the fact that the Dsz-function doesn't take in account the decimal part of the number in the register. Here they are:

| 000 08 8 | 000 78 S+ | 000 08 8 |
| 001 93 . | 001 09 9 | 001 93 . |
| 002 07 7 | 002 93 . | 002 07 7 |
| 003 42 STD | 003 07 7 | 003 42 STD |
| 004 00 00 | 004 72 ST+ | 004 09 09 |
| 005 72 ST+ | 005 00 00 | 005 43 RCL |
| 006 00 00 | 006 69 DF | 006 09 09 |
| 007 75 . | 007 30 30 | 007 72 ST+ |
| 008 01 1 | 008 43 RCL | 008 09 09 |
| 009 95 . | 009 00 00 | 009 57 Dsz |
| 010 77 CE | 010 77 CE | 010 09 09 |
| 011 00 00 | 011 00 00 | 011 00 00 |
| 012 03 03 | 012 04 04 | 012 05 05 |
| 013 91 R/S | 013 91 R/S | 013 91 R/S |

Good solutions were given by the members: Douws, Bleekemolen, Jani, Ristanovic, Gillis, Poels, Hens, Ernst, Deleau, Vanden Driessche, Ruelens and Ducros. Congratulations!!

TEXAS INSTRUMENTS
SOFTWARE EXCHANGE
Problem 3

For this problem there are two possibilities. One can make a program that is very long (about 480 steps) but does the job very quick. It is also possible to make extremely short programs but then we need a lot of time to fill the registers.

We thought about taking a solution that has not too many programming steps and no need to fill registers.

The program chosen has 86 steps and gives a nice printout where the codes of the characters can be seen.

```
006 76 LBL 022 95 = 044 68 DP 066 98 ADV
001 12 B 023 22 INV 045 05 05 067 91 R/S
002 43 RCL 024 52 EE 046 92 RTN 068 76 LBL
003 00 00 025 92 RTN 047 43 RCL 069 11 A
004 65 x 026 76 LBL 048 03 03 070 69 DP
005 01 1 027 13 C 049 52 EE 071 00 00
006 52 EE 028 12 B 050 08 8 072 01 1
007 08 8 029 69 DP 051 95 = 073 42 STD
008 85 + 030 02 02 052 22 INV 074 00 00
009 69 DP 031 69 DP 053 52 EE 075 13 C
010 20 20 032 20 20 054 69 DP 076 96 ADY
011 43 RCL 033 12 B 055 01 01 077 47 CMS
012 00 00 034 55 + 056 13 C 078 08 8
013 65 x 035 01 1 057 03 3 079 42 STD
014 01 1 036 00 0 058 44 SUM 080 02 02
015 52 EE 037 00 0 059 00 00 081 69 DP
016 04 4 038 95 = 060 69 DP 082 23 23
017 85 + 039 69 DP 061 23 23 083 61 GTO
018 69 DP 040 03 03 062 97 D Sz 084 00 00
019 20 20 041 12 B 063 02 02 085 47 47
020 43 RCL 042 69 DP 064 00 00
021 00 00 043 04 04 065 47 47
```

To start the program just press A.

The following members gave correct solutions for this problem: Van Boxel, Gillis, Jani, Hens, Deleau and Ruelens.

The solution we gave is neither the shortest nor the fastest.

Thanks a lot for the interest shown for this part of our Newsletter.
ABOUT PPX

It was brought to our attention by TI-Belgium that from this year on PPX (the American TI-users club) accepts members from outside the US again. We thought this is important news.

Now, we as a club, already are member of PPX.
The conditions of membership are the following:

  membership fee is 30$ !
The cost of a program from PPX is 4$.

Our proposition now is the following.
You can order your PPX-programs as you did before and at the same price. It saves you the 30$ membership fee and you have the same advantages. After the program ordered is sent to the person who ordered it, it is included in our (TISOFT) catalog. The next time that particular program can thus be ordered from our catalog at our prices.

Of course with every addendum to our catalog we'll update the PPX professional category index too and we'll keep you at all times informed about the articles appearing in PPX-exchange (the newsletter of the American club).

We hope that in this way we'll be of still more help to you.

ABOUT THE CODING FORMS

May we ask those members who are in the possession of a printer to take special care of their coding forms. They are used to stick their listings, made by the printer, on the coding forms.
That's fine, but it would be better if they did stick them just to one side and not both sides. The reason we ask this is that the copies made on our copy-machine are much better then, because the coding form is much more flexible. Thanks in advance!!

SUPPLEMENTARY PROBLEM!!

Because of the difficulty of the second problem in the problem corner we propose a third one both for TI-57 and TI-58/59.
It is a rather simple problem about the cosine law.
The routine should compute c out of the cosine law:

\[ c^2 = a^2 + b^2 - 2ab\cos\theta \]

given a, b and C.
The program should not exceed 15 steps on TI-58/59 and 12 steps on TI-57 (input a, b and C included!!).
Good luck for this one too !!!!
PRACTICAL INFORMATION ABOUT TISOFT

1. WE DO NOT DEAL IN HARDWARE OR SOFTWARE DISTRIBUTED BY TEXAS INSTRUMENTS. WE ONLY PUBLISH THE CONTENTS OF MODULES AND PACKETS BECAUSE SOME CLUB-MEMBERS MAY BE INTERESTED.

2. TO BECOME A MEMBER OF TISOFT UNTIL DECEMBER 31ST 1981 TRANSFER 400 FB TO OUR POSTAL ACCOUNT (BELGIAN RESIDENTS ONLY):
   PCC 000 - 1149689 - 45
   TEXAS INSTRUMENTS SOFTWARE EXCHANGE CLUB
   TENNISSTRAAT 16
   2610 WILRIJK (BELGIUM)
   THE MEMBERSHIP FEE IS 500 FB FOR NON-BELGIAN RESIDENTS (IF PAID BY POSTAL ORDER) AND 600 FB (WHEN PAID OTHERWISE).

3. WHAT HAPPENS WHEN YOU BECOME A MEMBER?
   A) YOU MAY SUBMIT YOUR OWN PROGRAMS TO THE CLUB ON SPECIAL FORMS. YOU RECEIVE WHEN BECOMING A MEMBER. FOR EVERY ACCEPTED PROGRAM YOU MAY FREELY CHOOSE TWO OTHERS FROM OUR CATALOG IN EXCHANGE.
   B) WE SEND YOU OUR CATALOG AND THE ADDENDA AS SOON AS THEY APPEAR.
   C) YOU RECEIVE OUR "NEWSLETTER", APPEARING FOUR TIMES A YEAR.
   D) YOU MAY ORDER PROGRAMS FROM OUR CATALOG AND FROM THE PPX-CATALOG.

4. A) SEND ALL YOUR PROGRAMS AND LETTERS ABOUT PROGRAMMING TO THE FOLLOWING ADDRESS:

   TISOFT
   PB 63
   2080 KAPELLEN (BELGIUM)

   B) FOR FRENCH RESIDENTS THE CORRESPONDANCE ADDRESS IS:

   TISOFT
   VAN EYCKLEI 41A/5
   2000 ANTWERPEN (BELGIUM)

   C) FOR ALL LETTERS ABOUT MONEY MATTERS, PROGRAM ORDERS AND ADDRESS CHANGES WRITE TO:

   TISOFT
   TENNISSTRAAT 16
   2610 WILRIJK (BELGIUM)

5. PROCESSING OF YOUR SUBMITTED PROGRAMS AND ANSWERS TO LETTERS MAY BE A LITTLE SLOWER DURING THE SUMMER MONTHS.

6. A FREE MEMBERSHIP IS OFFERED TO THE MEMBERS FOR THE NEXT CALENDAR YEAR WHEN COMPLETING A SUBMISSION OF 15 ACCEPTED PROGRAMS.

7. DO NOT SEND REGISTERED LETTERS. WE CAN NOT POSSIBLY RECEIVE THEM.

8. ALL CORRESPONDANCE SHOULD BE IN ENGLISH.

9. ALL SUBMITTED PROGRAMS SHOULD BE STATED IN ENGLISH.

10. BACK-ISSUES OF THE "NEWSLETTER" CAN BE OBTAINED BY TRANSFERRING 100 FB PER VOLUME TO OUR PCC-ACCOUNT.

11. THE "NEWSLETTER" IS SENT ONCE TO NON-MEMBERS WHO SENT THEIR WARRANTY CARD TO TI-BRUSSELS TO INFORM THEM OF THE EXISTENCE OF TISOFT.

TÉXAS INSTRUMENTS
SOFTWARE EXCHANGE
RESP. EDITOR : TH. COPPENS  

POSTBOX 63  

2080 KAPELLEN (BELGIUM)