# TI-59 Emulator User's Manual

**www.claudiolarini.altervista.org**

Last revision August 03, 2023

# 1. REQUIREMENTS

The "*TI-59 emulator*" program, initially written for the TI-59 and then subsequently extended to other models of the Texas Instruments of the years 1975-1983, is available in two versions and precisely:

- a **16-bit** version written in Microsft **Basic P.D.S. 7.1** and consists of 14 single emulation programs and 3 utility programs. The software has been tested under MS-DOS 6.22 and various versions of Window (3x, 9x, NT/2000/XP, Vista and 7/8/10 32 bit) via command prompt; as known 16-bit executables don't work under 64-bit Operating System and you need a virtualization software (like *VM player*);

- a **32/64-bit** written using **QB64** (current version is **1.50**), which allow to be executed under any modern environment (from Win XP SP3 and up), keeping in the meantime the total compatibility with the original 16-bit version. Every executable files have their name ending with "**-64**".

*Note:* 16-bit executables don't work correctly under DosBox 0.74.

**Important:** English version of the emulation programs is obtained with the **/ENG** switch (see § 4.1)

## 2. INSTALLATION

All the software is contained in a single compressed file, ***ti59emul.rar***. For installation it is enough to unzip it in the root directory of any available device (usually hard disk C:): it's created - or overwrited if already existent – the **\TI59** folder with the following sub-folders:

```
─── TI59 ────┬────── 57TO59 (used by 57TO59.EXE translator)
             │
             ├────── BAK (for temporary files)
             │
             ├────── CRAM (for CRAM modules - TI-66)
             │
             ├────── CROM (for CROM modules - TI-59/58/58C)
             │
             ├────── DOCUMENT (help & documentation)
             │
             ├────── EXE64 (64 bit executables)
             │
             ├────── FORM (.FRM files for mouse use)
             │
             ├────── HRAST (import/export file for .M59 format)
             │
             ├────── MENU (Control menu items – a file for every model)
             │
             ├────── PRINTERS (.PRN files for the emulator’s printings)
             │
             ├────── PROG ─┬── (program library to use in .SOA format)
             │             │
             │             ├── BAX59 (programs to be compiled with BAX-59)
             │             │
             │             ├── LIMACE57 (programs to be compiled with LIMACE-57)
             │             │
             │             ├── SR52 (programs per SR-52)
             │             │
             │             ├── SR56 (programs per SR-56)
             │             │
             │             ├── SR60A (programs per SR-60A)
             │             │
             │             ├── TI53 (programs for TI-53)
             │             │
             │             ├── TI55 (programs for TI-55)
             │             │
             │             ├── TI55II (programs for TI-55II)
             │             │
             │             ├── TI62 (programs for TI-62)
             │             │
             │             └── TIMBA (programs for TI-MBA)
             │
             ├────── SCHEDE59 (programs on magnetic cards - TI-59)
             │
             ├────── SCHEDE52 (programs on magnetic cards - SR-52)
             │
             ├────── SCHEDE60 (programs on magnetic cards - SR-60A)
             │
             └────── SCHERMI (screen files in .BLD format)
```

*<u>Note for Windows Operating Systems:</u>* No icon will be created on the user desktop.

The **executables**, according to the last available version, are in two folders: the emulation programs are highlighted in **red**, in **brown** those with the new memory model, in **blue** the utility programs, in **green** the programs that make up *BAX59*, the cross-compiler from BASIC Dartmouth to TI-59 AOS and the one from *Limace* (a BASIC-like language) to TI-57 AOS.

```
                 Installation drive
                     ▼
Il volume nell'unità X è HD_250
 Numero di serie del volume: 0CF0-4490

 Directory di X:\TI59
03/08/2023  11.27          2.849.280 57TO59-64.EXE
03/08/2023  11.03            101.088 57TO59.EXE
10/07/2023  15.58          2.697.216 BUILDER-64.EXE
06/06/2023  15.19             58.674 BUILDER.EXE
29/07/2023  10.30          2.512.896 CC5B-64.EXE
29/07/2023  10.33            100.964 CC5B.EXE
11/05/2023  16.11          2.757.120 ELIMACE2-64.EXE
11/05/2023  16.10             69.512 ELIMACE2.EXE
11/05/2023  16.10             69.890 FLIMACE2.EXE
14/07/2023  10.51          2.708.480 IMPORT-64.EXE
14/07/2023  10.47             61.096 IMPORT.EXE
14/07/2023  09.37            112.608 KC.EXE
10/07/2023  15.58          2.675.200 LEGGIMI-64.EXE
23/04/2023  14.28             54.388 LEGGIMI.EXE
01/08/2023  22.51            158.108 NEWT58CX.EXE
01/08/2023  22.49            149.276 NEWTI58C.EXE
01/08/2023  21.08            154.580 NEWTI59.EXE
01/08/2023  21.38            154.670 NEWTI59A.EXE
01/08/2023  22.48            161.252 NEWTI59G.EXE
18/07/2023  10.41            144.194 NEWTI66.EXE
14/07/2023  09.37            110.114 SEP.EXE
23/06/2023  11.23             67.924 SR50.EXE
14/07/2023  09.29            134.612 SR52.EXE
10/07/2023  16.27            123.134 SR56.EXE
10/07/2023  16.27            137.358 SR60A.EXE
10/07/2023  16.27            110.480 TI53.EXE
10/07/2023  16.27            123.384 TI55.EXE
10/07/2023  16.27            127.134 TI55II.EXE
10/07/2023  16.27            139.390 TI57.EXE
10/07/2023  16.27            126.564 TI57LCD.EXE
01/08/2023  20.57            149.158 TI58C.EXE
01/08/2023  21.02            157.536 TI58CX.EXE
01/08/2023  20.53            153.816 TI59.EXE
01/08/2023  20.54            154.076 TI59ALT.EXE
02/08/2023  17.19            160.812 TI59GR.EXE
10/07/2023  16.27            133.276 TI62.EXE
18/07/2023  09.55            144.034 TI66.EXE
10/07/2023  16.27            123.212 TIMBA.EXE
10/07/2023  16.27            106.230 TIPROG.EXE
09/07/2023  17.17             84.008 ZONE.EXE
              40 File     20.316.744 byte

 Directory di X:\TI59\EXE64

10/07/2023  15.57          2.968.064 KC-64.EXE
02/08/2023  17.11          3.593.728 NEWTI58C-64.EXE
02/08/2023  17.12          3.629.056 NEWTI58CX-64.EXE
02/08/2023  17.05          3.675.136 NEWTI59-64.EXE
02/08/2023  17.07          3.675.136 NEWTI59ALT-64.EXE
02/08/2023  17.09          3.735.552 NEWTI59GR-64.EXE
```

```
12/07/2023  10.10         3.425.280 NEWTI66-64.EXE
10/07/2023  15.57         2.911.232 SEP-64.EXE
10/07/2023  15.58         2.735.616 SR50-64.EXE
12/07/2023  10.15         3.314.688 SR52-64.EXE
10/07/2023  15.56         3.151.872 SR56-64.EXE
10/07/2023  15.56         3.313.152 SR60A-64.EXE
10/07/2023  15.34         2.944.512 TI53-64.EXE
10/07/2023  15.35         3.072.000 TI55-64.EXE
10/07/2023  15.35         3.060.224 TI55II-64.EXE
12/07/2023  10.03         3.195.904 TI57-64.EXE
10/07/2023  15.36         3.025.920 TI57LCD-64.EXE
02/08/2023  16.04         3.580.416 TI58C-64.EXE
02/08/2023  17.16         3.614.720 TI58CX-64.EXE
02/08/2023  09.28         3.663.360 TI59-64.EXE
02/08/2023  09.30         3.661.312 TI59ALT-64.EXE
02/08/2023  15.50         3.722.240 TI59GR-64.EXE
10/07/2023  15.40         3.133.440 TI62-64.EXE
10/07/2023  15.47         3.426.304 TI66-64.EXE
10/07/2023  15.48         3.119.104 TIMBA-64.EXE
10/07/2023  15.50         2.910.720 TIPROG-64.EXE
              27 File     86.258.805 byte

     Totale file elencati:
              67 File    106.575.432 byte
```

An emulator for an **"extended" TI-58C** has also been implemented with additional features: its file name is *TI-58CX .EXE* (or *TI58CX-64.EXE* – see §5.4).

Files starting with "**NEW**" constitute an even more adherent implementation of the real machines, using a new memory model: for this reason they have their own support files that differe from those relating to "standard" emulators.

## 3. AVAILABLE EMULATORS

The emulated calculators are:

- **TI-59** with **PC-100C** printer
  - **TI59.EXE** (standard)
  - **TI59ALT.EXE** (functionally equivalent, but with a different skin)
  - **TI59GR.EXE** (that implements "graphics" on the PC-100C printer– see § 7.2)
- **TI-58** with **PC-100C** printer [included in TI-59 emulator]
- **TI-58C** with **PC-100C** printer
  - **TI58C.EXE** (standard)
  - **TI58CX.EXE** (extended emulator – see § 5.4)
- **TI-57**
- **TI-57LCD**
- **TI-66** with **PC-200** printer
- **TI-55**
- **TI-55 II**
- **SR-52** with **PC-100A** printer
- **SR-56** with **PC-100A** printer
- **TI-53**
- **TI MBA** (financial),
- **SR-60A** (in absolute world preview)
- **SR-50**
- **TI Programmer LCD**

The last two are NOT programmables. There also two emulators for non-TI calculators:

- **Netronics NX3000**
- **Sinclair Enterprise Programmable**.

## 3.1. COMPATIBILITY WITH THE REAL CALCULATORS.

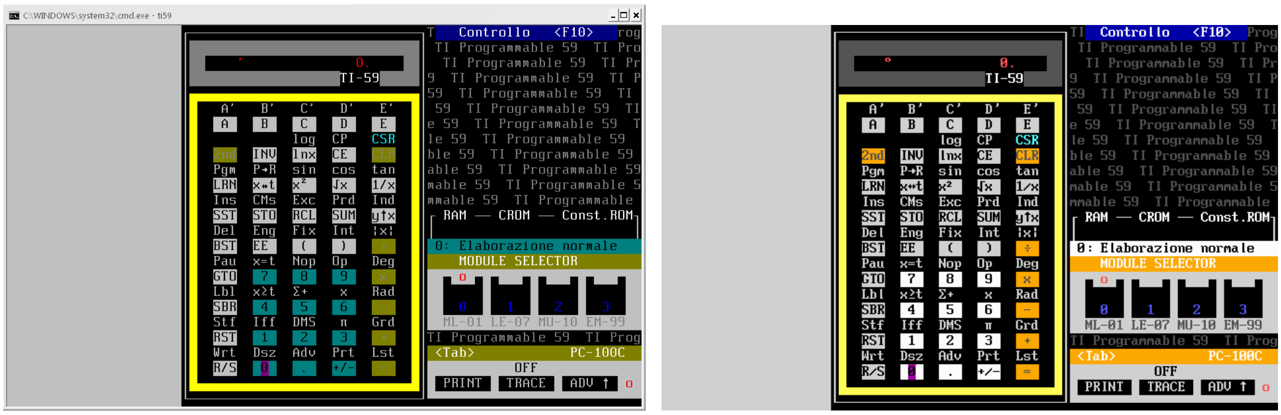
All emulators aren't based directly on the emulation of the machine' ROM, and occasionally it is possible that some real program, despite the care taken in realizing the software, may not run properly and minor changes would be necessary. At the moment these modifications concern particular cases only and they are noticed in the *PROG* folder files. In any case, all the (few) blocking situations of real machines have been corrected such as the sequence **21 38** [2nd sin] encountered in a program on the SR-52, TI-59 and TI-58 or the (apparently harmless) operation **1 EE 99 +/- x 0 =** on the TI-57.

Also, due to some "bugs" of QB64, some (very few) results provided by 32/64-bit programs may be different (wrong) from those provided by **16-bit** programs (right), which therefore remains the **reference version**.

## 4. RUNNING THE EMULATOR

Every program is executed as any other DOS/Windows application and it can be run or in a window or in "all-screen" mode (if the graphic card and/or the Operating System consent this).

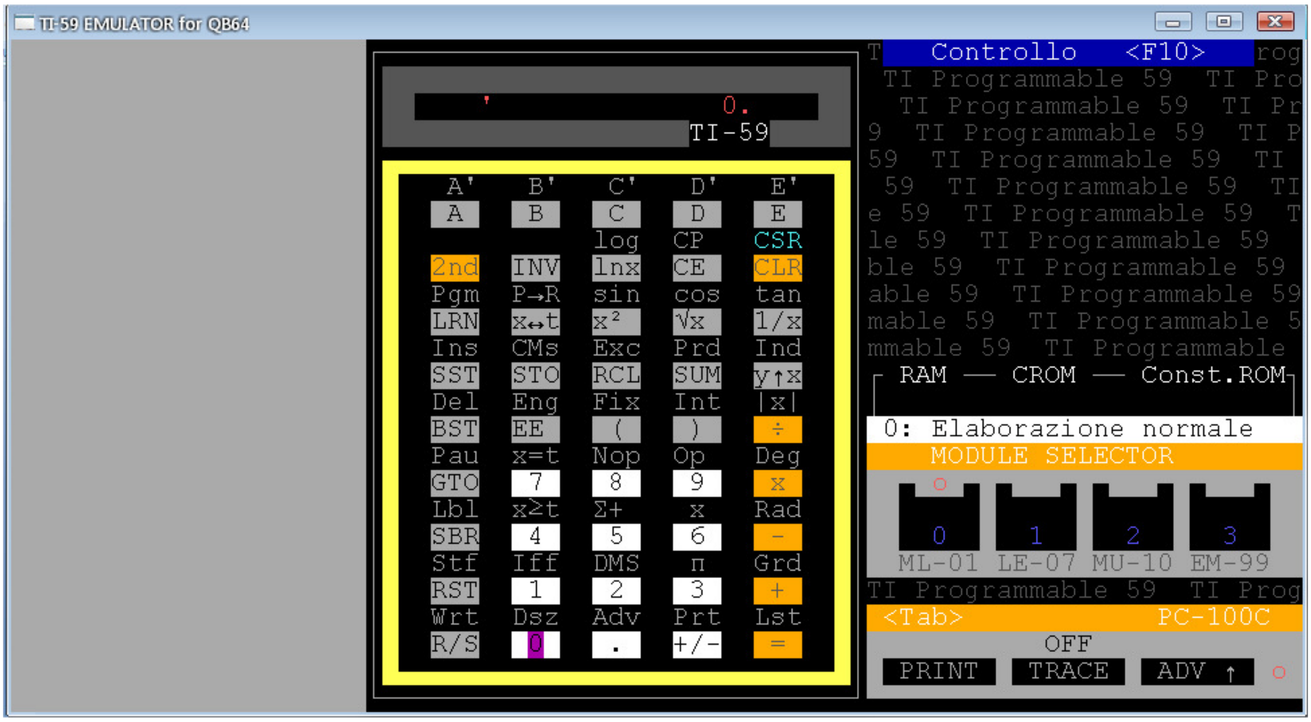
***Important***: The 16-bit versions are programmed to run in full screen, because they manipulate the color palette. Running these applications instead through the Windows Command Prompt that does not manipulate the color palette, we may have chromatically inadequate screens – it is therefore advisable to use the 32/64 bit versions that behave correctly.



*The colors of the 16-bit version for TI59.EXE: on the left in a window instead on the right in full screen mode: note that the Windows Command Prompt does not handle the color palette.*

All versions use **codepage 437**, which must be active before execution.

Below there is a run of the 32/64 bit version, which can only be performed in a window (resizable as desired) **preserving the color palette**.

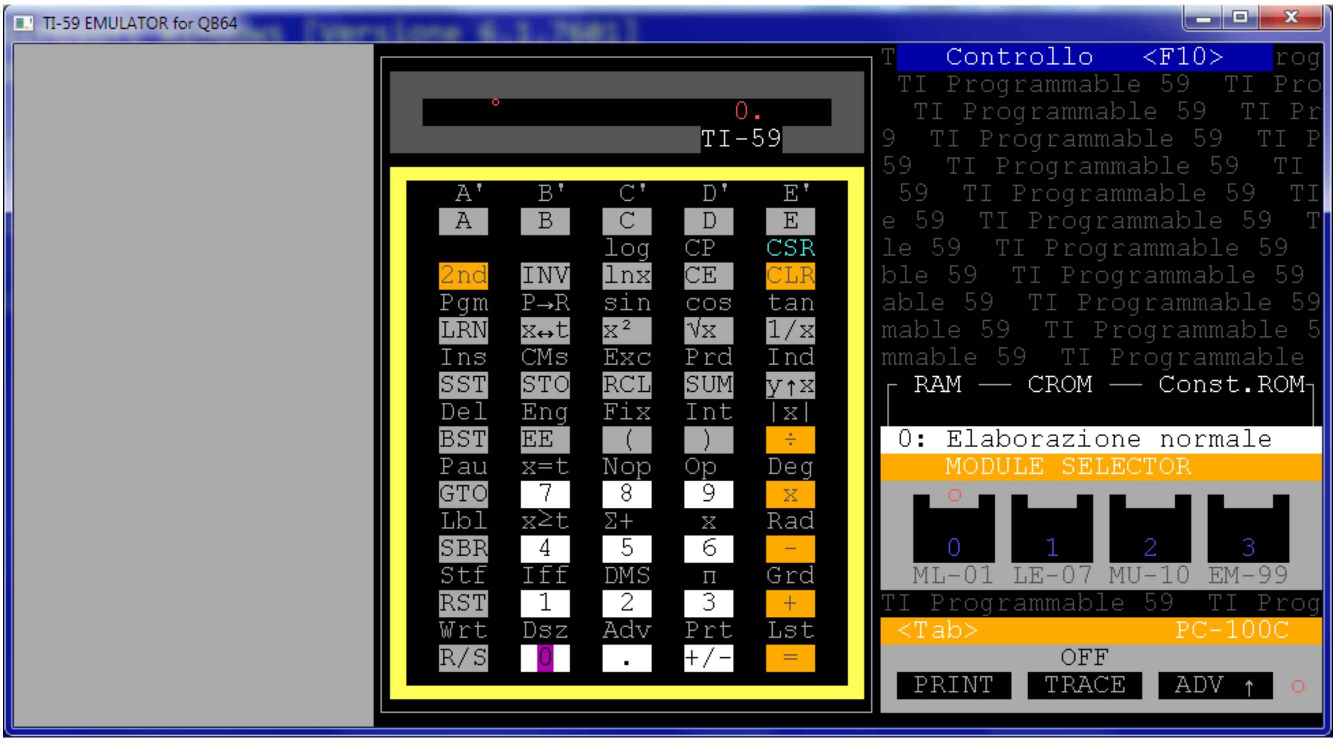


*Note*: The 64-bit versions must executed in \TI59 folder as **EXE64\file_name**

## 4.1 THE COMMAND LINE AND THE SUPPORT FILES.

The complete command line for every emulator is the following:

- **TI59** [/P1|/P2|/P3] [/58] [/58X] [/D] [/A] [/ENG] [/?]
  (for **TI59ALT** and **TI59GR** too)
- **TI58C** [/P1|/P2|/P3] [/D] [/A] [/ENG] [/?]
- **TI57** [/S] [/D] [/ENG] [/?]
- **TI55** [/D] [/ENG] [/?]
- **TIMBA** [/D] [/ENG] [/?]
- **TI55II** [/NUM] [/ENG] [/?]
- **TI57LCD** [/NUM] [/EXT] [/ENG] [/?]
- **TI66** [/P1|/P2|/P3] [/HEX] [/N] [/D] [/A] [/ENG] [/?]
- **TIPROG** [/ENG] [/?]
- **SR52** [/P1|/P2|/P3] [/EXT] [/D] [/A] [/L] [/ENG] [/?]
- **SR56** [/P1|/P2|/P3] [/D] [/A] [/L] [/ENG] [/?]
- **SR60A** [/P1|/P2|/P3] [/A] [/?]
- **SR50** [/A] [/?]

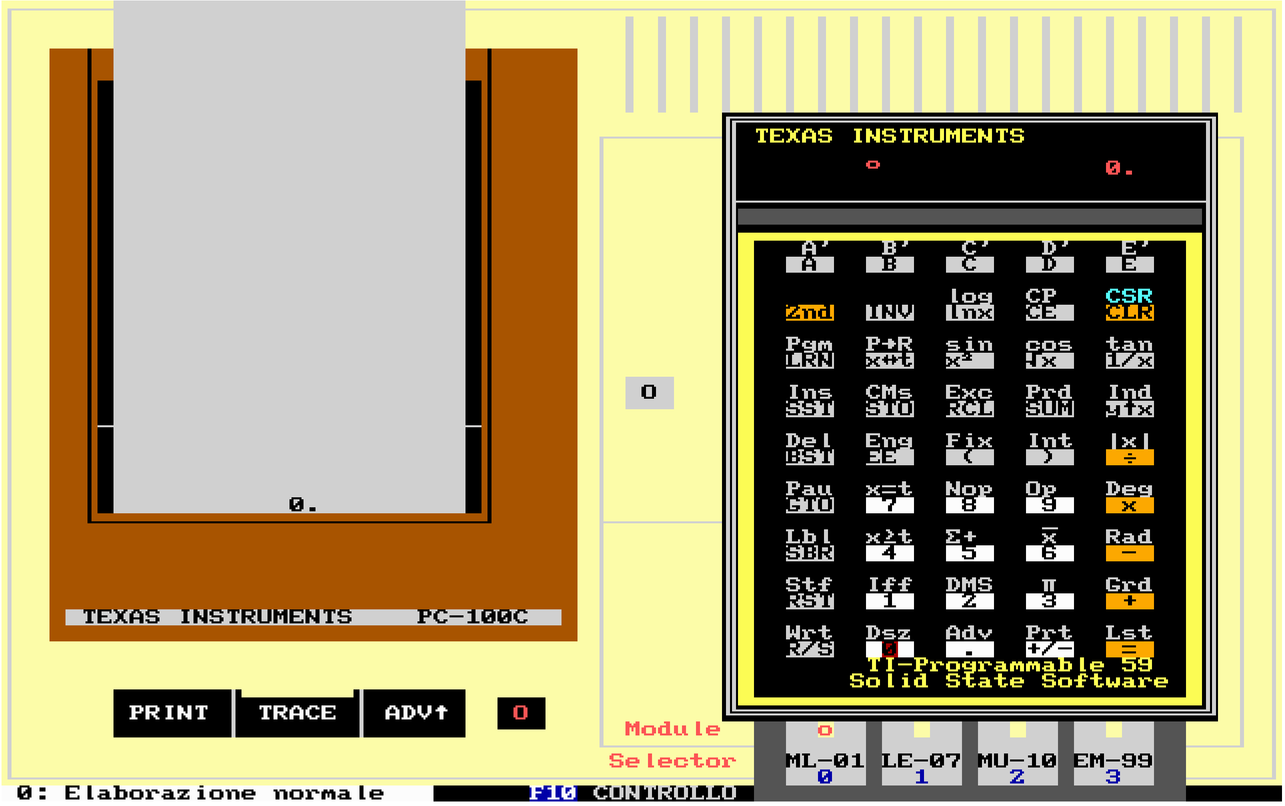


*TI59-64.EXE in execution*

with the following meaning:

- **/P1, /P2, /P3** allow to redirect emulator printings on a PC printer (real or vitualized with a NET USE command) respectively connected on LPT1:, LPT2:, LPT3: by default printouts are directed to a specific file **.PRN** (*TI59.PRN, TI58.PRN, TI58X.PRN, TI58C.PRN, TI66.PRN, SR52.PRN, SR56.PRN* e *SR60.PRN*).

- **/A** (all models with printer) allow to “append” the printouts to the previous .PRN file, otherwise every time you run the program, the content of the . PRN file is cleared.
- **/58** for the TI-59 allows to emulate a TI-58 while with **/58X** it allows to emulate an "extended" TI-58 with 4 memory chips. For a better usage, two batch files (TI58.BAT e TI58X.BAT) for the 16 bit version are provided and two others for the 32/64 bit version (TI58-64.BAT e TI58X-64.BAT in EXE64 folder).
- **/S** for the TI-57 allows to view the intermediate results during a program execution as in the real machine.
- **/HEX** for the TI-66 allows to list the program on PC-200 printer with the hexadecimal codes of the instructions.
- **/N** for the TI-66 allows you to perform the emulator at "normal" speed: by default the speed is in "turbo" mode.


0.
TEXAS INSTRUMENTS PC-100C
PRINT
TRACE
ADV↑
0
TEXAS INSTRUMENTS
0.
A' B' C' D' E'
A B C D E
2nd INV log lnx CP CE CSR CLR
Pgm LRN P→R x⇄t sin x² cos √x tan 1/x
Ins SST CMs STO Exc RCL Prd SUM Ind yˣ
Del BST Eng EE Fix ( Int ) |x| ÷
Pau GTO x=t 7 Nop 8 Op 9 Deg ×
Lbl SBR x≥t 4 Σ+ 5 x̄ 6 Rad −
Stf RST Iff 1 DMS 2 π 3 Grd +
Wrt R/S Dsz 0 Adv . Prt +/− Lst =
TI-Programmable 59
Solid State Software
Module Selector
ML-01 0
LE-07 1
MU-10 2
EM-99 3
0: Elaborazione normale
F10 CONTROLLO

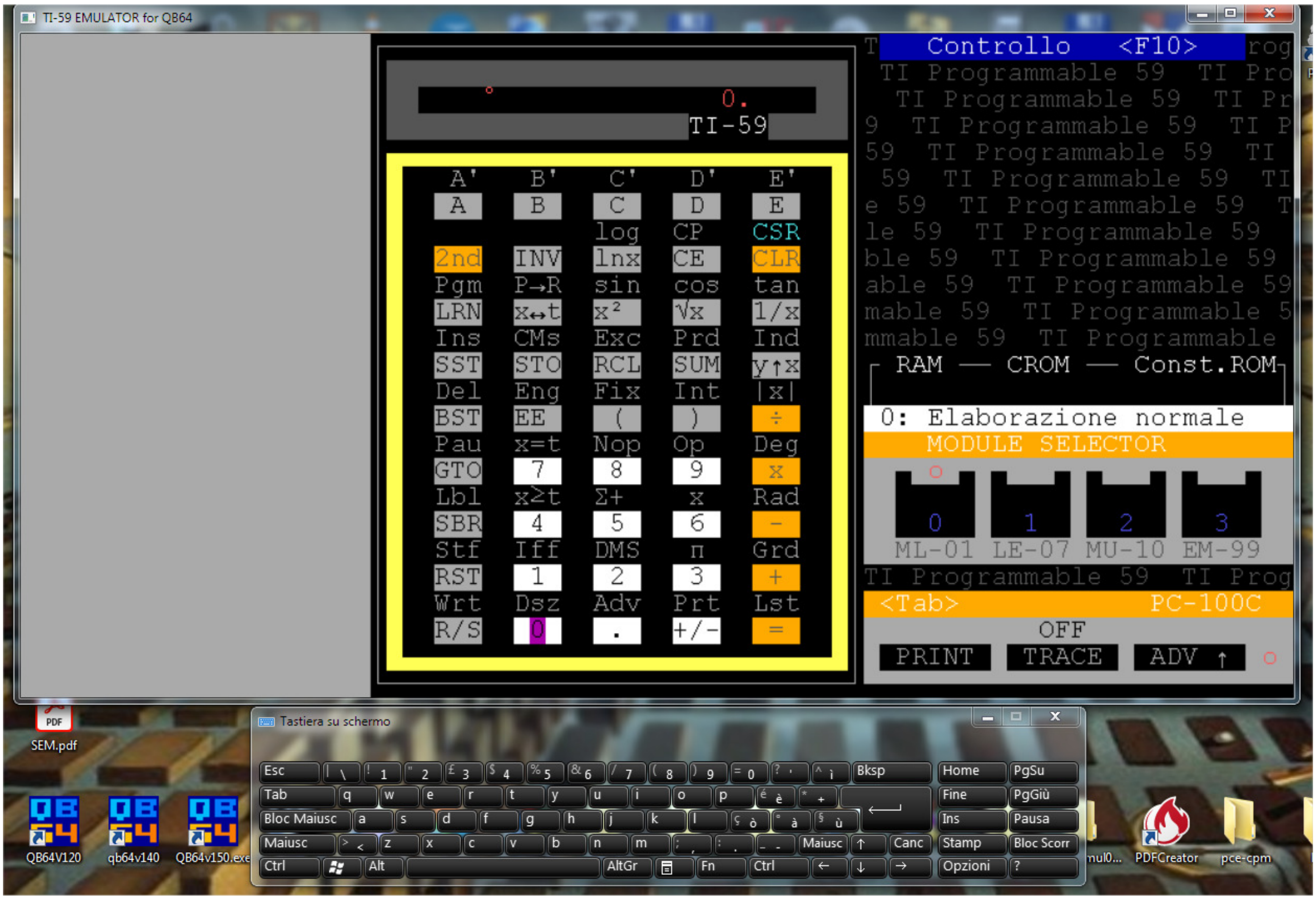
*TI59ALT.EXE in execution: it is functionally equivalent to TI59.EXE*

- **/EXT** for the TI-57LCD emulates the "extended" mode (48 bytes of memory more than the real calculator) in order to be able to perform the programs of the TI-57 LED.
- **/NUM** for the TI-57LCD and the TI-55 II activates the numerical LRN-Mode (that of the real machines): the emulator has, by default, the alphanumeric mode as on the TI-66 for practicality of use. It is possible, during execution, to change from alpha-numeric LRN-Mode to standard numerical one (and vice versa) by pressing **Alt+S**.
- **/EXT** for the SR-52 emulates the “extended” mode (on the real machine it was possible to add two other memory chips that brought the overall memory equal to that of TI-59, though differently organized).
- **/L** (for SR-52 and SR-56) allows you to obtain the programs list on PC-100A as on the real machine, that is, without mnemonics: by default the printing is the same as that obtained by the TI-58/58C/59 on the PC-100C.

- **/A,** for the SR-50 only, allows to emulate the SR-50A calculator, which has only a different keyboard layout.
- **/D** (for various emulators) allows to view the numbers' positions of the display.
- **/ENG** allows you to view the simulators' messages in English.

Each emulator relies on two files with the same name as the emulator and with extensions,**.INI** and **.MEM** which contain, respectively, the configuration settings for the emulator and the constant memory of the machine itself. For example, in the case of the TI-59 we will have:

| **TI59.INI** | **TI59.MEM** |
|---|---|
| 0 ← status of the PC-100C TRACE key<br>5913DGT ← name of the help file of the program currently resident in memory (can be recalled with Shift + F1). | constant memory of the TI-59, in accordance with the memory model used by the emulator (saved at the exit of the emulation program). |



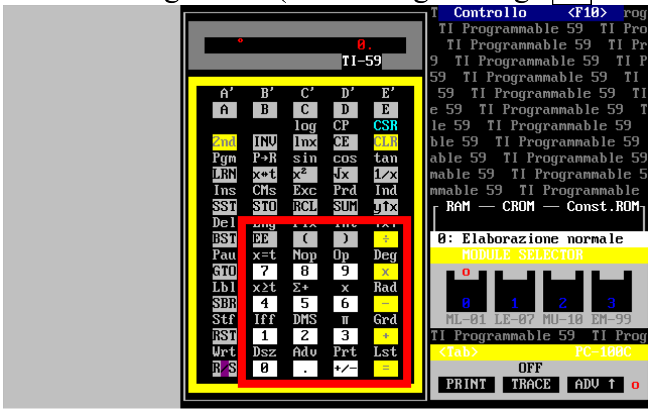
0: Elaborazione normale

*TI59-64.EXE in execution with "on-screen keyboard" under Windows7-64bit*

# 5. USING THE EMULATOR

The emulators an be used, for the moment, only with the keyboard - *except SR50 emulator* - , namely:

- **cursor keys** for movement on the keyboard: at the start the cursor is positioned on 0 key);
- **Enter** key selects the object under the cursor;
- **Home** key brings the cursor back to 0 key;
- the keys from **0** to **9**, ●, +, -,*, /. ^, =, ( , ), \, **lower e** directly activate the numerical section of the emulating model ('\' is the sign change +/- and **e** is EE);



- **function keys** access to special functions (each model has its own specific) including **F10** to access the Control menu: the choice of the menu items is done with the cursor + <Enter>;
- **Tab** accesses the PC-100A/C printer keys (where applicable);
- **Esc** stops long elaborations (all models) – as R/S;
- **Alt+S** modifies angular mode selector (Deg ↔ Rad sulla SR-52 e Deg ↔ Grad sulla SR-56) during program execution;
- **Alt+T** switches the TRACE button of the PC-100 A/C printer during program execution.
- **Alt+G** shows display during program execution (on TI-57/58/58C/59 is equivalent to holding down GTO key)– on TI-66 is **Alt+R,** for RST.

The movement on the keyboard is circular (from the last line you can go to the first and from the first column to the last and vice versa). It's also possible to use the Windows application "On-Screen Keyboard" (*osk.exe*) to be able to use only the mouse, even if not expected in the application.

Pressing 2nd or INV keys is indicated with a "*" above the key and also the angular mode with ° for Deg, ' for Rad and " for Grad - as on the TI-30.

***Note:*** In models with LCD display these features are highlighted in the bottom row of the display itself, as on real machines.

All printouts made using the emulators of the models equipped with a printer (TI-58, TI-58C, TI-66, SR-52, SR-56) are both shown on the screen and written to their own **.PRN** file (e.g. TI59.PRN for TI-59). These files are deleted each time the emulators are reused, so to keep old printouts, you must rename or copy this file under another name before reuse, or run the emulator with the **/A** switch.

All models emulated by the program, contrary to some real calculators such as the TI-57, TI-58, TI-59, SR-52 and SR-56, have **constant memory**, which means that programs, data and some parameters (decimal fix and angular mode) are retained from one execution to other of the emulator.

*It's possible to save a particolar calculator's status saving this file with another name and manually recover it when it needs.*

## 5.1. FUNCTION KEYS

The meaning of the function keys is the following (Shift + function key is showed as **f**):

- **F1** → help system (callable from the Control Panel too): it is organized as a textfile which can be viewed with the cursor keys up and down; with Esc you exit the option.
- **F2** → conversion utility between printers' codes (PC-100C for TI-59/58/58C or PC-200 for TI-66) and ASCII and vice versa: it's useful if you do not have in hands the table of alphanumeric coding *(TI-59/58/58C/66 only)*.
- **F2** → Conversion of TI-57 program to TI-58/58C/59 (*TI-57 only*).
- **F3** → show the internal stack used by the emulator (allowing you to build the TI internal registers)
- **F4** → keycodes table (42 STO, 43 RCL ecc..)
- **F5** → reference tables (HIR suffixes, Op instruction list, CROM module list……)
- **F6** → "cut" PC-100C/PC-200 paper tape.
- **F7** → import and management of the .SOA text files (see 1.6)
- **F8** → allows to store "synthetic" (not-standard) codes in program memory: it's active for TI59GR. EXE, TI66.EXE, TI57.EXE and SR52.EXE only.
- **F9** → DOS Shell (access the command prompt for housekeeping operations).
- **F10**→ control menù (see §5.2).
- **F11**→ activate graphic mode for PC-100 printer (for TI-59 only).
- **F12** → activate CROM Emulator device (for TI-58/58C/59).
- **f1**→ show help (if exist) for the program loaded with Import function (**F7**). For example, if you load the sample file *59FACT.SOA* the part marked in red is shown when you press **Shift F1**.

```
{ Factorial calculation }
{ To execute: n A }

LBL A STO 01 RCL 01 * DSZ 01 00 04 1 = R/S
```

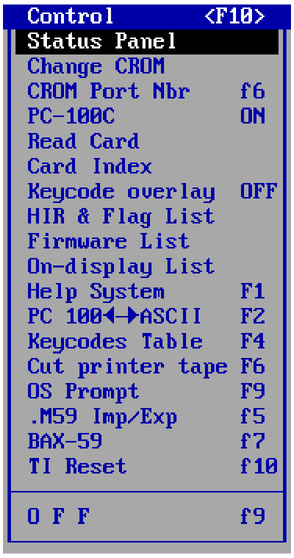
- **f2** → shows help of the CROM Module program actually in use (if activated) in a manner like the non-magnetic card provided by Texas Instruments. For the TI-66 **f2** shows help of the CRAM module actually in use.
- **f3** → sets the path used by Import utility (see F7).
- **f4** → shows the contents of the internal memory used by the emulator, according to the memory model.
- **f5** → allows import/export for files in .M59 format (used in Hrast Programmer TI-59E emulator).
- **f6** → allows to select the active port of 'Module Selector' (for TI-59/58/58C only).

- **f7** → access to BAX-59 cross compiler.
- **f9** → quits the emulator without using the appropriate item in the Control menu (where it exists).
- **f10** → resets memory emulator (for TI 7/58/58C/59 only).

For convenience we report, in the Appendix, also a sticker (in Italian and in English) with the meaning of the function keys.

## 5.2 THE CONTROL MENU

**F10** access the control menu. Every model has its own items. For the TI-59 emulator the following are available:



- **Status Panel** → shows the contents of the TI-59 internal registers and the main settings (angle unit, partition, decimal fix ...)

- **Change CROM** → allows you to "load" a CROM module - actually are all standard module: **Master 01**, **Applied Statistics 02**, **Real Estate/Investment 03** , **Surveying 04**, **Marine Navigation 05**, **Aviation 06**, **Leisure 07**, **Securities Analysis 08**, **Business Decisions 09**, **Math/Utilities 10, Electrical Engineering 11, Agriculture 12, RPN Simulator 13, Structural Enginerring 14** and **Surveying II 15.**. Two special CROM modules are also provided: **00** (**NOCROM**) indicating the empty module and **99** (**EMULCROM** - for its use see § 1.9).

- **CROM Port Nbr** → allows to select active port of 'Module Selector' (0 to 3).

- **PC-100C ON/OFF** → enable/disable the printer.

- **Read Card** → allows reading a magnetic card (see below)

- **Keycode overlay ON/OFF** → enables/disables the display of the numeric key codes (like the "keycode overlay" provided by Texas Instruments).

- **HIR & Flag List** → prints the content of HIR registers and the status of the flags (TI58/58C/59)

- **Firmware List** → prints the list of the firmware coded in AOS (TI58/58C/59 & SR56).

- **On-display List** → lists program codes on the display step by step (TI58/58C/59).

- **Help System (F1)** → emulator general help based on a text-file. Other helps are available for user program currently in memory (**Shift F1**) or for the CROM programs, when selected with **Pgm** instruction. (**Shift F2**).

- **PC100◄─►ASCII (F2)** → conversion utility between PC-100C codes (or PC-200 for TI-66) and ASCII and vice versa (useful if you don't have on hand a table of alphanumeric codes of PC-100C or PC-200).

- **Keycodes Table (F4)** → keycodes table (42 STO, 43 RCL ecc..)

- **Cut printer tape (F6)** → "cut" paper tape of the printer (PC-100A for SR-52 and SR-56, SR-60A internal printer, PC-100C for TI-58/58C/59 and PC-200 for TI-66).

- **OS Prompt (F9)** → access the command prompt for housekeeping operations.

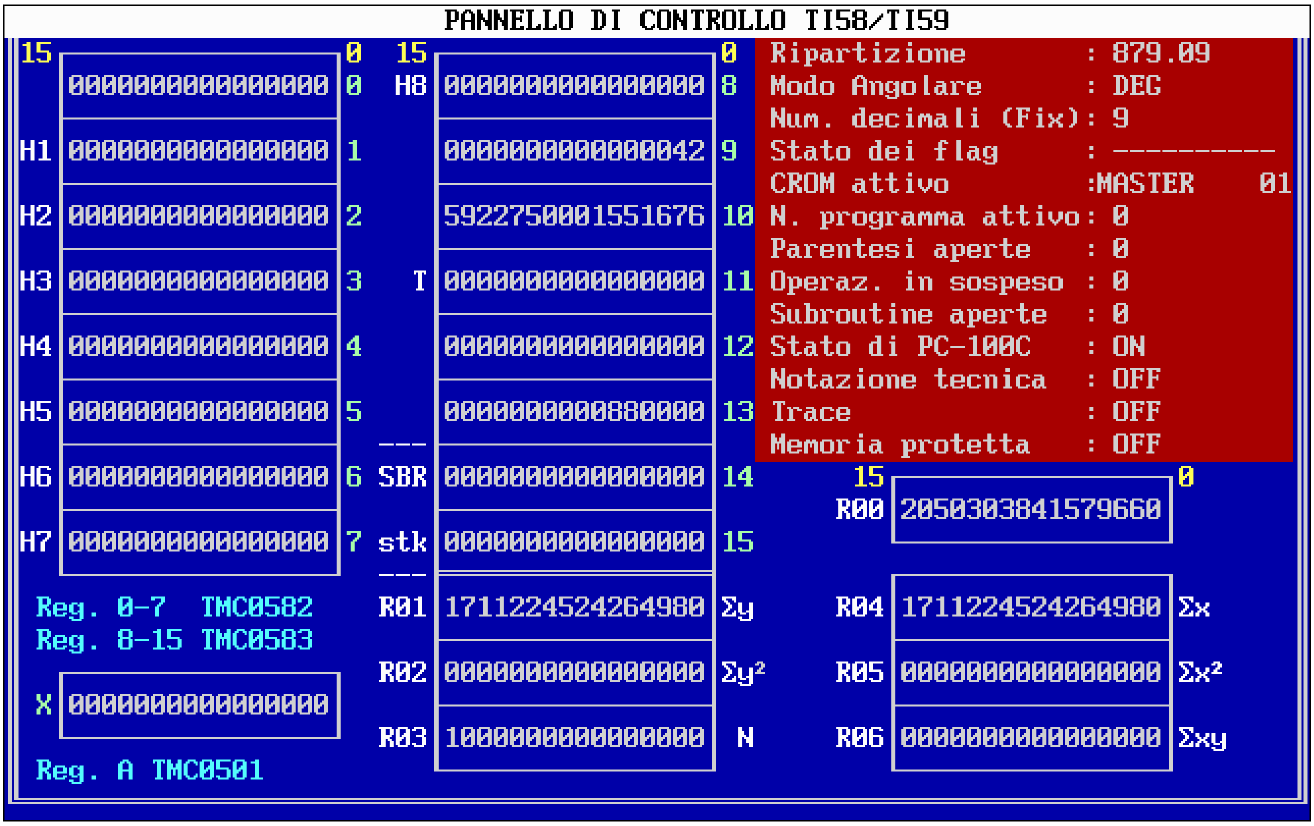
- **.M59 Import/Export (f5)** → allows import/export for files in .M59 format (used in Hrast Programmer TI-59E emulator).

- **BAX-59 Compiler (f7)** → access to BAX-59 cross-compiler (or LIMACE-57 Compiler for the TI-57).

- **TI Reset (f10)** → resets the calculator (with confirmation).

- **OFF (f9)** → quits program. Calculators with the OFF key (eg. TI-57LCD or TI-55II) are simply turned off by selecting the above key.

For each model, the Control menu items are stored on a file in the **MENU** folder.



```
                      PANNELLO DI CONTROLLO TI58/TI59
15                0  15                 0 Ripartizione        : 879.09
   0000000000000000 0 H8 0000000000000000 8 Modo Angolare       : DEG
                                           Num. decimali (Fix): 9
H1 0000000000000000 1    0000000000000042 9 Stato dei flag      : ----------
                                           CROM attivo         :MASTER    01
H2 0000000000000000 2    5922750001551676 10 N. programma attivo: 0
                                           Parentesi aperte    : 0
H3 0000000000000000 3  T 0000000000000000 11 Operaz. in sospeso  : 0
                                           Subroutine aperte   : 0
H4 0000000000000000 4    0000000000000000 12 Stato di PC-100C    : ON
                                           Notazione tecnica   : OFF
H5 0000000000000000 5    0000000000880000 13 Trace               : OFF
                     ---                   Memoria protetta    : OFF
H6 0000000000000000 6 SBR 0000000000000000 14      15                0
                                                R00 2050303841579660
H7 0000000000000000 7 stk 0000000000000000 15
                     ---
 Reg. 0-7  TMC0582   R01 1711224524264980 Σy      R04 1711224524264980 Σx
 Reg. 8-15 TMC0583
                     R02 0000000000000000 Σy²     R05 0000000000000000 Σx²
 X 0000000000000000
                     R03 1000000000000000  N      R06 0000000000000000 Σxy
 Reg. A TMC0501
```

*A Control Panel screen*

## 5.3. UTILITY PROGRAMS

As said before, we have three utility programs available, namely:

- **IMPORT** which manages (as seen before and for all models) import of .SOA files in the various emulation programs. Its use is automatically managed by the emulators themselves (see § 6.3 and following);

- **57TO59** that allows you to "translate" a TI-57 program in its corresponding TI-58/58C/59: this conversion allows you to replicate the behavior of the TI-57 when using the t-register and the statistical instructions which, as it is known, have a different registers’ use in the two models (see § 10.1);

- **BUILDER** that allows you to "build" a custom CROM module or examines an existing one (see § 7.2).

## 5.4 NEW FEATURES

Some new features have been implemented (also existing in other models), namely:

- **CSR** (code 20) replaces **2nd Pgm 01 SBR CLR**: this is an instruction existent on the TI-66– (for TI-58/58C/59);
- **CP** (code 10) clears the TI-57 program memory in “LRN mode” mode as in the TI-57LCD: it wasn’t present on the original machine because it was enough to turn it off ....;
- **x!** on the TI-66 (the factorial instruction is present only on SR-52, TI-57LCD, SR-60A);
- **HIR Ind** is available as instruction in a program with codes 82 40 XX: on real models this was possibile from keyboard only (for TI-58/58C/59);
- **INV Op 00** prints a list with the meaning of Op codes (for TI-58/58C/59/66);
- some "synthetic" functions, obtained by using sequence of "unorthodox" keystrokes, are available for TI59GR.EXE, TI66.EXE, TI57.EXE e SR52.EXE: these functions are generated with the **F8** function key (see § 7.3 for an example);

A separate case is the "extended" emulation of the TI-58C (*TI58C.EXE* and *TI58CX-64.EXE*) which allows, for EXPERIMENTAL PURPOSE only, the use of new plausible instructions without "distorting" the emulation and that Texas Instruments could have easily added by optimizing the ROM of its calculators:

- **x!** (implemented as $\boxed{\text{INV}}$ $\boxed{1/x}$) – see SR-52 and TI-57II;
- **MOD** (implemented as $\boxed{\text{INV}}$ $\boxed{\div}$) - the remainder of division - which the HP-41 has as a standard instruction;
- **extended BST** that allows the complete editing of the number currently in input;
- **integer powers of negative bases**, as on the HP. For example 5 $\boxed{+/-}$ $\boxed{y^x}$ 3 supplies -125 and not 125 with flashing display as on real models.

Everything else is in Texas Instruments manuals "*Personal programming*".”*Introduction to Programming*”, “*TI-66 Sourcebook*”, ......

## 5.5 MY POINT OF VIEW ON TI-59 EMULATION PROGRAMS

There are various TI-59 "emulators" (and also for TI-58/58C), available on the net both for Windows and for other platforms, which distort the real calculators by introducing absurd features (like 1000 program steps and 10000 registers (??) without setting any memory partition and/or equally imaginative instructions that have nothing to do with the factual reality of the TI world: from my point of view these programs (although graphically more attractive and more "user friendly" than the emulators treated in this manual) remain personal experiments of the authors and therefore I can safely call them **pseudo-emulators**.

Therefore, I think that a passage from these pseudo emulators to any TI-58C or TI-59 for those who have never known the real calculators will be traumatic and didactically fruitless. To demonstrate this thesis in the *PROG* folder I inserted a small program, called **59TEST.SOA**, and taken from TI-PPC Notes:

```
{ Test for *real* emulators!  From TI-PPC Notes V5 N4/5 P8 }
{ List all keycodes in step 478: press A to start          }

LBL A 7 OP 17 1 0 0 STO 00 9 . 2 0 0 7 6 0 8 6 9 STO 60 LBL A' 6 OP 17
SBR 04 75 7 OP 17 . 0 0 1 SUM 60 DSZ 00 A' R/S
```

which produces the output shown below.

```
478  00   0
478  01   1
478  02   2
478  03   3
478  04   4
478  05   5
478  06   6
478  07   7
478  08   8
478  09   9
478  10   E'
478  11   A
478  12   B
478  13   C
478  14   D
478  15   E
478  16   A'
478  17   B'
478  18   C'
478  19   D'
478  20  CSR
478  21  2ND
478  22  INV
478  23  LNX
478  24  CE
478  25  CLR
478  26  2ND
478  27  INV
478  28  LOG
478  29  CP
478  30  TAN
478  31  LRN
478  32  X:T
478  33  X²
```

```
478  34  √X
478  35  1/X
478  36  PGM
478  37  P/R
478  38  SIN
478  39  COS
478  40  IND
478  41  SST
478  42  STO
478  43  RCL
478  44  SUM
478  45  Y^X
478  46  INS
478  47  CMS
478  48  EXC
478  49  PRD
478  50  IXI
478  51  BST
478  52  EE
478  53   (
478  54   )
478  55   ÷
478  56  DEL
478  57  ENG
478  58  FIX
478  59  INT
478  60  DEG
478  61  GTO
478  62  PG*
478  63  EX*
478  64  PD*
478  65   x
478  66  PAU
478  67   EQ
```

```
478  68  NOP
478  69  OP
478  70  RAD
478  71  SBR
478  72  ST*
478  73  RC*
478  74  SM*
478  75   -
478  76  LBL
478  77   GE
478  78  Σ+
478  79   Xm
478  80  GRD
478  81  RST
478  82  HIR
478  83  GO*
478  84  OP*
478  85   +
478  86  STF
478  87  IFF
478  88  DMS
478  89   π
478  90  LST
478  91  R/S
478  92  RTN
478  93   .
478  94  +/-
478  95   =
478  96  WRT
478  97  DSZ
478  98  ADV
478  99  PRT
```

At the moment, **only** the Hrast Programmer emulator (which is based on the direct emulation of the ROM microcode) and mine, treated in this manual, are able to overcome this “pico-test”….

# 6. USER PROGRAMS MANAGEMENT

This chapter is dedicated to the management of user programs by the emulator.

## 6.1 PROGRAM LOADING

Generally you can load the programs in three ways:

1) by keyboard, entering LRN-mode and inputting the program;
2) by magnetic card, using the appropriate option in the Control menu. Magnetic cards are, obviously, virtual and recorded as files (in *SCHEDE59* or *SCHEDE52* or *SCHEDE60* folder, according to the model) as *SCHEDA.XXX* with *XXX* between 000 and 999. The program also requires the side of the card (side A or side B). For a correct loading of the card, it is necessary to respect what has been said in chapter VII of the TI-59 manual (partition of memory, decimals fixing, etc ...). In the same folder it's contained a file named **INDEX** that provides a list of the contents of the magnetic cards.
   (*Note*: file INDEX update is **not** automatic and is therefore user-care).

*SCHEDA.XXX* files of the TI-59 are formed by **two** blocks of **256** numeric codes (between 00 and 99) each, with the following meaning:

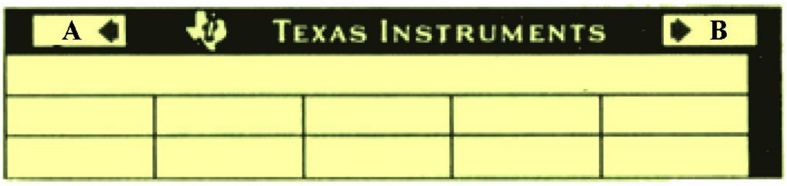
| | |
|---|---|
| code #1 | - protect memory (0=free,1=blocked) |
| code #2 | - program partition (in block of ten register) |
| code #3 | - number of saved |
| code #4 | - block checksum for hardware control: sum of the 240 codes (between #16 and #255) MOD 100 |
| code #5 | - data type on the card (10=program 11=data reg. dati or mixed program/data reg.) |
| code#6-#15 | - not used |
| code#16-#255 | - memory content relevant to the selected block |

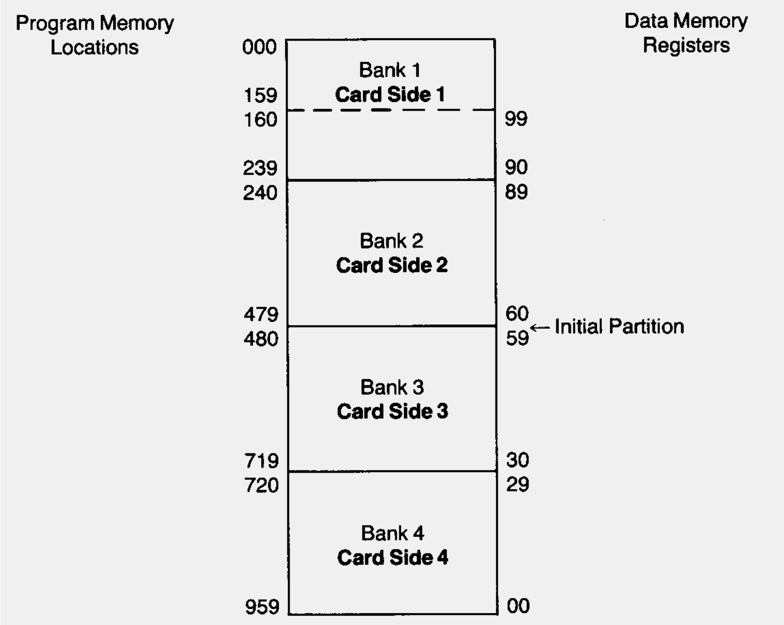
For SR-52 cards we have **two** blocks of **128** codes and for SR-60A two blocks of **512** codes.

3) by Import function (with the **F7** key) which allows you to import (and to manage) text files having **.SOA** extension. Note: **SOA** is the abbreviation for Italian "*Sistema Operativo Algebrico*"; **AOS** "*Algebraic Operating System*" in English.

## 6.2. SAVING ON CARD

● As in the real TI-59 you can save programs and/or data using magnetic cards: to do this just use the instructions *n 2nd Write* (directly) and *n INV 2nd Write* (in program mode): the emulator, as above, will require a card number ( 000-999) and the side (A or B). **n** is, as in Texas Instruments standards, from 1 to 4 (see picture below).

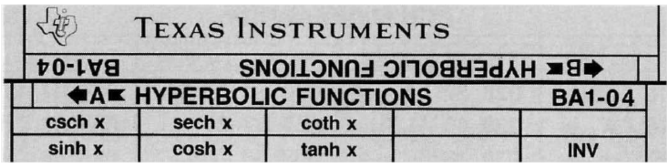




For the SR-52, on the other hand, the procedure listed in chapter IX of the SR-52 manual must be followed, note that there is an internal flag by which you read/write side A or side B of a magnetic card. It will be enough to do:

- For writing: CLR (reset the internal flag) INV 2nd Read (side A) INV 2nd Read (eventual side B)
- For reading: CLR (reset the internal flag) 2nd Read (side A) 2nd Read (eventual side B)

An HLT instruction after the first 2nd Read blocks the ongoing action and allows you to read the side B only .



For the SR-60a follow the instructions of chapter V of the manual: keep in mind that each side of the card can contain 480 steps of programs or 60 data registers.

Programs can be saved with RESET 2nd Write (first 480 steps) followed by more 2nd Write instructions, but with the program counter placed at the position (partition - 480). For example, the first 560 program steps can be saved as: RESET 2nd Write (first 480 steps) GTO 80 [560-480] 2nd Write.

A decidedly complicated procedure subject to errors . . . .

For reading, instead, just press RESET 2nd Read and the prompt mechanism, typical of the SR-60a, takes over.

The registers can be read/saved with the CLR 2nd Read/2nd Write ... 60 2nd Read/2nd Write sequence and so on 60 registers at a time.

SIDE B PROGRAM
TO WRITE – COVER WINDOW WITH A BLACK LABEL
TEXAS INSTRUMENTS
CALCULATOR PROGRAM CARD P/N 1030178-100
©1976
SIDE A PROGRAM

## 6.3 SOA FILES AND THEIR STRUCTURE

.SOA files are basically textfiles and can be edited using any text editor (the program attempts to recall the program indicated by IMPORT.INI file which, usually, is *EDIT* for MS-DOS/Windows or *NOTEPAD* for a 64-bit Windows version) respecting these simple rules:

A. Import will usually begin at step 000. It's possible, for the TI-59 only, to set the import step with an **[ORG: xxx]** where xxx is the step number required.
B. Recognized keywords are encoded in the various **CNVxx.CSV** files (where xx is 52, 53, 55, 55II, 56, 57, 57LCD, 58, 59, 62, 66, MBA, 60 - one for each emulated model of TI) in "**numeric code;mnemonic**" format (one per line). This file can be modified and/or expanded losing compatibility, however, with the standard. For example, the file **CNV59.CSV** for the TI-59 is the following (the columns layout is only for convenience):

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **00;0** | **25;CLR** | **50;IXI** | **75;-** | **00;00** | **25;25** | **50;50** | **75;75** |
| **01;1** | **26;2ND'** | **51;BST** | **76;LBL** | **01;01** | **26;26** | **51;51** | **76;76** |
| **02;2** | **27;INV'** | **52;EE** | **77;GE** | **02;02** | **27;27** | **52;52** | **77;77** |
| **03;3** | **28;LOG** | **53;(** | **78;STA** | **03;03** | **28;28** | **53;53** | **78;78** |
| **04;4** | **29;CP** | **54;)** | **79;AVG** | **04;04** | **29;29** | **54;54** | **79;79** |
| **05;5** | **30;TAN** | **55;/** | **80;GRD** | **05;05** | **30;30** | **55;55** | **80;80** |
| **06;6** | **31;LRN** | **56;DEL** | **81;RST** | **06;06** | **31;31** | **56;56** | **81;81** |
| **07;7** | **32;X:T** | **57;ENG** | **82;HIR** | **07;07** | **32;32** | **57;57** | **82;82** |
| **08;8** | **33;X^2** | **58;FIX** | **83;GO*** | **08;08** | **33;33** | **58;58** | **83;83** |
| **09;9** | **34;SQR** | **59;INT** | **84;OP*** | **09;09** | **34;34** | **59;59** | **84;84** |
| **10;E'** | **35;1/X** | **60;DEG** | **85;+** | **10;10** | **35;35** | **60;60** | **85;85** |
| **11;A** | **36;PGM** | **61;GTO** | **86;STF** | **11;11** | **36;36** | **61;61** | **86;86** |
| **12;B** | **37;P/R** | **62;PG*** | **87;IFF** | **12;12** | **37;37** | **62;62** | **87;87** |
| **13;C** | **38;SIN** | **63;EX*** | **88;DMS** | **13;13** | **38;38** | **63;63** | **88;88** |
| **14;D** | **39;COS** | **64;PD*** | **89;PI** | **14;14** | **39;39** | **64;64** | **89;89** |
| **15;E** | **40;IND** | **65;*** | **90;LST** | **15;15** | **40;40** | **65;65** | **90;90** |
| **16;A'** | **41;SST** | **66;PAU** | **91;R/S** | **16;16** | **41;41** | **66;66** | **91;91** |
| **17;B'** | **42;STO** | **67;EQ** | **92;RTN** | **17;17** | **42;42** | **67;67** | **92;92** |
| **18;C'** | **43;RCL** | **68;NOP** | **93;.** | **18;18** | **43;43** | **68;68** | **93;93** |
| **19;D'** | **44;SUM** | **69;OP** | **94;+/-** | **19;19** | **44;44** | **69;69** | **94;94** |
| **20;CLR'** | **45;Y^X** | **70;RAD** | **95;=** | **20;20** | **45;45** | **70;70** | **95;95** |
| **21;2ND** | **46;INS** | **71;SBR** | **96;WRT** | **21;21** | **46;46** | **71;71** | **96;96** |
| **22;INV** | **47;CMS** | **72;ST*** | **97;DSZ** | **22;22** | **47;47** | **72;72** | **97;97** |
| **23;LNX** | **48;EXC** | **73;RC*** | **98;ADV** | **23;23** | **48;48** | **73;73** | **98;98** |
| **24;CE** | **49;PRD** | **74;SM*** | **99;PRT** | **24;24** | **49;49** | **74;74** | **99;99** |

C. The separator used in the .SOA files is the *blank character*: absolute jump addresses should be written in groups of two. For example *GTO 528* has to be written as GTOb5b28. Even the numbers must be written digit by digit: -1.35 will be entered as –b1b.b3b5).

D. **[BEGIN]** signals the begin of program text.
E. **[DATA]** indicates the contents of the data registers in the format ***nn:content*** (one per line), and it's mandatory to place it **after** the program.
F. **[END]** ends the useful content of the program; everything else is considered comment and it is, of course, excluded from the translation.
G. You can insert comments within the program using the { character at beginning until to the end of line (you can terminate the comment for the sake of readability even with } although it is not mandatory). Initial comments are also used to build the program's help file - for consultation with the **Shift + F1** key.

The structure of a generic **.SOA** file is the following:

| .SOA file | Type | Remarks |
|---|---|---|
| **{ …………………….. }**<br>**{ …………………….. }**<br>etc. | Initial remarks: used to build the help file available with **Shift + F1** | *Optional* |
| **[ORG: xxx]** | Sets the initial step from which import begins. Usually this step is zero. | *Optional*: it has effect on a TI-59 only. |
| **[BEGIN]** | Idicates the beginning of the text of the program | *Optional:* it also signals the end of the initial comments. |
| **.SOA program text** | In the above format according to *CNNxx.CSV* file. It can be mixed with remarks - a remark begins with the { character to the end of line. | *Mandatory* → create IMPORT. OUT file |
| **[DATA]**<br>XX:valore - *repeated n times* | Registers to be loaded in the format *register number:value* | *Optional*: → create DATA. OUT file |
| **[END]** | | *Mandatory* only if it's not the last line of .SOA file |
| **Free text** | | *Optional*, it's not considered by F7 - Import: if exists it must be anyway preceeded by [END]. |

*Note*: The file **IMPORT.INI** is composed of two lines; in the first line it's indicated the name of the editor file for the 16-bit version (EDIT is proposed) while in the second one the file for 64-bit version (NOTEPAD is proposed). It is assumed that both files are executable from the folder \TI59.

## 6.4 .SOA FILES TRANSLATION

The translation, depending on the emulated calculator type, can be **normal** (each keyword and / or address and / or numerical data is translated as a program step) or **grouped** (for TI 57, TI 57 and TI

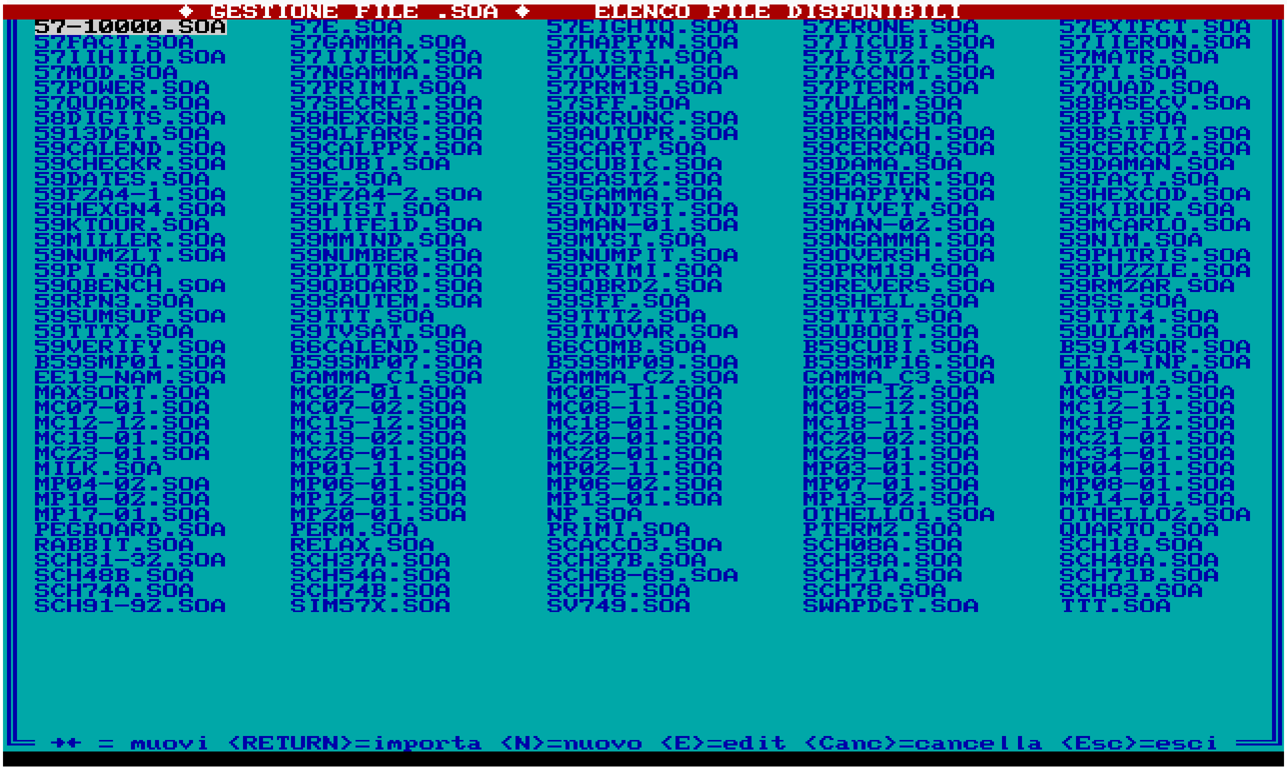
62 LCD - every composite instruction incorporates the following token). For example STO 05 in the TI 59 are two steps (42 and 05) while in the TI 57 is only one (32 5).

The .SOA files are generic, so it is possible (length permitting) to import files created, for example, for the TI-59 on the TI-66, unless they don't contain specific instructions for that model. For example, if you use in a program **Pgm 20** instruction, the corresponding .SOA file can be imported successfully only on the TI-58/58C/59 because **Pgm** is possible only on those models: so it will be convenient to use a significant name in these cases or to put a few final notes. The translation will still be tried, but will provide error.

Output will provide three files:

- *IMPORT.LOG* that contains four lines
  - the number of errors;
  - the name of the relevant help files;
  - a list of the .SOA files line numbers in which errors occurred (in any);
  - the initial step from which import begins. Usually this value is zero and can be set only for a TI-59 program.
- *IMPORT.OUT* that contains the .SOA file translated files into numerical codes;
- *DATA.OUT* that contains the values of registers to be loaded.

Pressing **F7** the content of the folder pointed to by the import path (which, let's remember, can be modified with the function key **f3**) is displayed:



```
              ♦ GESTIONE FILE .SOA ♦      ELENCO FILE DISPONIBILI
57-10000.SOA    57E.SOA         57EIGHTQ.SOA    57ERONE.SOA     57EXTFCT.SOA
57FACT.SOA      57GAMMA.SOA     57HAPPYN.SOA    57IICUBI.SOA    57IIERON.SOA
57IIHILO.SOA    57IIJEUX.SOA    57LIST1.SOA     57LIST2.SOA     57MATR.SOA
57MOD.SOA       57NGAMMA.SOA    57OVERSH.SOA    57PCCNOT.SOA    57PI.SOA
57POWER.SOA     57PRIMI.SOA     57PRM19.SOA     57PTERM.SOA     57QUAD.SOA
57QUADR.SOA     57SECRET.SOA    57SFF.SOA       57ULAM.SOA      58BASECV.SOA
58DIGITS.SOA    58HEXGN3.SOA    58NCRUNC.SOA    58PERM.SOA      58PI.SOA
5913DGT.SOA     59ALFARG.SOA    59AUTOPR.SOA    59BRANCH.SOA    59BSTFIT.SOA
59CALEND.SOA    59CALPPX.SOA    59CART.SOA      59CERCAQ.SOA    59CERCQ2.SOA
59CHECKR.SOA    59CUBI.SOA      59CUBIC.SOA     59DAMA.SOA      59DAMAN.SOA
59DATES.SOA     59E.SOA         59EAST2.SOA     59EASTER.SOA    59FACT.SOA
59FZA4-1.SOA    59FZA4-2.SOA    59GAMMA.SOA     59HAPPYN.SOA    59HEXCOD.SOA
59HEXGN4.SOA    59HIST.SOA      59INDTST.SOA    59JIVET.SOA     59KIBUR.SOA
59KTOUR.SOA     59LIFE1D.SOA    59MAN-01.SOA    59MAN-02.SOA    59MCARLO.SOA
59MILLER.SOA    59MMIND.SOA     59MYST.SOA      59NGAMMA.SOA    59NIM.SOA
59NUM2LT.SOA    59NUMBER.SOA    59NUMPIT.SOA    59OVERSH.SOA    59PHIRIS.SOA
59PI.SOA        59PLOT60.SOA    59PRIMI.SOA     59PRM19.SOA     59PUZZLE.SOA
59QBENCH.SOA    59QBOARD.SOA    59QBRD2.SOA     59REVERS.SOA    59RM2AR.SOA
59RPN3.SOA      59SAUTEM.SOA    59SFF.SOA       59SHELL.SOA     59SS.SOA
59SUMSUP.SOA    59TTT.SOA       59TTT2.SOA      59TTT3.SOA      59TTT4.SOA
59TTTX.SOA      59TVSAT.SOA     59TWOVAR.SOA    59UBOOT.SOA     59ULAM.SOA
59VERIFY.SOA    66CALEND.SOA    66COMB.SOA      B59CUBI.SOA     B59I4SQR.SOA
B59SMP01.SOA    B59SMP07.SOA    B59SMP09.SOA    B59SMP16.SOA    EE19-INP.SOA
EE19-NAM.SOA    GAMMA_C1.SOA    GAMMA_C2.SOA    GAMMA_C3.SOA    INDNUM.SOA
MAXSORT.SOA     MC02-01.SOA     MC05-11.SOA     MC05-12.SOA     MC05-13.SOA
MC07-01.SOA     MC07-02.SOA     MC08-11.SOA     MC08-12.SOA     MC12-11.SOA
MC12-12.SOA     MC15-12.SOA     MC18-01.SOA     MC18-11.SOA     MC18-12.SOA
MC19-01.SOA     MC19-02.SOA     MC20-01.SOA     MC20-02.SOA     MC21-01.SOA
MC23-01.SOA     MC26-01.SOA     MC28-01.SOA     MC29-01.SOA     MC34-01.SOA
MILK.SOA        MP01-11.SOA     MP02-11.SOA     MP03-01.SOA     MP04-01.SOA
MP04-02.SOA     MP06-01.SOA     MP06-02.SOA     MP07-01.SOA     MP08-01.SOA
MP10-02.SOA     MP12-01.SOA     MP13-01.SOA     MP13-02.SOA     MP14-01.SOA
MP17-01.SOA     MP20-01.SOA     NP.SOA          OTHELLO1.SOA    OTHELLO2.SOA
PEGBOARD.SOA    PERM.SOA        PRIMI.SOA       PTERM2.SOA      QUARTO.SOA
RABBIT.SOA      RELAX.SOA       SCACCO3.SOA     SCH08A.SOA      SCH18.SOA
SCH31-32.SOA    SCH37A.SOA      SCH37B.SOA      SCH38A.SOA      SCH48A.SOA
SCH48B.SOA      SCH54A.SOA      SCH68-69.SOA    SCH71A.SOA      SCH71B.SOA
SCH74A.SOA      SCH74B.SOA      SCH76.SOA       SCH78.SOA       SCH83.SOA
SCH91-92.SOA    SIM57X.SOA      SV749.SOA       SWAPDGT.SOA     TTT.SOA

 →← = muovi <RETURN>=importa <N>=nuovo <E>=edit <Canc>=cancella <Esc>=esci
```

The following operations are possible:

- <**cursor keys**> keys allow the selection of a file;
- <**E**> edit an existing file;

- <**N**> creates a new file (name is given without the .SOA extension, it is automatically created) that can be edited using DOS/Windows EDIT or NOTEPAD for 64-bit Operating systems;
- <**Canc**> clears a file (with confirmation);
- <**Esc**> exit without making imports (the emulator report an error anyway).

In *PROG\ITMAG* folder there are several files marked *MPnn-xx.DOC* and *MCnn-xx.SOA*: these files come from programs published in two Italian magazines "**Micro & Personal Computer"** (1979-1983) and "**MC Microcomputer"** (1981-1984): *nn* is the number of the magazine, while *xx* is the progressive number of the programs published within the same number (if xx is below 10 programs are for the TI-58/58C/59, if greater than or equal to 10 programs are for the TI-57).

## 6.5. WHAT TO DO IF IMPORT GIVES AN ERROR

Usually import errors are due to text typing mistakes and, for the moment, the diagnostics is limited to providing the number of total errors and line numbers in which the errors occurred. IMPORT immediately asks if you want to correct the SOA file in question: if you respond negatively the Import program terminates, otherwise import is tried again after your corrections.
A quick and more "scientific" way to see where and what the errors are is, once re-entered in the emulator, to run OS-Shell with **F9**, and execute **EDIT** (or **NOTEPAD**) **PROG\IMPORT. OUT**: this last file consists of lines containing all numeric codes; in case of error, however, the incorrect part is kept followed by the string "<======": then, it will be enough to correct it in the corresponding. SOA file.

For example, if the file is *59FACT.SOA*

```
{ Factorial import test: press n A to calculate n! }
{ It contains an intentional error !!!             }

LBL A STO 01 RCP 01 * DSZ 01 00 04 1 = R/S
```

the corresponding IMPORT.OUT (the red part is an explanation remark) will be

```
76
11
42
01
RCP  <====== not found in CNV59.CSV file, then it's highlightened
01
65
97
01
00
04
01
95
91
```

Clearly in this case, "RCP" should become "RCL" (with code 43) in the file FACT59.SOA. You can edit directly the file from the emulator using the **F7** Key facility (see § 16 note 3).

*N.B .: If you abort the import with <Esc>, the emulator situation will not be changed.*

## 6.6 STANDALONE USAGE OF 'IMPORT' PROGRAM

You can use *IMPORT.EXE* (or *IMPORT-64.EXE*) as a standalone program, with its own switches: the complete syntax is è

**IMPORT file_name /type /CROM /V /PATH=**path **/ENG**

where

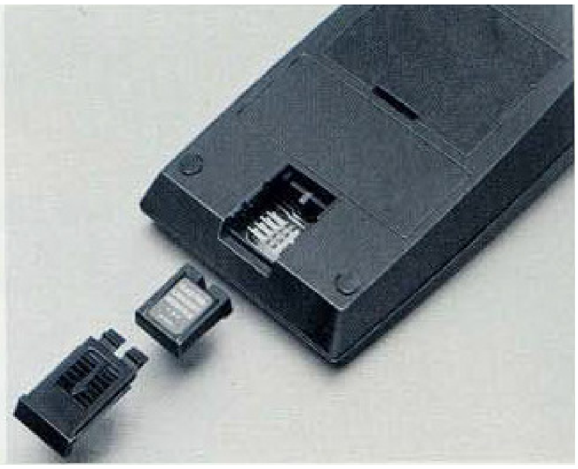
**"filename"** is the name of the file to be converted (without .SOA extension)
**"type"** can be /57 /58 /59 /66 /57LCD /55 /52 /56 /55II /62 /53 /MBA /60 /KC /SE
**/CROM** allows the use of CROM Emulator (solo per /59, /58C e /58)
**/V** displays the translation in progress
**/PATH=** allows you to define a new import path. If absent, **PROG** is used.
**/ENG** displays messages in English

**Important:** Switches (if used) must follow the syntax order shown above.

# 7 TI-59/58/58C EMULATORS

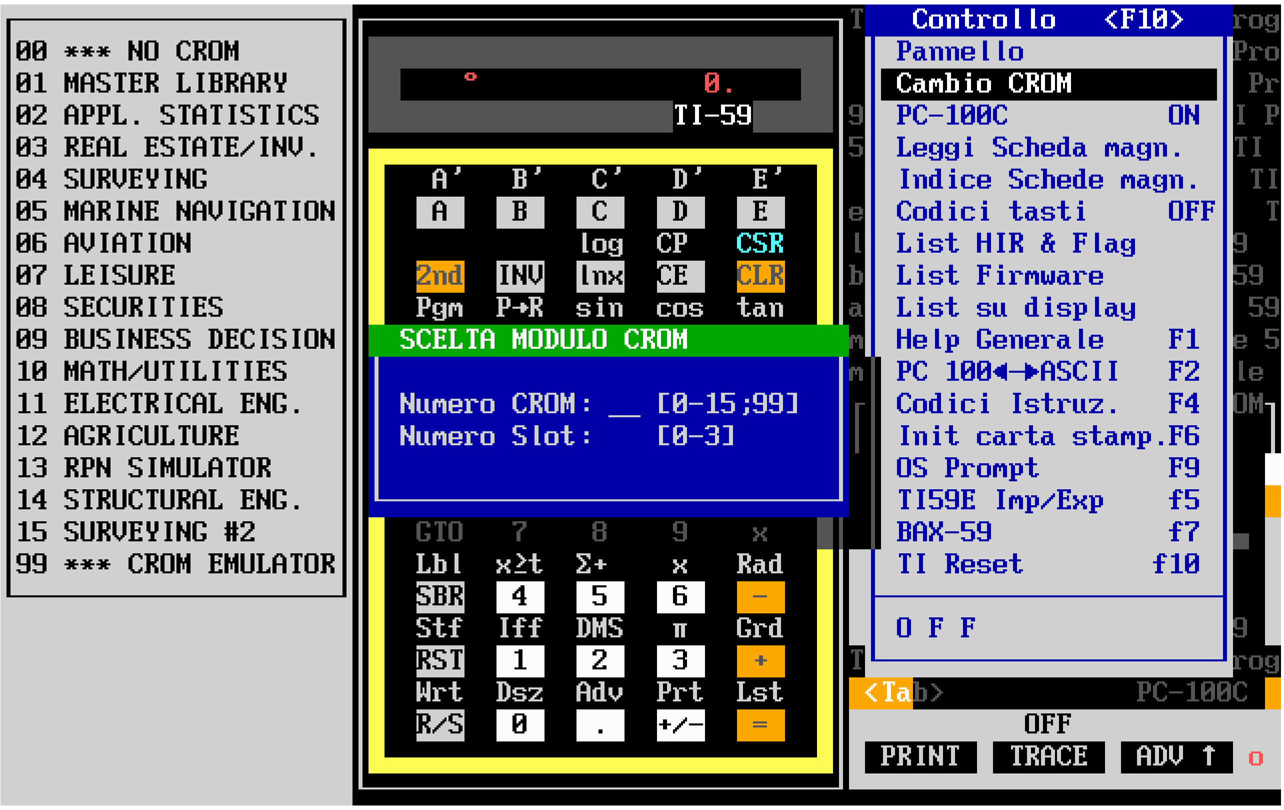
The hardware of the TI-59/58/58C is more complex than that of the other models, especially for the ge-style of the peripherals and their operating methods, namely the CROM modules ("Module Selector" and "Crom Emulator")) and the extensive management of the PC-100C printer, which are the subject of the rest of the chapter. Statistical and conversio funztion are implemented using the internal firmware, written in AOS language and 512 bytes long – included the "infamous" instruction HIR 20: this scheme was derived from the one used in the SR-56.

## 7.1. CROM MODULES MANAGEMENT



The TI-59 (and the two minor "sisters" 58 and 58C) have a slot on the back of the calculator itself which allows the insertion of **one** CROM module that can be selected from those produced by Texas Instruments or those produced by third parties ("custom" modules). In order to get more modules online (such as the HP-41C/CV that has four slots available) a third company has developed a device called "**Module Selector**" (available in two versions - one *manual* and one *automatic*) that allows to have online **up to four** CROM modules.

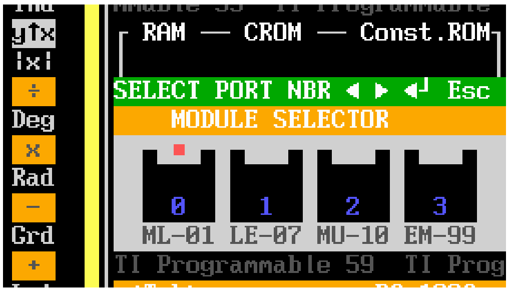
The emulator implements both *automatic* and *manual* versions. The first is used in the same way as the original through the sequence **X 77 Op 04 Op 06** which allows you to select the **X** slot (0 to 3). This sequence can be used by keyboard or by program.



```
00 *** NO CROM
01 MASTER LIBRARY
02 APPL. STATISTICS
03 REAL ESTATE/INV.
04 SURVEYING
05 MARINE NAVIGATION
06 AVIATION
07 LEISURE
08 SECURITIES
09 BUSINESS DECISION
10 MATH/UTILITIES
11 ELECTRICAL ENG.
12 AGRICULTURE
13 RPN SIMULATOR
14 STRUCTURAL ENG.
15 SURVEYING #2
99 *** CROM EMULATOR
```

```
SCELTA MODULO CROM

Numero CROM: __ [0-15;99]
Numero Slot:    [0-3]
```

Instead the second mode is activated with **Shift** + **F6** which displays the following small menu (in green):



with the left and right cursor keys you select the port by confirming with **Return** or quitting the operation with **Esc**.

---

Another device connected to the CROM modules is available with the emulator: the "**CROM Emulator**" which allows you to test the operativity of a CROM module prior its final commercialization. Such device (to my knowledge) has **never** been commercialized and it was probably only available at Texas Instruments; the emulator allows you to use it by following these steps:

1) Load the special module **EMULCROM** (number 99) and activate the slot that houses it.
2) With **F12** key recall program to be tested
3) Program is now available with the usual commands preceeded, obviously, with Pgm 01.

All this means, given the greater capacity of a module CROM than the RAM of the TI-59, you can run long programs up to 5000 steps using contemporaneously up to 100 data registers (60 for 58 / 58C). This method has certainly been used for the realization and the testing of "*Structural Engineering*" (or "*Civil Engineering*") module that has two programs that exceed 1000 steps.

Contrary to programs contained in the CROM "normal" modules, it’s possibile using TRACE when you run programs in this mode.

An example of a program executable in this mode only is 'SCACCO3.SOA' that plays a final chess game and can be found, as usually, in the '*PROG*' folder.

## 7.2 THE 'BUILDER' UTILITY PROGRAM

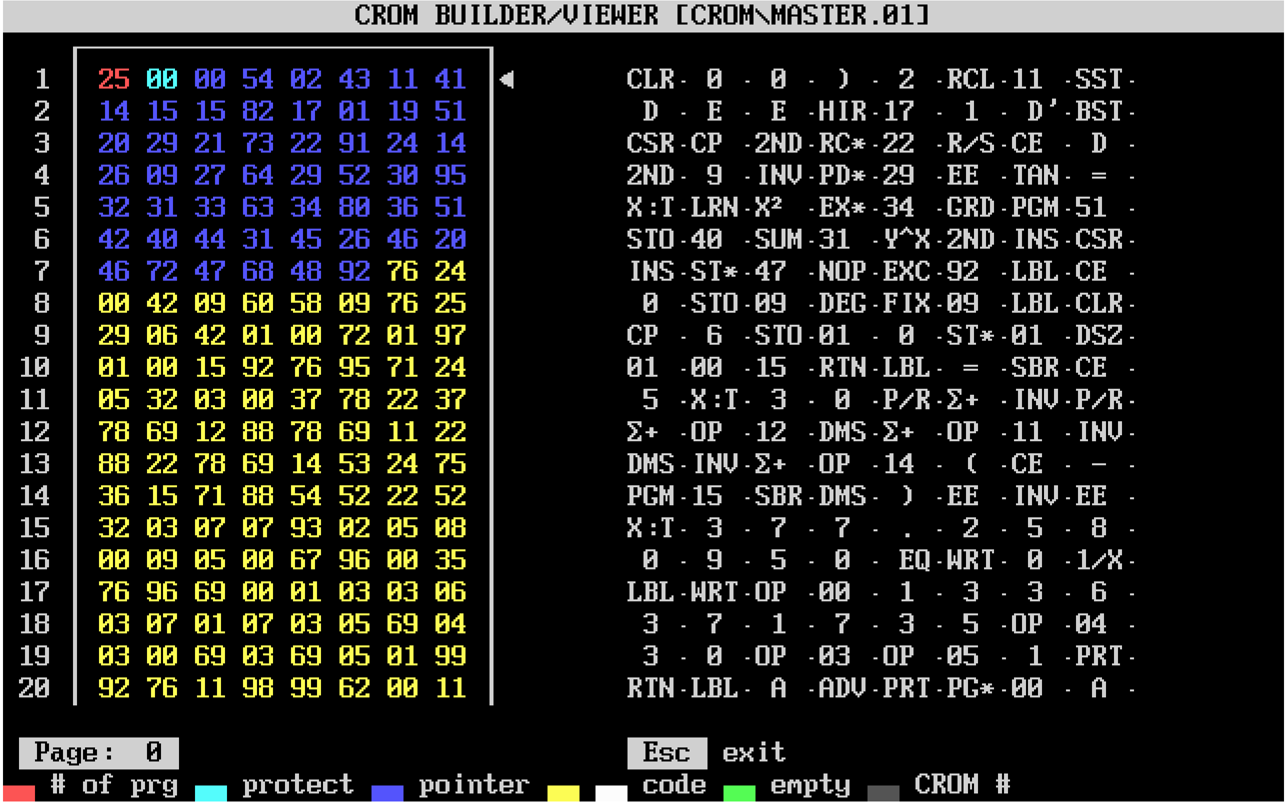
A utility related to the management of CROM modules is **BUILDER** that allows you to create a custom CROM module to be used later with the emulator or to view an existing one. For using BUILDER, just go to OS Shell with the **F9** key and invoke the utility: the complete command line is

**BUILDER [/?] [CROM_file]**

The **/?** switch displays, as usual, the modes of use, while following a file name, the program enters in "view" mode and allows you to see the content of the module. Each color represents a different part, as highlighted in the 25th row of the screen. For example with you get the screen shown on the next page. The cursor, Pag ↑ and Pag ↓ keys allow the cursor ◄ to be moved, while **Esc** allows exit.

Instead, invoking **BUILDER** without options, you enter 'create' mode. The number of programs to be inserted with their respective lengths, the number to be assigned to the module and the name of the file containing the numerical codes (one per line) are requested: obviously any anomaly is reported. Once the input data has been accepted, BUILDER automatically switches to 'view' mode and the **F10** key allows the module to be saved. (Be careful not to overwrite a standard form!!)

**Note:** It is recommended that you use the CROM folder for all BUILDER operations.



```
                CROM BUILDER/VIEWER [CROM\MASTER.01]

 1  25 00 00 54 02 43 11 41 ◄   CLR- 0 - 0 - ) - 2 -RCL-11 -SST-
 2  14 15 15 82 17 01 19 51      D - E - E -HIR-17 - 1 - D'-BST-
 3  20 29 21 73 22 91 24 14     CSR-CP -2ND-RC*-22 -R/S-CE - D -
 4  26 09 27 64 29 52 30 95     2ND- 9 -INV-PD*-29 -EE -TAN- = -
 5  32 31 33 63 34 80 36 51     X:T-LRN-X² -EX*-34 -GRD-PGM-51 -
 6  42 40 44 31 45 26 46 20     STO-40 -SUM-31 -Y^X-2ND-INS-CSR-
 7  46 72 47 68 48 92 76 24     INS-ST*-47 -NOP-EXC-92 -LBL-CE -
 8  00 42 09 60 58 09 76 25      0 -STO-09 -DEG-FIX-09 -LBL-CLR-
 9  29 06 42 01 00 72 01 97     CP - 6 -STO-01 - 0 -ST*-01 -DSZ-
10  01 00 15 92 76 95 71 24     01 -00 -15 -RTN-LBL- = -SBR-CE -
11  05 32 03 00 37 78 22 37      5 -X:T- 3 - 0 -P/R-Σ+ -INV-P/R-
12  78 69 12 88 78 69 11 22     Σ+ -OP -12 -DMS-Σ+ -OP -11 -INV-
13  88 22 78 69 14 53 24 75     DMS-INV-Σ+ -OP -14 - ( -CE - - -
14  36 15 71 88 54 52 22 52     PGM-15 -SBR-DMS- ) -EE -INV-EE -
15  32 03 07 07 93 02 05 08     X:T- 3 - 7 - 7 - . - 2 - 5 - 8 -
16  00 09 05 00 67 96 00 35      0 - 9 - 5 - 0 - EQ-WRT- 0 -1/X-
17  76 96 69 00 01 03 03 06     LBL-WRT-OP -00 - 1 - 3 - 3 - 6 -
18  03 07 01 07 03 05 69 04      3 - 7 - 1 - 7 - 3 - 5 -OP -04 -
19  03 00 69 03 69 05 01 99      3 - 0 -OP -03 -OP -05 - 1 -PRT-
20  92 76 11 98 99 62 00 11     RTN-LBL- A -ADV-PRT-PG*-00 - A -

 Page:  0                       Esc  exit
🟥 # of prg 🟦 protect 🟦 pointer 🟨 ⬜ code 🟩 empty ⬛ CROM #
```

*BUILDER in'view' mode*

## 7.3 "GRAPHICS" MODE ON PC-100C PRINTER

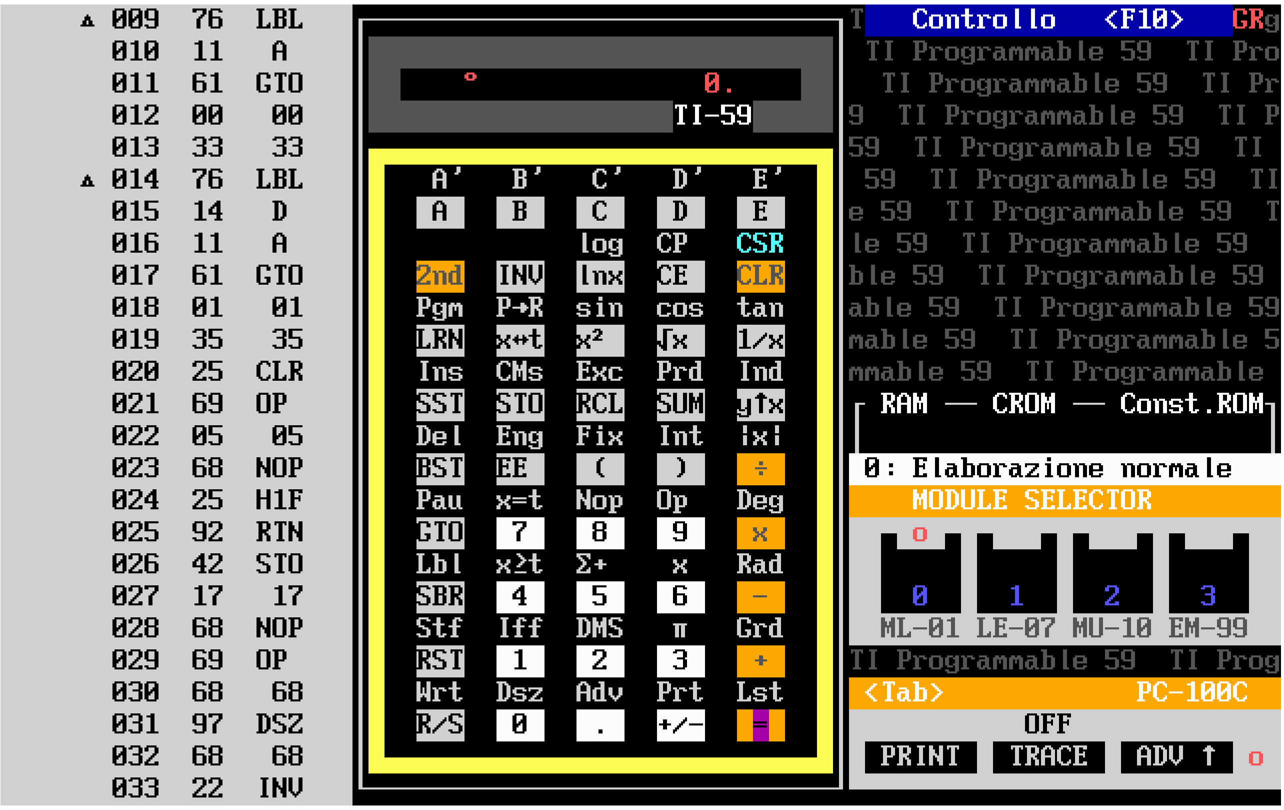
The PC-100C printer (and earlier models A and B) are thermal printers for plain text and commands for their use are widely explained in the manuals of Texas Instruments. In fact you can use the PC-100 in "graphics mode" with a programming techniques that use the generation of hexadecimal codes by using sequence of "unorthodox" keystrokes. The result is that the hex code **h25**, entered in RAM in a step multiple of eight, can control the single point of your printer. "PLOT60" program, appeared on the bulletin *TI PPC Notes*, can in this way generate graphs of functions.

Using the emulator is decidedly less difficult than the procedure to be followed on a real TI-59 and consists simply. using the **F8** key (*synthetic codes*), of introducing in a program step multiple of 8 (as in the real machine) the h25 code. In this listing it appears as "**H1F**". After running the program, it appears the normal result and pressing **F11**, however, appears the graphic result that is clearly more "squashed" than text (see next page).

*Note:* The **F8** key also works on **TI57.EXE**, **TI66.EXE** and **SR-52.EXE** to generate synthetic codes typical of these models, according to the following table:

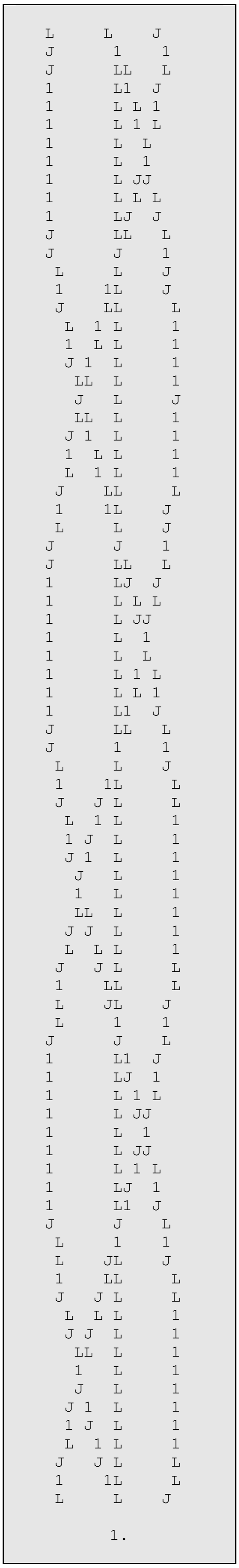
| | |
|---|---|
| **TI57** | 11, 12, 16, 17, 21, 31, 0A .. 0F |
| **TI66** | 0A .. 0F |
| **SR52** | 31, 62, 63, 64, 71, 72, 73, 74, 82, 84, 92 |

the meaning of which is explained in the various issues of 52Notes and TI-PPC Notes.



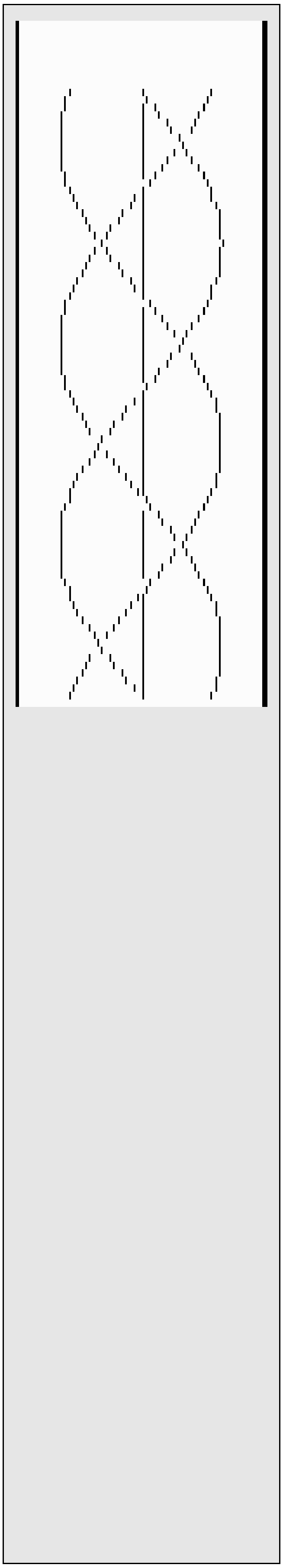
```
▲ 009  76  LBL
  010  11   A
  011  61  GTO
  012  00   00
  013  33   33
▲ 014  76  LBL
  015  14   D
  016  11   A
  017  61  GTO
  018  01   01
  019  35   35
  020  25  CLR
  021  69  OP
  022  05   05
  023  68  NOP
  024  25  H1F
  025  92  RTN
  026  42  STO
  027  17   17
  028  68  NOP
  029  69  OP
  030  68   68
  031  97  DSZ
  032  68   68
  033  22  INV
```

*Note the acronym GR in the upper right which identifies the type of TI-59 emulator*



```
L     L    J
J      1    1
J      LL   L
1      L1  J
1      L L 1
1      L 1 L
1      L  L
1      L  1
1      L JJ
1      L L L
1      LJ  J
J      LL   L
J      J    1
 L     L    J
 1    1L    J
 J    LL     L
  L  1 L     1
  1  L L     1
  J 1  L     1
   LL  L     1
   J   L     J
   LL  L     1
  J 1  L     1
  1  L L     1
  L  1 L     1
 J    LL     L
 1    1L    J
 L     L    J
J      J    1
J      LL   L
1      LJ  J
1      L L L
1      L JJ
1      L  1
1      L  L
1      L 1 L
1      L L 1
1      L1  J
J      LL   L
J      1    1
 L     L    J
 1    1L     L
 J   J L     L
  L  1 L     1
  1 J  L     1
  J 1  L     1
   J   L     1
   1   L     1
   LL  L     1
  J J  L     1
  L  L L     1
 J   J L     L
 1    LL     L
 L    JL    J
 L     1    1
J      J    L
1      L1  J
1      LJ  1
1      L 1 L
1      L JJ
1      L  1
1      L JJ
1      L 1 L
1      LJ  1
1      L1  J
J      J    L
 L     1    1
 L    JL    J
 1    LL     L
 J   J L     L
  L  L L     1
  J J  L     1
   LL  L     1
   1   L     1
   J   L     1
  J 1  L     1
  1 J  L     1
  L  1 L     1
 J   J L     L
 1    1L     L
 L     L    J

      1.
```

*Normal*



*Graphic*

*PLOT60 example execution: with F11 you can obtain the right screen*

## 7.4 TI-59 AND “FAST MODE”

One of the most used techniques in the TI58/59 programs for the obtaining of higher execution speed was the so-called '*fast mode*'. Clearly the emulator speed on current PC is superior to those of the original TI so that this technique is useless. It’s possible to run programs written for the fast mode on the emulator too, ignoring inputs required for this mode. For example, the benchmark program appeared on *www.hpmuseum.org* web site to compare the speed of various models of calculators (including the TI) using the "Eight Queens" problem has been so written for the TI fast mode:

```
000  R/S
001  CLR STO 10
004  8 STO 11 STO 01
009  1 STO 00
012  RCL 00  X:T  RCL 11 EQ 070
020  CLR X:T
022  OP  20
024  RCL 11 ST* 00
028  1  SUM  10
031  RCL 00 STO 09
035  INV DSZ 09 012
040  RC* 00 - RC* 09 = EQ 061
049  |X| + RCL 09 - RCL 00 =
057  INV X=T 035
061  DSZ IND 00 028
066  DSZ 00 061
070  RCL 10 PAUSE RST
...
145  LBL A
147  INV FIX
149  1 0  OP 17
153  CMS . 1
156  √X X² 1/X
159  STF
```

*Instructions:*

*To use the program, press A and you will see a blinking **10.** on the display. Press 7 and then EE and wait for the completion of the calculation. You can’t stop execution if not turning off the calculator.*

To use it on the emulator simply replace the part of the program (in red) that begins at step 154 with the sequence **GTO 00 01** so the initialization sequence starts the fast mode at step 001 and to run the program just starting with A .

## 7.5 PC-100C/PC-200↔ASCII CONVERSION UTILITY.

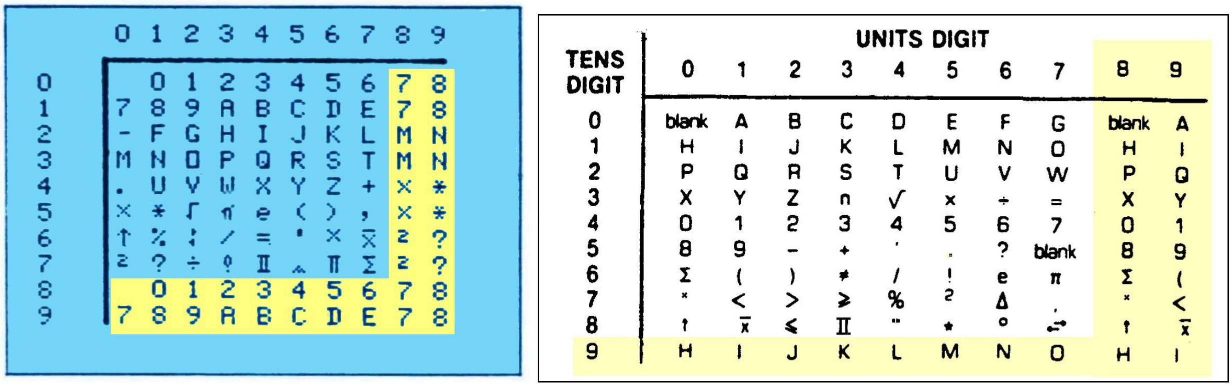
This program allows the conversion between the alphanumeric codes of the PC-100C/PC printers 200 and the ASCII codes. The menu is extremely simple and consists of 3 options:

1) input a sequence of a maximum of 20 characters (16 for the PC-200) to obtain their respective alphanumeric codes divided by quarter-printing (op 01 ... op 04)

2) input 20 (16) PC-100C numerical codes (PC-200) to obtain their respective ASCII codes

3) exit and return to the emulator.

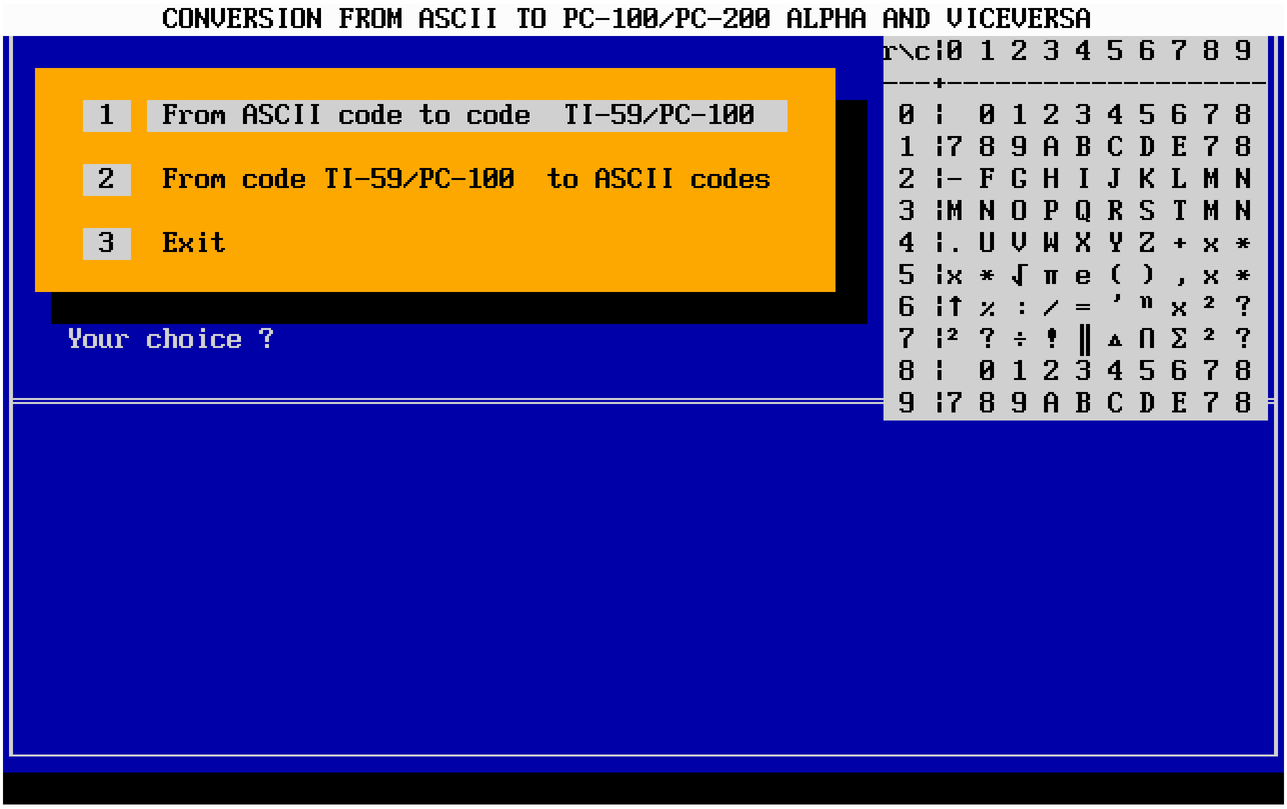
Note: the utility manages the extended table of the Alfa codes.



|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 |  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | 7 | 8 | 9 | A | B | C | D | E | 7 | 8 |
| 2 | - | F | G | H | I | J | K | L | M | N |
| 3 | M | N | O | P | Q | R | S | T | M | N |
| 4 | . | U | V | W | X | Y | Z | + | × | * |
| 5 | × | * | √ | π | e | ( | ) | , | × | * |
| 6 | ↑ | % | ⁝ | / | = | ' | × | x̄ | ² | ? |
| 7 | ² | ? | ÷ | ! | ‖ | ▲ | Π | Σ | ² | ? |
| 8 |  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 9 | 7 | 8 | 9 | A | B | C | D | E | 7 | 8 |

| TENS DIGIT | UNITS DIGIT 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | blank | A | B | C | D | E | F | G | blank | A |
| 1 | H | I | J | K | L | M | N | O | H | I |
| 2 | P | Q | R | S | T | U | V | W | P | Q |
| 3 | X | Y | Z | n | √ | × | ÷ | = | X | Y |
| 4 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 |
| 5 | 8 | 9 | − | + | ' | . | ? | blank | 8 | 9 |
| 6 | Σ | ( | ) | ≠ | / | ! | e | π | Σ | ( |
| 7 | × | < | > | ≥ | % | ² | Δ | . | × | < |
| 8 | ↑ | x̄ | ≤ | Π | " | * | ° | ⇄ | ↑ | x̄ |
| 9 | H | I | J | K | L | M | N | O | H | I |

*Alpha extended codes tabels (PC-100C on the left and PC-200 on the right)*

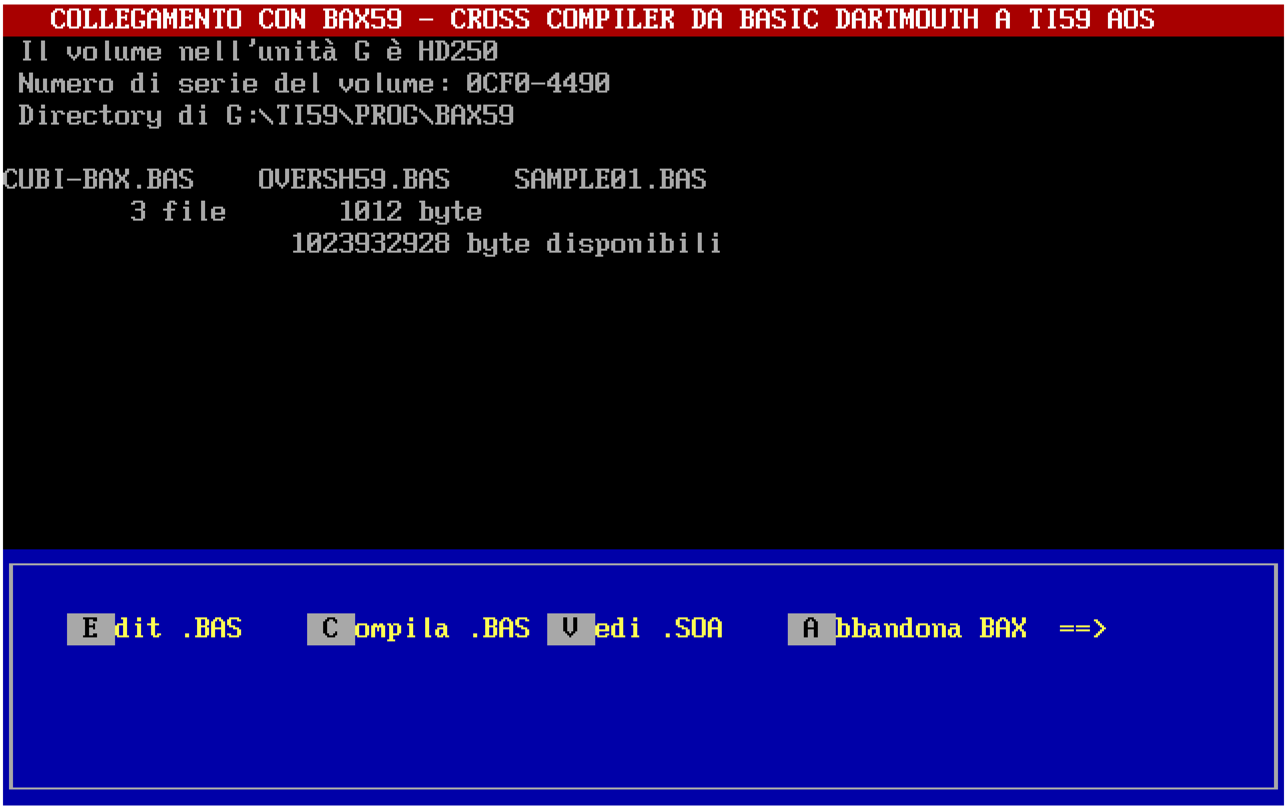


```
         CONVERSION FROM ASCII TO PC-100/PC-200 ALPHA AND VICEVERSA
                                                             r\c¦0 1 2 3 4 5 6 7 8 9
                                                             ---+-------------------
    1   From ASCII code to code  TI-59/PC-100                 0 ¦  0 1 2 3 4 5 6 7 8
                                                              1 ¦7 8 9 A B C D E 7 8
    2   From code TI-59/PC-100  to ASCII codes                2 ¦- F G H I J K L M N
                                                              3 ¦M N O P Q R S T M N
    3   Exit                                                  4 ¦. U V W X Y Z + x *
                                                              5 ¦x * √ π e ( ) , x *
                                                              6 ¦↑ % : / = ' " x ² ?
  Your choice ?                                               7 ¦² ? ÷ ! ‖ ▲ Π Σ ² ?
                                                              8 ¦  0 1 2 3 4 5 6 7 8
                                                              9 ¦7 8 9 A B C D E 7 8
```

*Conversion utility screen*

## 8. USING BAX-59 COMPILER.

You can now be used directly from within *TI59.EXE (TI59-64.EXE)* the **BAX59** cross-compiler that allows you to "translate" in the TI-59 A.O.S. programs written in BASIC Dartmouth. The **Shift F7** key leads to a minimal interface that allows you to organize your work:



```
    COLLEGAMENTO CON BAX59 - CROSS COMPILER DA BASIC DARTMOUTH A TI59 AOS
 Il volume nell'unità G è HD250
 Numero di serie del volume: 0CF0-4490
 Directory di G:\TI59\PROG\BAX59

CUBI-BAX.BAS     OVERSH59.BAS     SAMPLE01.BAS
        3 file         1012 byte
                  1023932928 byte disponibili
```

At the top there is the list of .BAS files contained in the *PROG/BAX59* folder; at this point you can use the following keys:

- **'E'** to create and / or edit a BASIC file;
- **'C'** which allows compiling: if this process is correct at the end the .SOA file will be found in the PROG directory, ready to be imported with the usual Import **F7** key;
- **'V'** allow you to view .SOA files;
- **'A'** quits and return to the emulation program.

It's well to pay attention that programs written in Basic Dartmouth are not fully compatible with, for example, GWBASIC and therefore may not be possible to make any preliminary test to see if the program is working properly and that, anyway, **BAX59 does not detect all possible errors**.

Overall features of the Basic Dartmouth used by **BAX59** are the following:

1) **real** and **integer** variables (with % suffix: they are equivalent to the TI-59 flags) and limited support to **string** variables;
2) every program line is numbered;
3) a single statement for every line;
4) mandatory LET statement use for assignment;
5) final END instruction mandatory.
6) arrays can be at most bidimensional – it's even accepted the implicit DIM;

7) DEF FN functions can also be multi-line with, at most, two parameters;
8) IF…THEN (o IF … GOTO) statement only jumps to a certain line;
9) FOR..TO..STEP..NEXT and WHILE..WEND loops;
10) seventeen predefined functions: ABS, ATN, INT, EXP, LOG, SIN, COS, TAN, RND, SQR, SGN, COT, TAB, LGD, ASC, CHR$, FRAC and the PI constant. RND function uses Master Library module; (COT - cotangent, LGD - decimal logarithm and FRAC - fractional part aren't GWBASIC/QBasic standard functions)
11) PRINT statement is adapted to the characteristics of the PC-100 printer: so ASCII codes aren’t supported but only TI PC-100 codes;
12) READ & DATA statements; GOSUB..RETURN for subroutines;
13) OPTION statement is **not standard** and it's used to support special features of the TI-59.

In the **BAX59** manual (*BAXMAN59.PDF*) you will find all the features of BASIC language used. The keyword of the language is listed below:

## BAX-59 BASIC LANGUAGE KEYWORD

```
ABS(
ASC(
ATN(
CHANGE
CHR$(
CLEAR
COS(
COT(
DATA
DEF FN
DIM (DIMENSION)
END
EXP(
FNEND
FOR
FRAC(
GOSUB
GOTO
IF
INPUT
INT(
LET
LGT(
LOG(
NEXT
NOT
ON
OPTION
PI
PRINT
PRINT USING
RANDOMIZE (RANDOM)
READ
REM
RESTORE
RETURN
RND(
SGN(
SIN(
SQR(
STEP (BY)
STOP
TAB(
TAN(
THEN
TO
WEND
WHILE
```

Finally an example of translation (*SAMPLE01.BAS* file taken from BAX-.59 distribution) on the calculation of the Greatest Common Divisor (G.C.D. - M.C.D. in Italian):

| BASIC Dartmouth | TI-59 AOS |
|---|---|
| `10 PRINT "A", "B", "C", "GCD"` | `LBL E' 13 STO 13` |
| `15 FOR I = 1 TO 3` | `1 SUM 13 6 0 ST* 13` |
| `20 READ  A, B, C` | `1 SUM 13 9 0 ST* 13` |
| `30 LET X = A` | `1 SUM 13 1 2 0 ST* 13` |
| `40 LET Y = B` | `1 SUM 13 3 8 4 5 6 ST* 13` |
| `50 GOSUB 200` | `1 SUM 13 6 4 8 7 2 ST* 13` |
| `60 LET X = G` | `1 SUM 13 9 8 7 6 5 ST* 13` |
| `70 LET Y = C` | ` 13 STO 13` |
| `80 GOSUB 200` | `RTN` |
| `90 PRINT A, B, C, G` | `LBL PRT 1 9 STO 01 RTN` |
| `100 NEXT I` | `LBL ADV ( ( ( LOG INT + 1) / 2 + . 5) INT *` |
| `105 GOTO 999` | `2 ) RTN` |
| `110 DATA 60, 90, 120` | `{BASIC line #10}` |
| `120 DATA 38456, 64872, 98765` | `1 3 OP 01 OP 05` |
| `200 LET Q = INT(X/Y)` | `1 4 OP 01 OP 05` |
| `210 LET R = X - Q*Y` | `1 5 OP 01 OP 05` |
| `220 IF R = 0 THEN 300` | `2 2 1 5 1 6 OP 01 OP 05` |
| `230 LET X = Y` | `{BASIC line #15}` |
| `240 LET Y = R` | `1 STO 04` |
| `250 GOTO 200` | `{BASIC line #20}` |
| `300 LET G = Y` | `LBL A` |

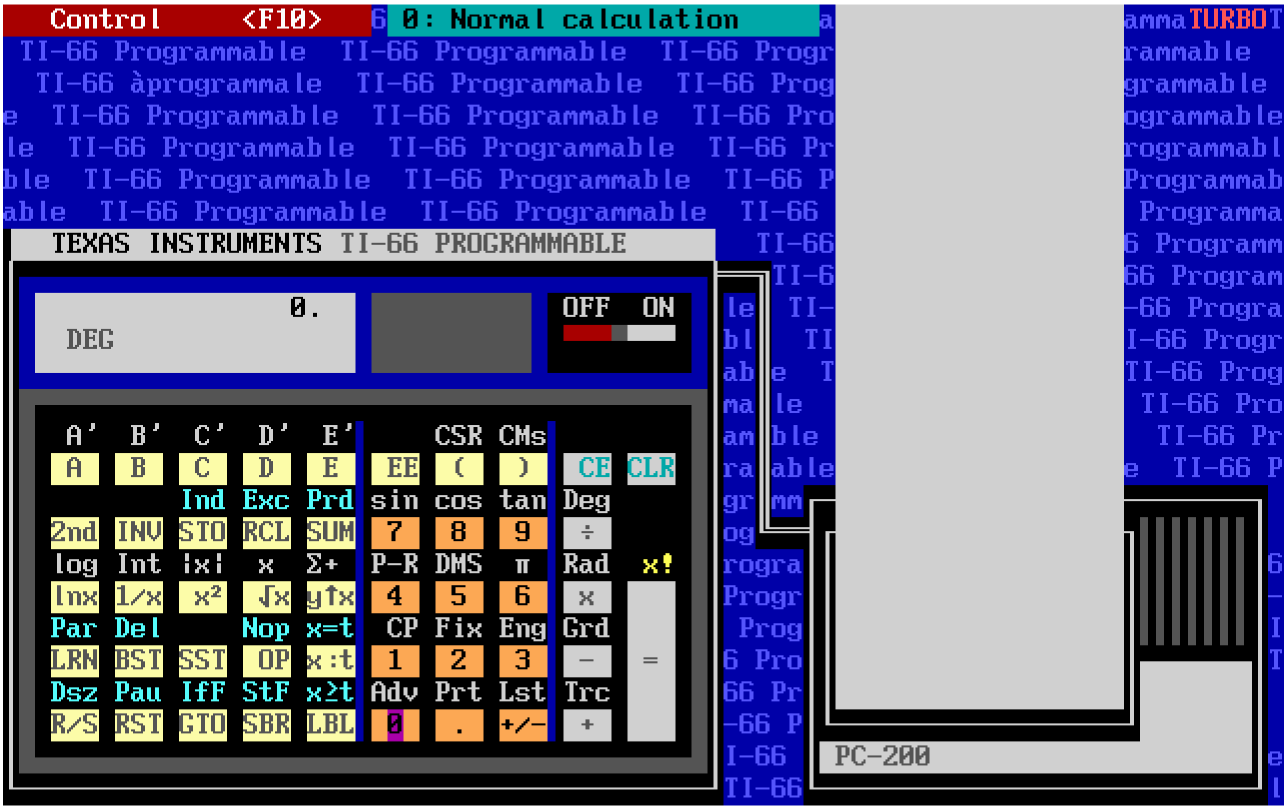
| | |
|---|---|
| 310 RETURN<br>999 END | 1 SUM 13 RC* 13 STO 05<br>1 SUM 13 RC* 13 STO 06<br>1 SUM 13 RC* 13 STO 07<br>{BASIC line #30}<br>RCL 05 STO 08<br>{BASIC line #40}<br>RCL 06 STO 09<br>{BASIC line #50}<br>C<br>{BASIC line #60}<br>( RCL 10 ) STO 08<br>{BASIC line #70}<br>RCL 07 STO 09<br>{BASIC line #80}<br>C<br>{BASIC line #90}<br>RCL 05 PRT<br>RCL 06 PRT<br>RCL 07 PRT<br>( RCL 10 ) PRT<br>{BASIC line #100}<br>LBL B<br>OP 24 RCL 04 X:T 3 GE A<br>{BASIC line #105}<br>GTO E<br>{BASIC line #110}<br>NOP<br>{BASIC line #120}<br>NOP<br>{BASIC line #200}<br>LBL C<br>( ( RCL 08 / RCL 09 ) INT ) STO 11<br>{BASIC line #210}<br>( RCL 08 - RCL 11 * RCL 09 ) STO 12<br>{BASIC line #220}<br>0 X:T RCL 12 EQ D<br>{BASIC line #230}<br>RCL 09 STO 08<br>{BASIC line #240}<br>( RCL 12 ) STO 09<br>{BASIC line #250}<br>GTO C<br>{BASIC line #300}<br>LBL D<br>RCL 09 STO 10<br>{BASIC line #310}<br>RTN<br>{BASIC line #999}<br>LBL E<br>R/S |

At the end of the compilation phase it is supplied the correct sequence of keys to start the program. You may also notice that each BASIC line is commented in the SOA translated file and at the beginning it's inserted an **initialization sequence** (in **green**) and a **utility library** (in **blue**).

**Note:** Another "high level" language has been implemented also for the TI-57 (**LIMACE-57**): for its features see § 10.2.

## 9. TI-66 EMULATOR

The TI-66 emulator also faithfully follows the original machine, including abbreviated addressing and synthetic codes 0A-0F, with the addition of the factorial function, implemented as 2nd =. An exception is also the management of the constant memory which also retains the fixing of the decimals and the angular mode, which is not present in the original machine.
The real TI-66 doesn’t have the capability to save a program on an external support (magnetic card or tape): to overcome this fact, CRAM modules have been implemented (as on the TI-88) which allow you to have a "mass memory" also on this machine (see next paragraph).


Control <F10>
0: Normal calculation
TURBO
TI-66 Programmable
TEXAS INSTRUMENTS TI-66 PROGRAMMABLE
0.
DEG
OFF ON
A' B' C' D' E' CSR CMs
A B C D E EE ( ) CE CLR
Ind Exc Prd sin cos tan Deg
2nd INV STO RCL SUM 7 8 9 ÷
log Int |x| x Σ+ P-R DMS π Rad x!
lnx 1/x x² √x y↑x 4 5 6 ×
Par Del Nop x=t CP Fix Eng Grd
LRN BST SST OP x:t 1 2 3 −
Dsz Pau IfF StF x≥t Adv Prt Lst Trc =
R/S RST GTO SBR LBL 0 . +/- +
PC-200

*TI-66.EXE (16 bit version - English)*

## 9.1 CRAM MODULES MANAGEMENT

As mentioned above, to make up for the lack in the TI-66 of a permanent storage system like magnetic cards or tape drives, in the emulator has been implemented (as suggested at the time - in 1984 - in an article in TI PPC Notes) the management of so-called CRAM modules that allow you to have more programs available without having to retype them. So you can "save" the situation of memory in a "CRAM module", which is assigned a number.

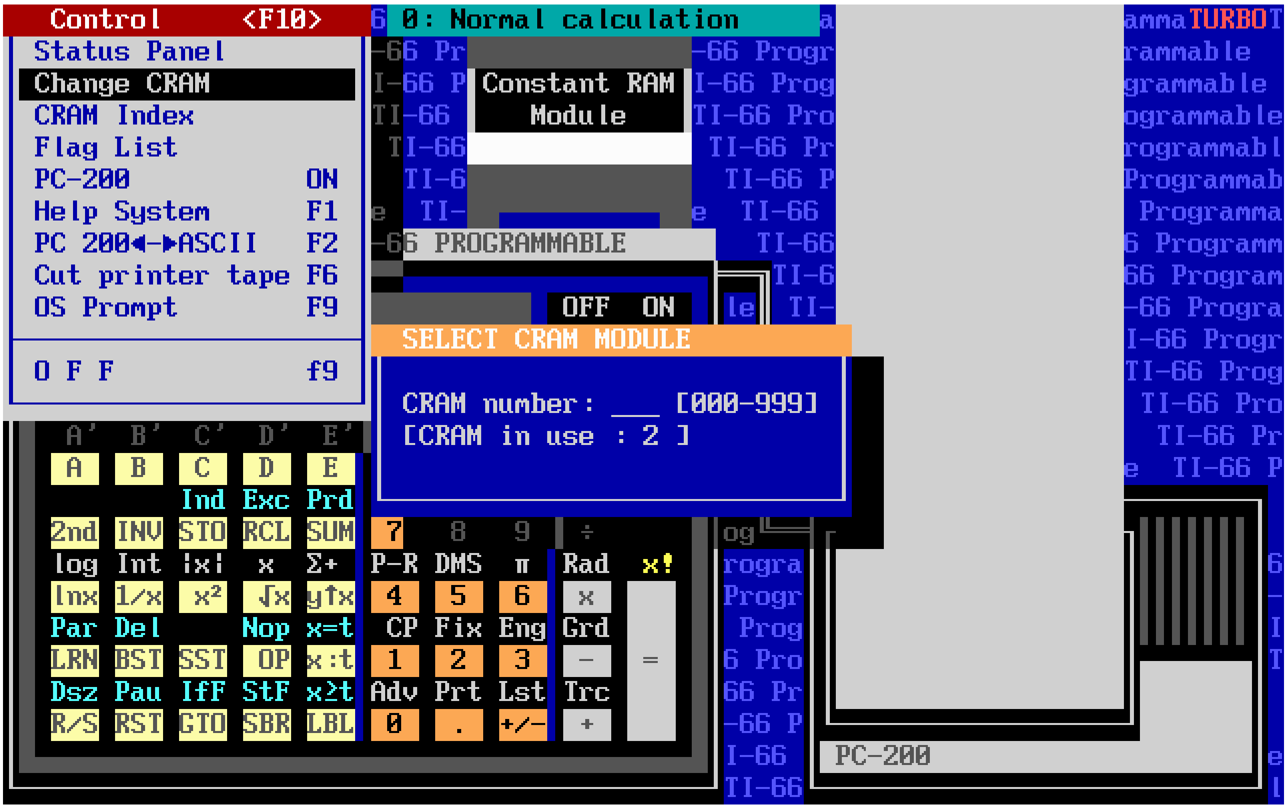
*TI66.EXE* remembers the last CRAM module used (which it’s in memory when the emulator starts) and, via the control menu - **F10 – Change CRAM**, you can choose the new CRAM to load. If the required CRAM does not exist it’s “created”, from scratch, empty otherwise the requested module is loaded after saving the current one.

In CRAM folder are contained the following files:

- **CRAM.xxx:** contains the source code, with xxx ranging from 000 to 999
- **INDEX:** is the list of CRAMs present in the folder. Its update must be managed by the user.

In a CRAM module, the first line contains the current partition followed by the program memory (512 positions – one keycode for line). By convention, an “empty” CRAM module is identified as 000: it is recommended not to use it for other purposes.

*Note*: passing from one CRAM to another, the fixing of the decimals and the angular mode are preserved, but not the partition of memory.



SELECT CRAM MODULE

```
CRAM number: ___ [000-999]
[CRAM in use : 2 ]
```

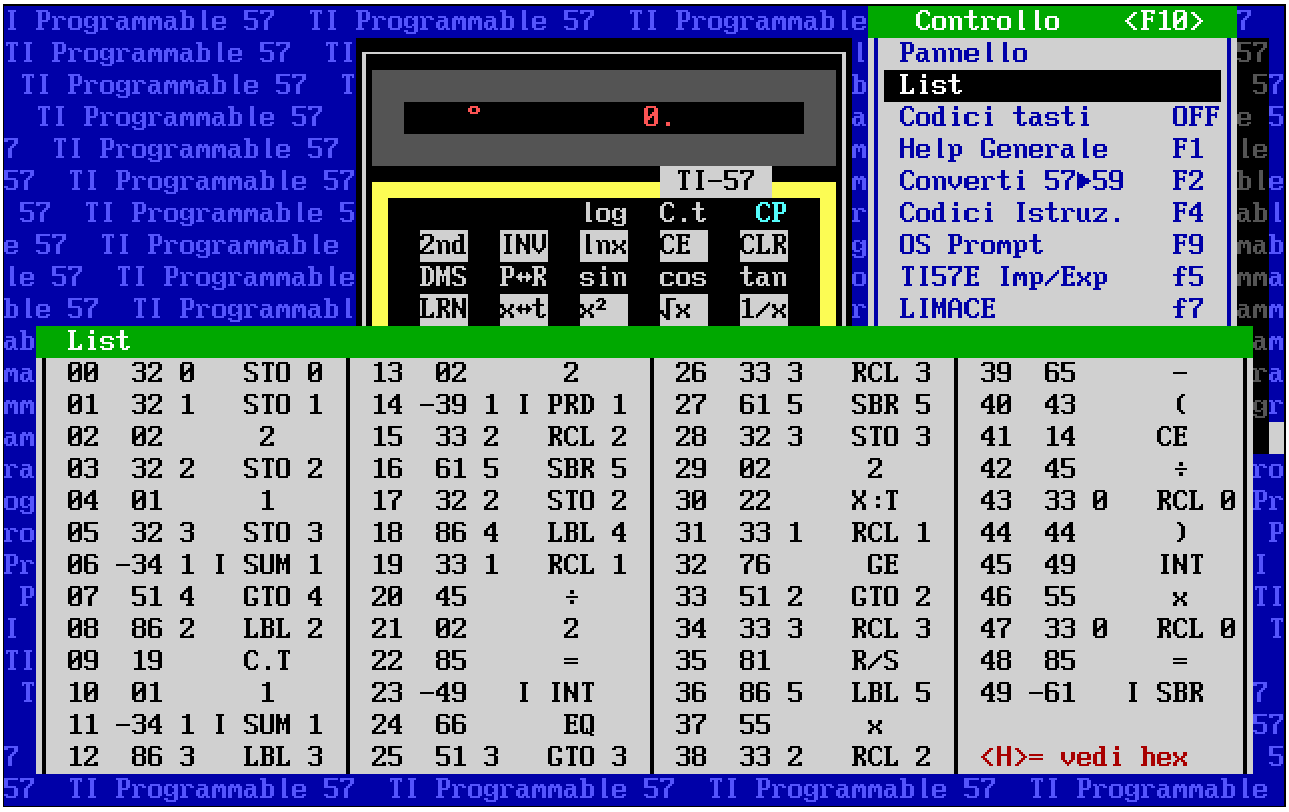
*CRAM module selection*

Finally, please note that the ALPHA PC200↔ASCII codes conversion is also active for the PC-200 printer (see § 7.4).

## 10. TI-57 EMULATOR

The TI-57 emulator has the same functionality as that TI-58/58C/59 emulators: differences concern obviously the hardware of the machines. For example, also if the printer is missing, it's possible to have the program listing using the **Control** (Controllo) menu (see picture in the next page) either with LRN-mode keycodes either with internal hexadecimal format of the TI-57.

It's also possible to recall directly the "57TO59" utility (see § 10) which programs are loaded/saved in *TI57* folder: the format is different from the text .SOA files used by the **F7- Import** function.



```
List
00  32 0    STO 0 | 13  02         2 | 26  33 3  RCL 3 | 39  65          -
01  32 1    STO 1 | 14 -39 1 I PRD 1 | 27  61 5  SBR 5 | 40  43          (
02  02        2   | 15  33 2   RCL 2 | 28  32 3  STO 3 | 41  14         CE
03  32 2    STO 2 | 16  61 5   SBR 5 | 29  02     2    | 42  45          ÷
04  01        1   | 17  32 2   STO 2 | 30  22    X:T   | 43  33 0    RCL 0
05  32 3    STO 3 | 18  86 4   LBL 4 | 31  33 1  RCL 1 | 44  44          )
06 -34 1 I  SUM 1 | 19  33 1   RCL 1 | 32  76     GE   | 45  49        INT
07  51 4    GTO 4 | 20  45         ÷ | 33  51 2  GTO 2 | 46  55          x
08  86 2    LBL 2 | 21  02         2 | 34  33 3  RCL 3 | 47  33 0    RCL 0
09  19      C.T   | 22  85         = | 35  81    R/S   | 48  85          =
10  01        1   | 23 -49     I INT | 36  86 5  LBL 5 | 49 -61      I SBR
11 -34 1 I  SUM 1 | 24  66        EQ | 37  55     x    |
12  86 3    LBL 3 | 25  51 3   GTO 3 | 38  33 2  RCL 2 | <H>= vedi hex
```

*Listing of the TI-57 program actually in memory*

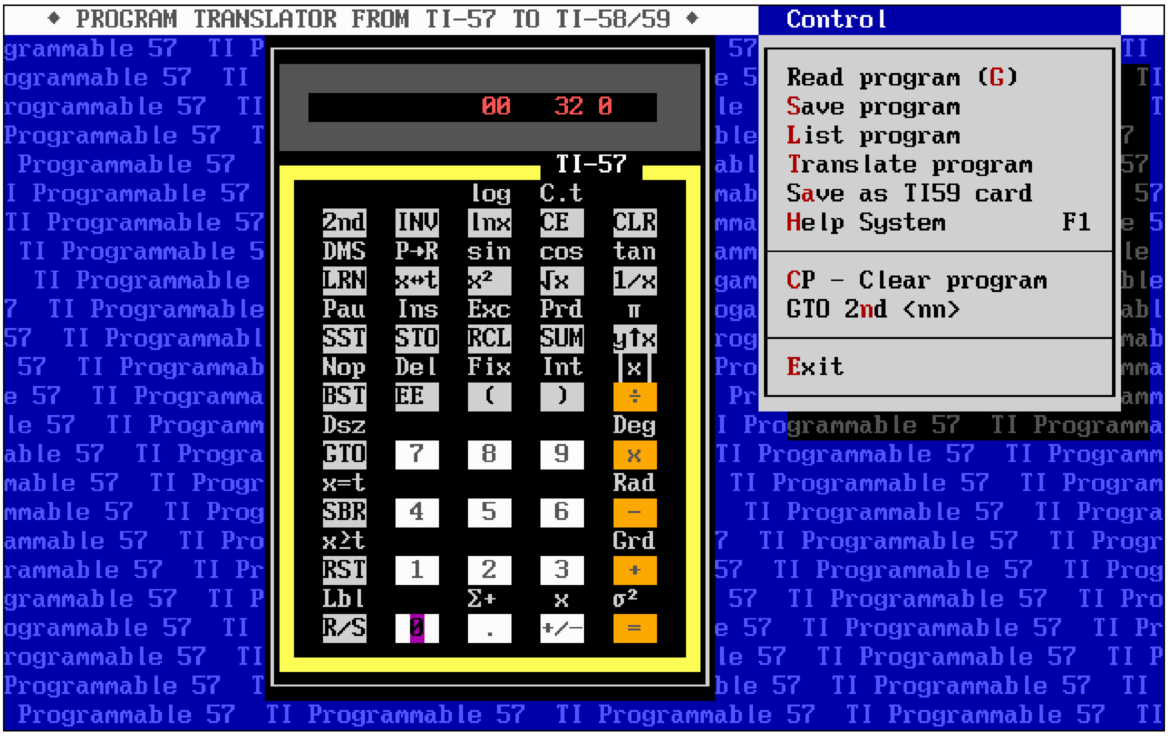
## 10.1 THE "57TO59" UTILITY PROGRAM

**57TO59** allows you to "translate" a TI-57 program in its corresponding TI-58/58C/59: this conversion allows you to replicate the behavior of the TI-57 when using the t-register and the statistical instructions which, as it is known, have a different registers' use in the two models.

The available commands in **Control** [Controllo] menu are the following:

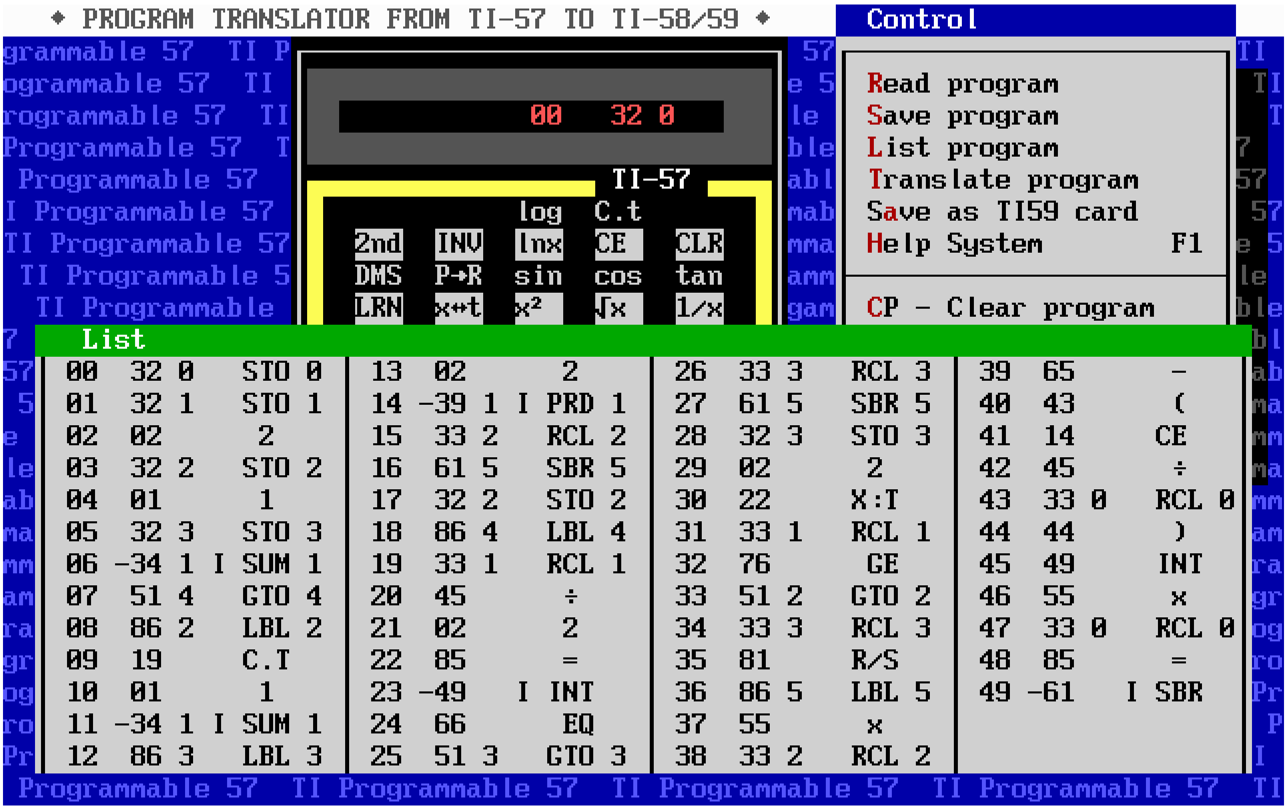
- Read program: load program to be translated.
- Save program: save program in memory.
- List: list program in memory.
- Translate program: translate program in memory to TI-59 format.
- Save as magnetic card (TI-59): saves program in memory to TI-59 magnetic card format (in *SCHEDE59* folder).
- H: show main help.
- CP (Clear program): clear program in memory.
- GTO 2nd <nn>: go to step <nn> (00 to 49).

- Exit: quit program.



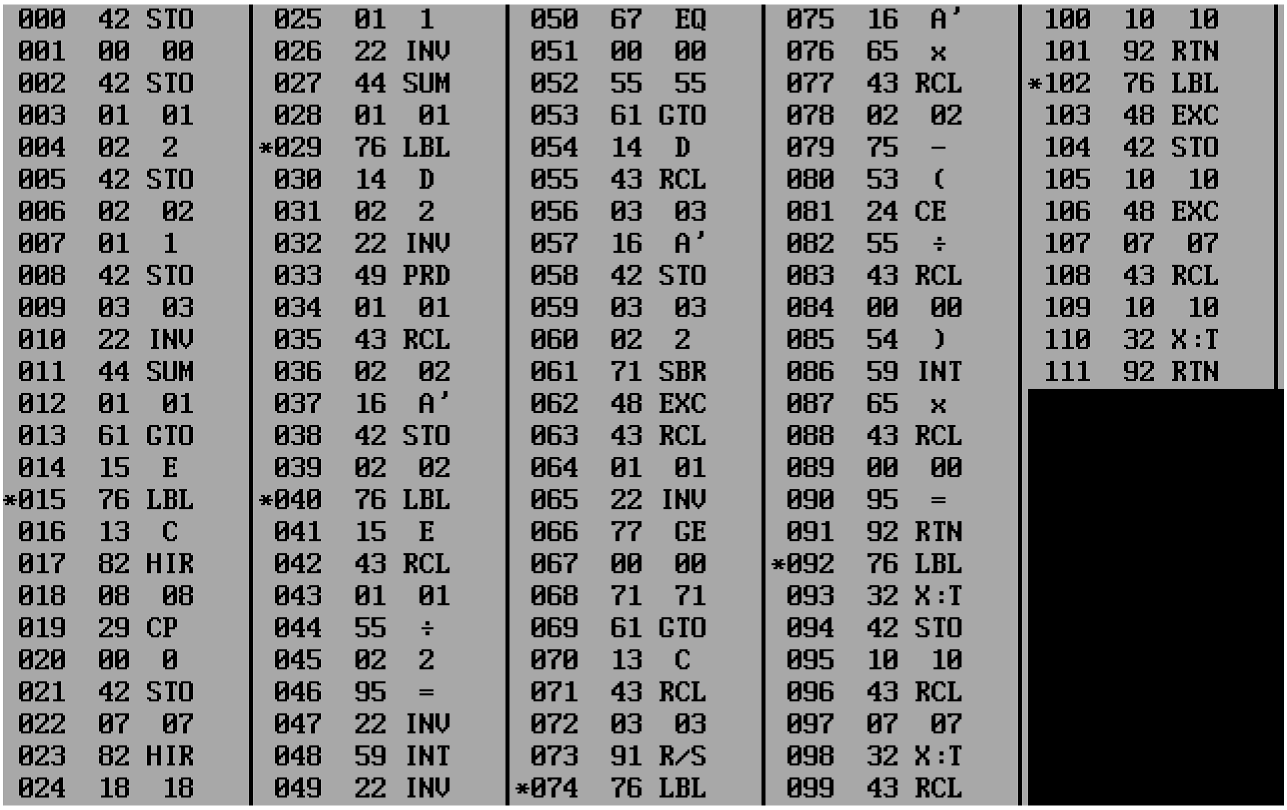
*57TO59.EXE is running…..*

For example, once loaded either directly or with the **G** key the following program (taken from *List* magazine of February 1985 and which implements the primality test according to Fermat), its listing is obtained with **L**:



```
 00   32 0    STO 0 | 13   02         2 | 26   33 3   RCL 3 | 39   65         -
 01   32 1    STO 1 | 14 -39 1 I PRD 1 | 27   61 5   SBR 5 | 40   43         (
 02   02        2   | 15   33 2   RCL 2 | 28   32 3   STO 3 | 41   14        CE
 03   32 2    STO 2 | 16   61 5   SBR 5 | 29   02        2  | 42   45         ÷
 04   01        1   | 17   32 2   STO 2 | 30   22      X:T  | 43   33 0   RCL 0
 05   32 3    STO 3 | 18   86 4   LBL 4 | 31   33 1   RCL 1 | 44   44         )
 06 -34 1 I SUM 1   | 19   33 1   RCL 1 | 32   76       GE  | 45   49       INT
 07   51 4    GTO 4 | 20   45         ÷ | 33   51 2   GTO 2 | 46   55         x
 08   86 2    LBL 2 | 21   02         2 | 34   33 3   RCL 3 | 47   33 0   RCL 0
 09   19      C.T   | 22   85         = | 35   81      R/S  | 48   85         =
 10   01        1   | 23 -49     I INT | 36   86 5   LBL 5 | 49 -61     I SBR
 11 -34 1 I SUM 1   | 24   66        EQ | 37   55        x  |
 12   86 3    LBL 3 | 25   51 3   GTO 3 | 38   33 2   RCL 2 |
```

while with **T**, I get the translation into TI-59 AOS:



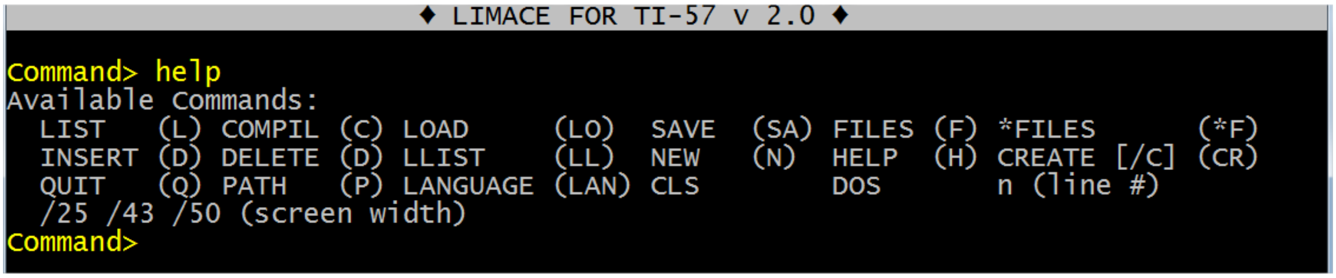
```
 000  42 STO   025  01  1    050  67  EQ   075  16  A'   100  10  10
 001  00  00   026  22 INV   051  00  00   076  65  ×    101  92 RTN
 002  42 STO   027  44 SUM   052  55  55   077  43 RCL  *102  76 LBL
 003  01  01   028  01  01   053  61 GTO   078  02  02   103  48 EXC
 004  02  2   *029  76 LBL   054  14  D    079  75  −    104  42 STO
 005  42 STO   030  14  D    055  43 RCL   080  53  (    105  10  10
 006  02  02   031  02  2    056  03  03   081  24 CE    106  48 EXC
 007  01  1    032  22 INV   057  16  A'   082  55  ÷    107  07  07
 008  42 STO   033  49 PRD   058  42 STO   083  43 RCL   108  43 RCL
 009  03  03   034  01  01   059  03  03   084  00  00   109  10  10
 010  22 INV   035  43 RCL   060  02  2    085  54  )    110  32 X:T
 011  44 SUM   036  02  02   061  71 SBR   086  59 INT   111  92 RTN
 012  01  01   037  16  A'   062  48 EXC   087  65  ×
 013  61 GTO   038  42 STO   063  43 RCL   088  43 RCL
 014  15  E    039  02  02   064  01  01   089  00  00
*015  76 LBL  *040  76 LBL   065  22 INV   090  95  =
 016  13  C    041  15  E    066  77  GE   091  92 RTN
 017  82 HIR   042  43 RCL   067  00  00  *092  76 LBL
 018  08  08   043  01  01   068  71  71   093  32 X:T
 019  29 CP    044  55  ÷    069  61 GTO   094  42 STO
 020  00  0    045  02  2    070  13  C    095  10  10
 021  42 STO   046  95  =    071  43 RCL   096  43 RCL
 022  07  07   047  22 INV   072  03  03   097  07  07
 023  82 HIR   048  59 INT   073  91 R/S   098  32 X:T
 024  18  18   049  22 INV  *074  76 LBL   099  43 RCL
```

*Note:* the ‘X:T’ and ‘Exc’ routines are added by the translator for the management of the T-register which on the TI-57 coincides with the memory #7.

***Note*****:** The list and the relative translation are stored in two text files (57List.Soa and 57Traduz.txt) located the '57TO59' folder.

## 10.2 THE ‘LIMACE’ LANGUAGE.

As on the TI-59, a high-level 'language' called LIMACE is also available on the TI-57, born from an idea of the French magazine '*L'Ordinateur Individuel*'. **Shift F7** calls up the programming environment that allows you (using a text user interface) to create, modify and compile a LIMACE program in the TI-57 AOS.

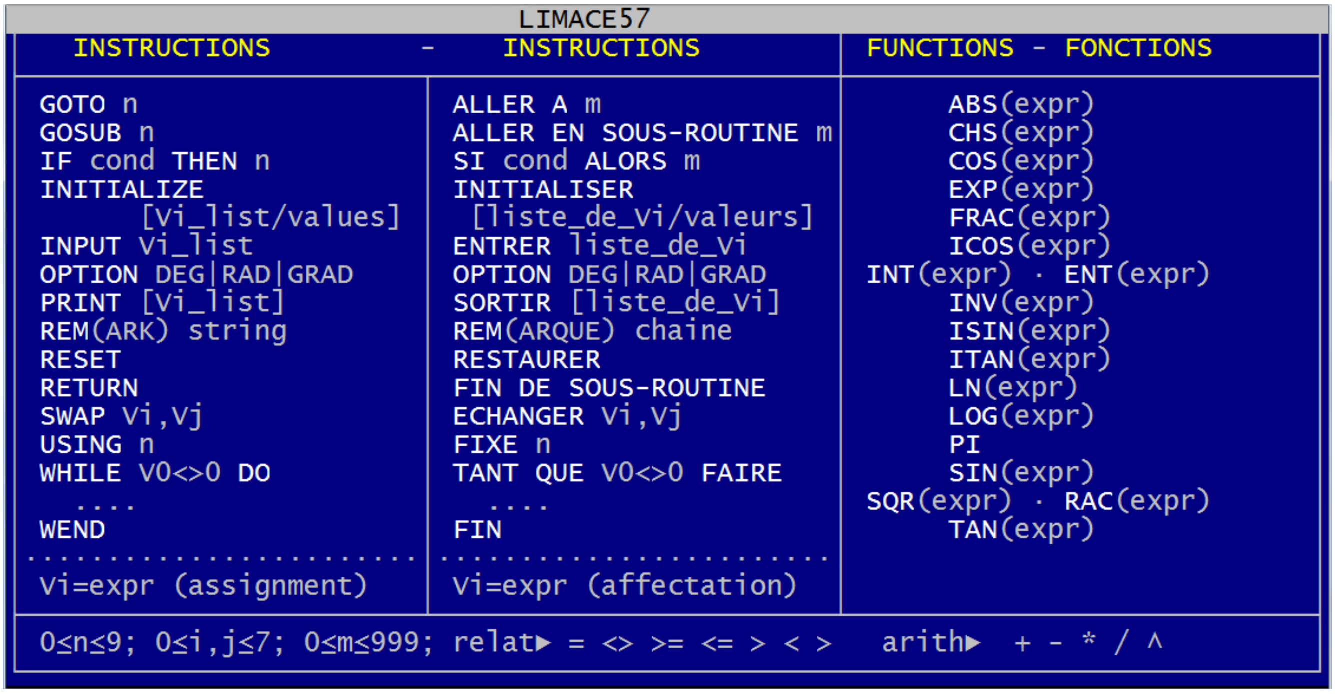


```
                      ♦ LIMACE FOR TI-57 v 2.0 ♦

Command> help
Available Commands:
  LIST   (L) COMPIL (C) LOAD     (LO)  SAVE  (SA) FILES (F) *FILES      (*F)
  INSERT (D) DELETE (D) LLIST    (LL)  NEW   (N)  HELP  (H) CREATE [/C] (CR)
  QUIT   (Q) PATH   (P) LANGUAGE (LAN) CLS        DOS       n (line #)
  /25 /43 /50 (screen width)
Command>
```

*LIMACE programming environment commands: english version is used.*

The main features of the language are:

- 8 variables, from V0 to V7;
- 10 lines callable by GOTO/GOSUB instructions (at maximum);
- WHILE …WEND loop can use V0 only;
- 999 LIMACE lines (at maximum);
- for LIMACE keywords and functions see table below.



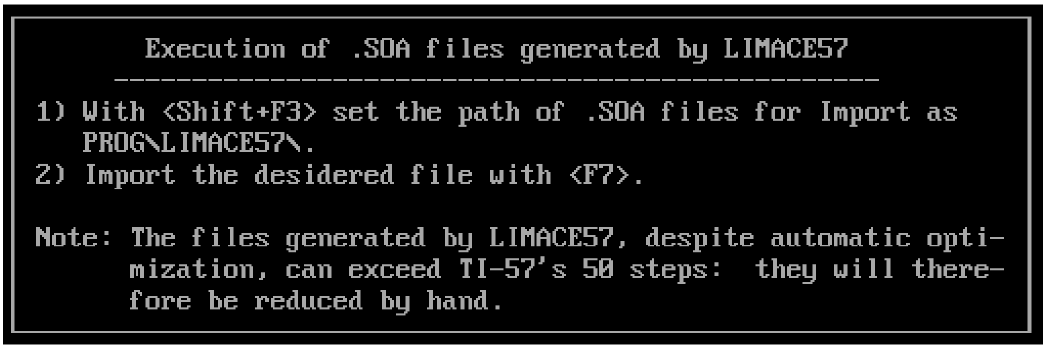
```
                                   LIMACE57
   INSTRUCTIONS              -     INSTRUCTIONS              FUNCTIONS - FONCTIONS

 GOTO n                     ALLER A m                              ABS(expr)
 GOSUB n                    ALLER EN SOUS-ROUTINE m                CHS(expr)
 IF cond THEN n             SI cond ALORS m                        COS(expr)
 INITIALIZE                 INITIALISER                            EXP(expr)
       [Vi_list/values]      [liste_de_Vi/valeurs]                 FRAC(expr)
 INPUT Vi_list              ENTRER liste_de_Vi                     ICOS(expr)
 OPTION DEG|RAD|GRAD        OPTION DEG|RAD|GRAD              INT(expr) · ENT(expr)
 PRINT [Vi_list]            SORTIR [liste_de_Vi]                   INV(expr)
 REM(ARK) string            REM(ARQUE) chaine                      ISIN(expr)
 RESET                      RESTAURER                              ITAN(expr)
 RETURN                     FIN DE SOUS-ROUTINE                    LN(expr)
 SWAP Vi,Vj                 ECHANGER Vi,Vj                         LOG(expr)
 USING n                    FIXE n                                 PI
 WHILE V0<>0 DO             TANT QUE V0<>0 FAIRE                   SIN(expr)
   ....                       ....                           SQR(expr) · RAC(expr)
 WEND                       FIN                                    TAN(expr)
.......................... ..........................
 Vi=expr (assignment)       Vi=expr (affectation)

 0≤n≤9; 0≤i,j≤7; 0≤m≤999; relat▶ = <> >= <= > < >   arith▶ + - * / ∧
```

*LIMACE language instructions (English and French)*

Let's see an example of a program translation (factorial calculation):

| EFACT.LIM | EFACT.SOA |
|---|---|
| 7 | |
| INPUT V0 | R/S STO 0 |
| V1=1 | 1 STO 1 |
| WHILE V0<>0 DO | LBL 0 |
| V1=V1*V0 | RCL 0 PRD 1 |
| WEND | DSZ GTO 0 |
| PRINT V0 | RCL 0 PAU |
| STOP | R/S |

Note: 7 is the number of lines of this program



```
       Execution of .SOA files generated by LIMACE57
     ------------------------------------------------
 1) With <Shift+F3> set the path of .SOA files for Import as
    PROG\LIMACE57\.
 2) Import the desidered file with <F7>.

 Note: The files generated by LIMACE57, despite automatic opti-
       mization, can exceed TI-57's 50 steps:  they will there-
       fore be reduced by hand.
```

*Instructions for running the programs compiled on the emulator*

## 11. TI-57 LCD EMULATOR.

The TI-57 LCD (or TI-57II) emulator tries to overcome two strong limitations of the original machine. As already mentioned in § 4.1, the emulator can be recalled with two main options **/NUM** e **/EXT**, so

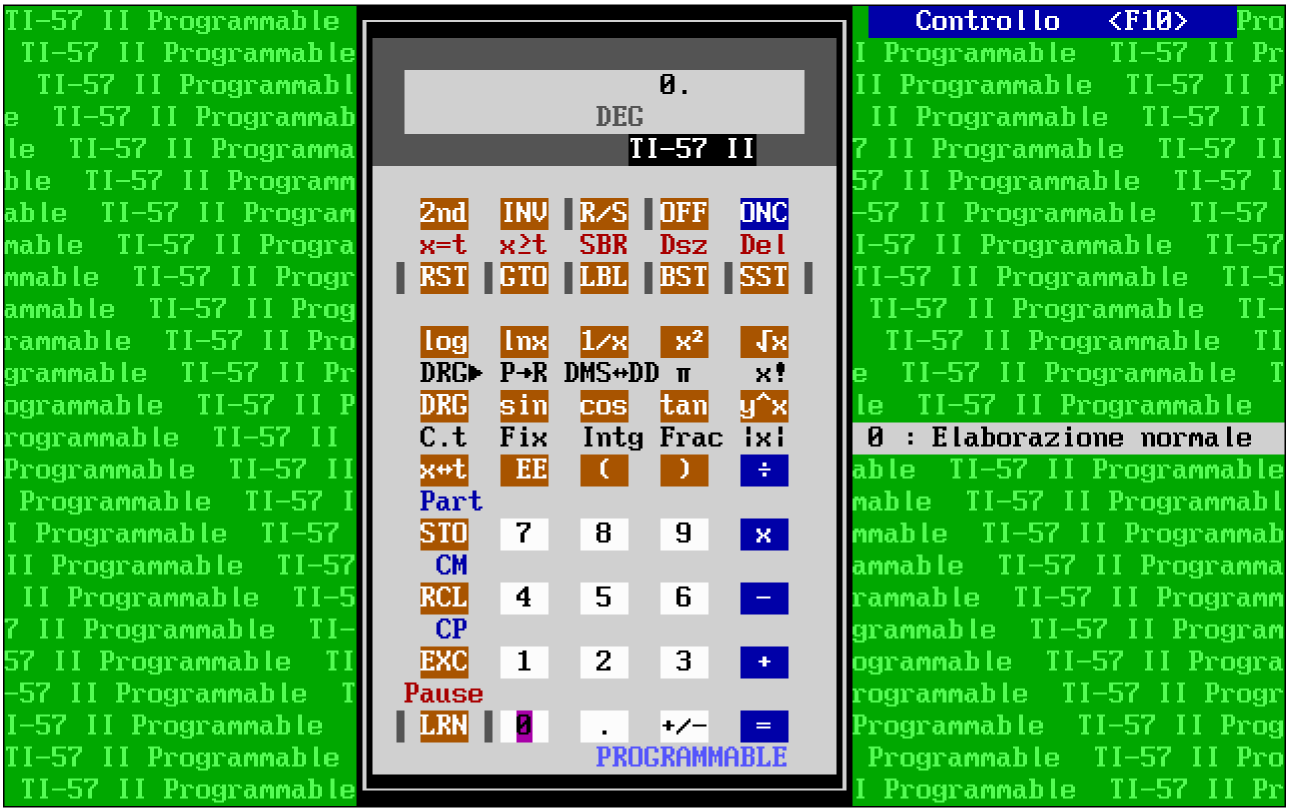
1) without the **/NUM** switch, it displays instructions in LRN-mode alphanumeric as on the TI-66. However, it is possible, during execution, switch from the alphanumeric LRN-mode to the standard numeric one (and vice versa) by pressing **Alt + S**.

2) with the **/EXT** switch, it allows you to have a TI-57LCD with a memory space comparable to that of the TI-57 (LED), that is 96 program steps or 7 memories instead of 48 program steps or 7 memories.

The recommended command-line for execution is therefore

***TI57LCD / EXT (or TI57LCD-64 /EXT)***

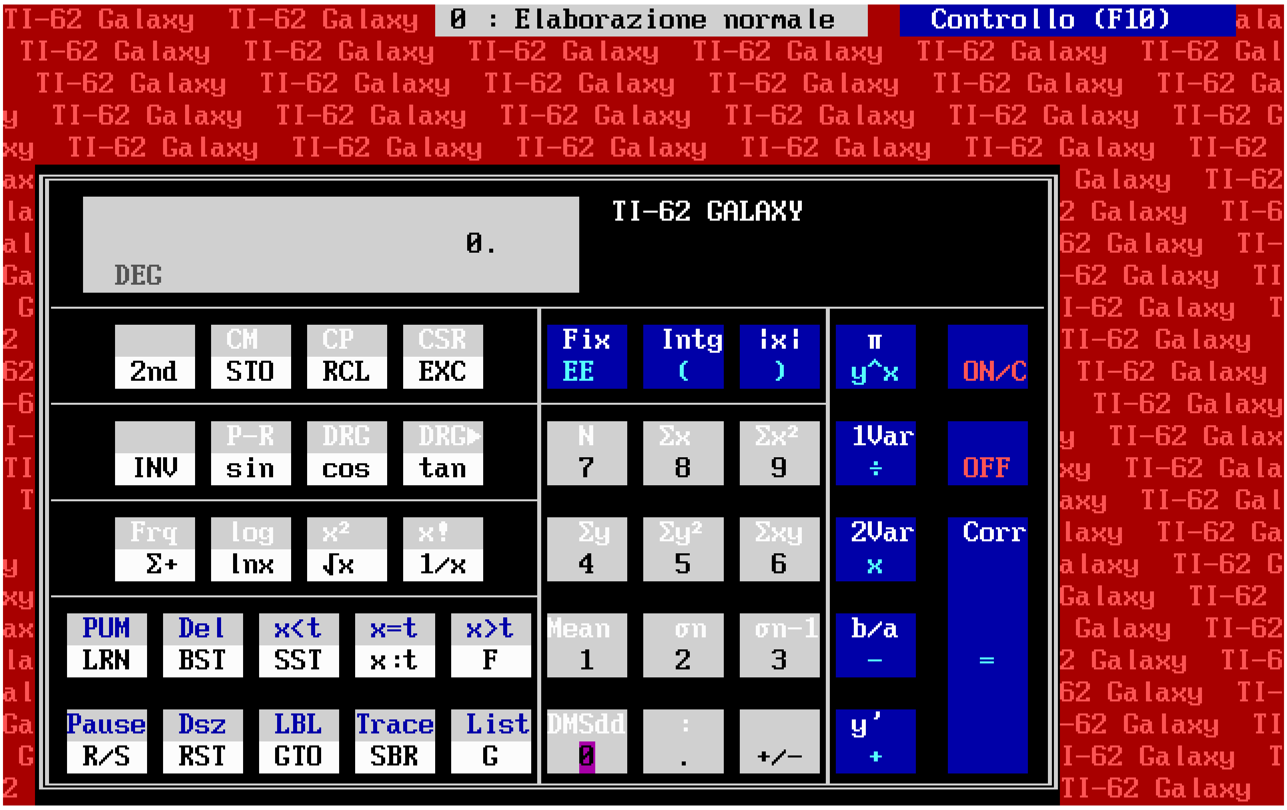
which allows you to have maximum flexibility in the use of the emulator.



*TI-57LCD.EXE*

## 12. TI-62 EMULATOR.

The TI-62 emulator faithfully follows the original machine, except for the fact that the system of indicators for the functions of polar/rectangular conversions and for the statistical functions for which the method of the others has not been implemented.



Let's see the differences on two examples.

1) Convert (45,60 °) from polar to rectangular coordinates

| | | | |
|---|---|---|---|
| [ON/C] [ON/C] | 0. | [ON/C] [ON/C] | 0. |
| [INV][x:t] | 0. | [INV][x:t] | 0. |
| [2nd][P-R] | 0.<br>ENTER<br>r <=> | 45 [x: t] | 0. |
| 45 [x: t] | 0.<br>ENTER<br>θ = | 60 [2nd][P-R] | 22.5 |
| 60 [ = ] | 22.5<br>x <=> | [x :t] | 38.97114317 |
| [x :t] | 38.97114317<br>y <=> | | |

TI-62 real TI-62 emulated

*Note*: On the real TI-62 an alternative execution is also possible, as on the TI-58/58C/59, by following the [ = ] key to [2nd][P-R].

2) Evaluate the following one-dimension statistical data :

13,13,13,15, 16 con media= 14
Frq=3

| | | | |
|---|---|---|---|
| [ON/C][ON/C]<br>[2nd][CSR] | 0. | [ON/C] [ON/C]<br>[2nd][CSR] | 0. |
| [1Var] | 0.<br>1 STAT | [1Var] | 0.<br>1 STAT |
| 13 [Σ+] | 1. 1 STAT<br>ENTER<br>x Σ/Frq | [2nd][Frq] 03 | FRQ 03<br>1STAT |
| 3 [2nd][Frq] | FRQ 00 1STAT<br>ENTER<br>f Σ | 13 [Σ+] | 3.<br>1 STAT |
| 15 [Σ+] | 4. 1 STAT<br>ENTER<br>x Σ/Frq | 15 [Σ+] | 4.<br>1 STAT |
| 16 [Σ+] | 5. 1 STAT<br>ENTER<br>x Σ/Frq | 16 [Σ+] | 5.<br>1 STAT |
| [2nd][Mean] | 14. 1 STAT<br>ENTER<br>x Σ/Frq | [2nd][Mean] | 14.<br>1 STAT |

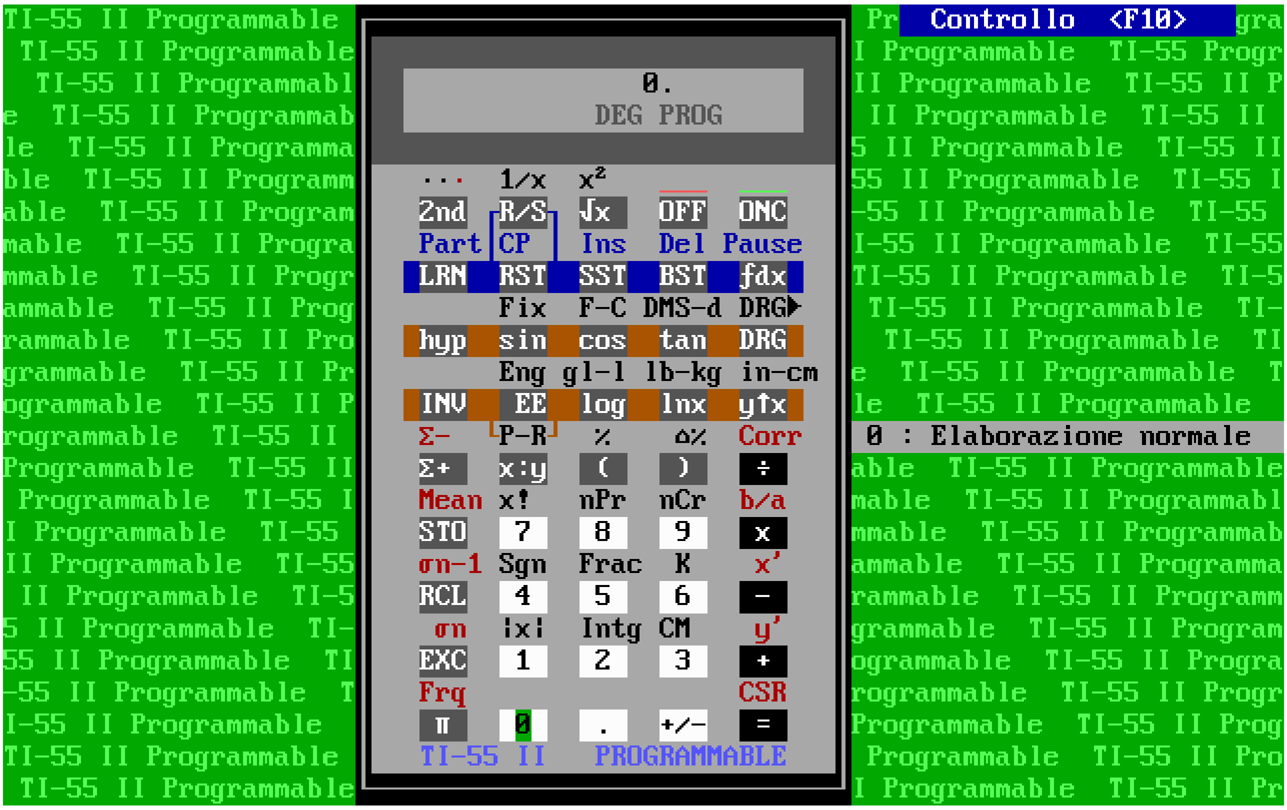
TI-62 real TI-62 emulated

In the emulated TI-62, the frequency input (Frq) must be done before entering the data, while in the real TI-62 it must be done after. From the method used by Texas Instruments we deduce that the design of these functions has not been done for programming, but for direct calculation only.

# 13. TI-55 II EMULATOR

The TI-55 II emulator faithfully follows the original machine, except for a few details listed here:

| TI-55 II – real | TI-55 II - emulated |
|---|---|
| • RST instruction repositions the program pointer to position 00 and stops program | • RST instruction repositions the program pointer to position 00 and doesn't stop the program (as on the TI-55). |
| • The OFF key **is not** programmable | • The OFF key **is** programmable |
| • In LRN-Mode the display shows instructions as numerical codes, according to the "Rule of the matrix". | • In LRN-Mode the display shows instructions in alphanumeric (like the TI-66) but, with the switch **/NUM** at statring numeric codes are used. As for the TI-57LCD **<At+S>** allows the transition between the two modes during the use of the emulator. |
| • The occupation of the single statistics registers is different from that of the TI-58 /58C /59, which in any case are not accessible during the STAT mode. | • The occupation of the single statistics registers is the same as that of the TI-58/58C/59, with equal results. The registers remain not accessible directly, but only viewable from the Status Panel. |



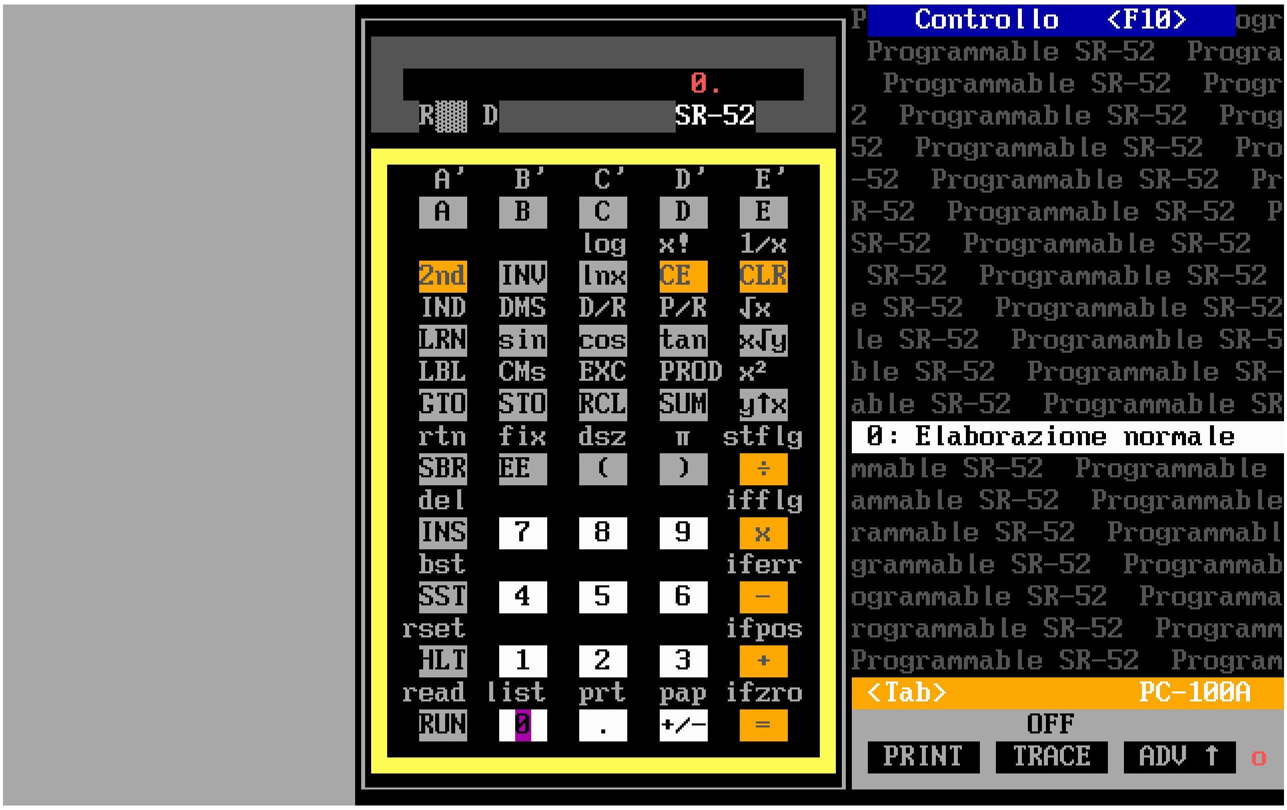
## 14. SR-52 AND SR-56 EMULATORS

The SR-52 and SR-56, together with their PC-100 printer, constitute the first generation of programmables from Texas Instruments. Beyond the constant memory, the differences of the emulator compared to the real models have been kept to a minimum, namely:

1) The arithmetic of the SR-52 is mixed (12 and 13 digits depending on whether direct ("display arithmetic") or memory ("memory arithmetic") arithmetic is used. The emulator, on the other hand, <u>always</u> uses 13-digit arithmetic;

2) It is possible, with the **/EXT** switch, to bring the memory capacity to 4 memory chips: as in the real expansion we will have two memory zones, which can be swapped at will (see figure on the next page);

3) It is possible to obtain the so-called "fractured digits" through "non-standard" keystrokes.



*SR-52.EXE (16 bit)*

4) For both models the printing possibilities have been extended: it is possible to have the program list in TI-59 format with the mnemonic codes, instead of just the numeric codes. In addition, the list of the contents of the memory registers (with **INV List**) not present in the real models has also been implemented.
5) The SR-56 emulator uses the 128 bytes internal firmware (written in AOS) for the **f(n)** instructions: an anticipation of what will be implemented on subsequent models (TI-58 / 58C / 59)

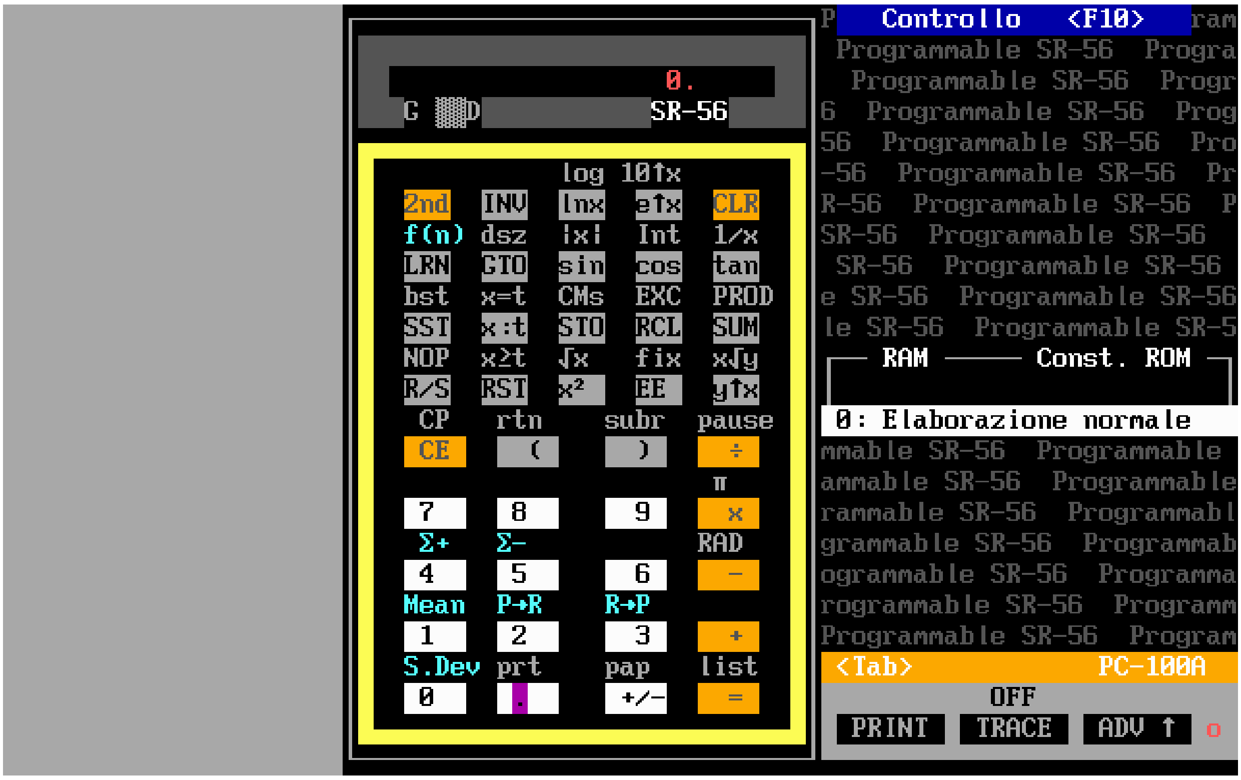
It is possible to import .SOA files with the **F7** key and, in the case of the SR-52, to save the program on a virtual "magnetic card": the format is similar to those of the TI-59 and was previously described in § 6.1.

|  | registers |  |
|---|---|---|
| Standard Memory<br>▨ | 00-19<br>20-29 | **RAM 1** |
| ▨ | 30-59 |  |
| Algebraic Stack<br>Program<br>224 steps | 60-69<br>70<br>89 | **RAM 2** |
| ▨ | 90-97 + [98-99 available memories]<br>100-119 |  |

**SR-52 standard configuration**

|  | registers |  |
|---|---|---|
| Standard Memory | 00-19<br>20-29 | **RAM 1** |
| Extended Memory | 30-59 | **RAM 2** |
| Algebraic Stack<br>Program #1<br>224 steps | 60-69<br>70<br>89 | **RAM 3** |
| Program #2<br>160 steps | 90-97 + [ 98-99 available memories]<br>100-119 | **RAM 4** |

**SR-52 extended configuration**

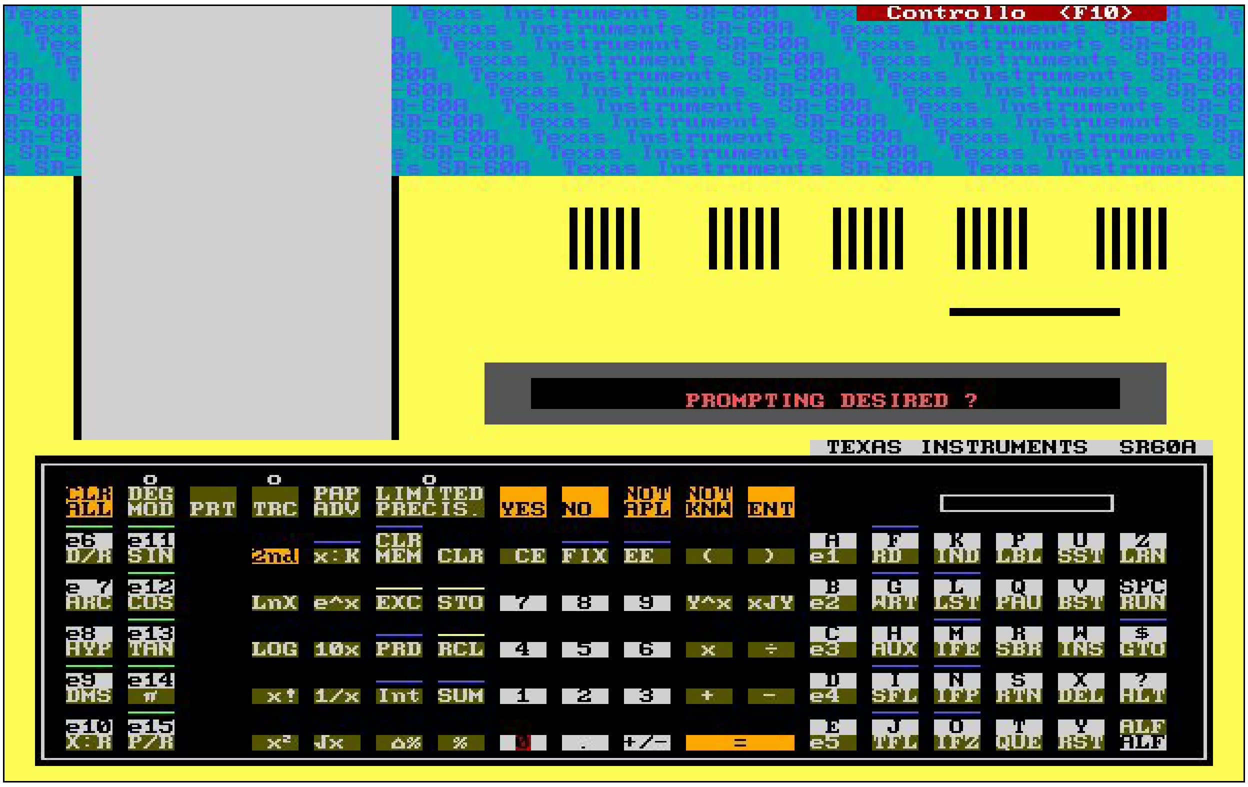


0: Elaborazione normale

*SR-56.EXE (16 bit)*

## 15. SR-60A EMULATOR.

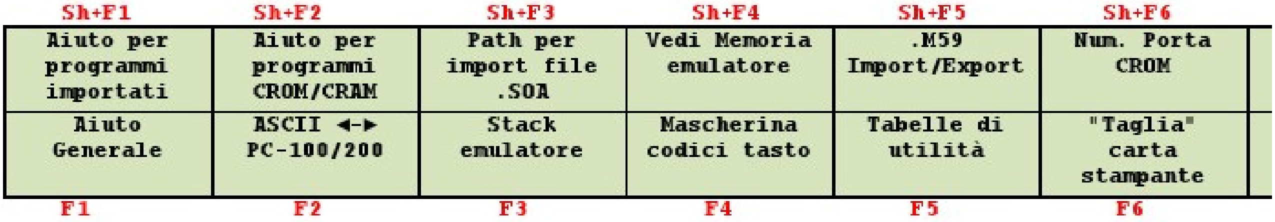
It's also available the emulator of SR-60A, a quite unusual machine in the world of programmable calculators of Texas Instruments. It’s, indeed, a desktop calculator with all integrated (thermal 20 columns printer and magnetic cards reader/writer) and a large memory capacity (up to 9600 bytes - 10 times greather than TI-59’s without Solid State Software module), a keyboard with 95 keys, an alphanumeric display, although the A.O.S. is on the level of that of SR-52 and, therefore, lower than that of the TI-59.



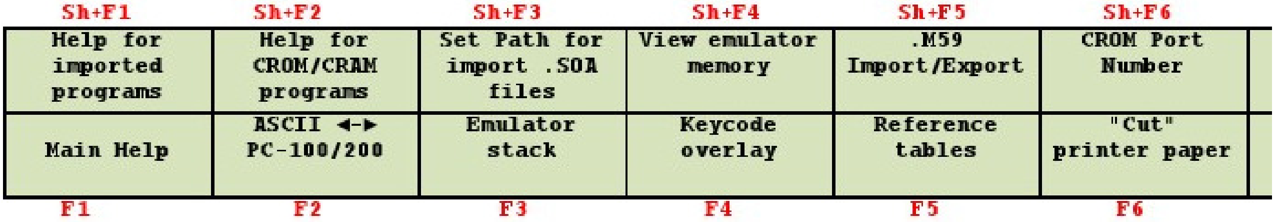
PROMPTING DESIRED ?

The emulator, unlike those for other models, does not have the constant memory although you can import .SOA files with the usual **F7** key, and, not having the real machine, all the information was taken from the operating manual and programming.

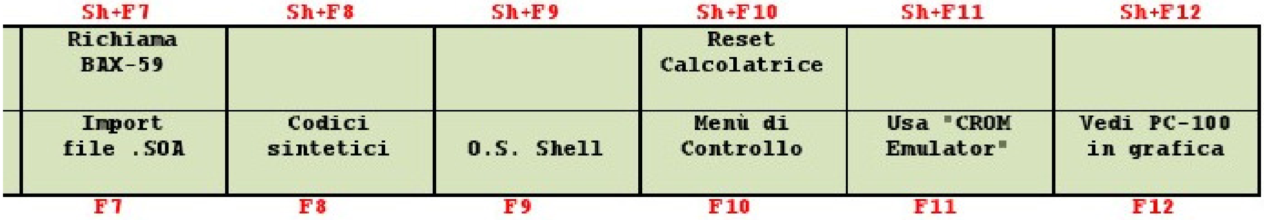
# APPENDIX – FUNCTION KEYS STICKER



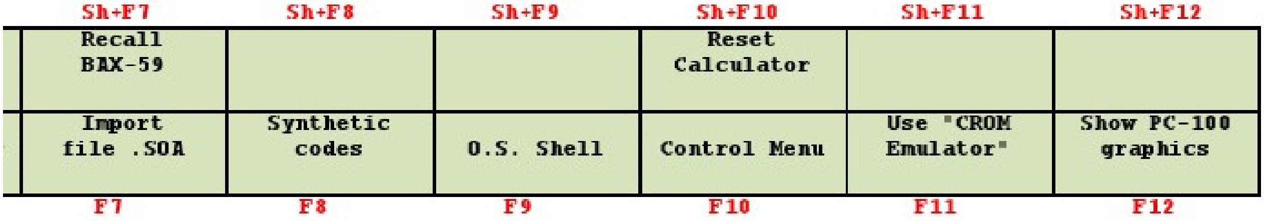
| Sh+F1 | Sh+F2 | Sh+F3 | Sh+F4 | Sh+F5 | Sh+F6 |
|---|---|---|---|---|---|
| Aiuto per programmi importati | Aiuto per programmi CROM/CRAM | Path per import file .SOA | Vedi Memoria emulatore | .M59 Import/Export | Num. Porta CROM |
| Aiuto Generale | ASCII ◄-► PC-100/200 | Stack emulatore | Mascherina codici tasto | Tabelle di utilità | "Taglia" carta stampante |
| F1 | F2 | F3 | F4 | F5 | F6 |



| Sh+F1 | Sh+F2 | Sh+F3 | Sh+F4 | Sh+F5 | Sh+F6 |
|---|---|---|---|---|---|
| Help for imported programs | Help for CROM/CRAM programs | Set Path for import .SOA files | View emulator memory | .M59 Import/Export | CROM Port Number |
| Main Help | ASCII ◄-► PC-100/200 | Emulator stack | Keycode overlay | Reference tables | "Cut" printer paper |
| F1 | F2 | F3 | F4 | F5 | F6 |



| Sh+F7 | Sh+F8 | Sh+F9 | Sh+F10 | Sh+F11 | Sh+F12 |
|---|---|---|---|---|---|
| Richiama BAX-59 | | | Reset Calcolatrice | | |
| Import file .SOA | Codici sintetici | O.S. Shell | Menù di Controllo | Usa "CROM Emulator" | Vedi PC-100 in grafica |
| F7 | F8 | F9 | F10 | F11 | F12 |



| Sh+F7 | Sh+F8 | Sh+F9 | Sh+F10 | Sh+F11 | Sh+F12 |
|---|---|---|---|---|---|
| Recall BAX-59 | | | Reset Calculator | | |
| Import file .SOA | Synthetic codes | O.S. Shell | Control Menu | Use "CROM Emulator" | Show PC-100 graphics |
| F7 | F8 | F9 | F10 | F11 | F12 |

# INDEX