

```

45 C$ = "DO IT THIS WAY IF YOU'RE DAFT"
50 IF LEN(C$)/2 < > INT(LEN(C$)/2) THEN C$ = C$ + " "
60 FOR A = 1 TO LEN(C$) STEP 2
70 B = ASC(MID$(C$,A+1,1)) * 256 + ASC(MID$(C$,A1))
80 DOKE 3017 + A,B
90 NEXT

```

Some "little known facts":

If you have to write a long DATA table or whatever consisting of very similar lines, you can type in the first line and press ENTER, then, using the cursor control keys, overwrite the line number with the number of the next line required and press ENTER, etc, etc. A LIST command will then show your series of identical lines for editing as necessary.

When entering a line of the full 48 characters the act of typing the final character places the cursor on the following line and pressing ENTER has no effect. It is only necessary, however, to place the cursor back up on the line - at any point - for a ENTER to enter the whole line.

I have added a Bits & P.C.'s hex pad to my Nascom 2, which works well for hex and integers but lacks a decimal point. The eventual solution must be a hardware fix, but my interim solution is a short series of Nas-Sys instructions which copy out the keyboard table from ROM into that portion of RAM not accessed by BASIC, switch the values for "." and "F" and gives Nas-Sys the address of the new table. Full credit goes to the writer of Nas-Sys for leaving the door open!

```

At switch-on:
C 059E 0C80 0060      (Copy out table)
M 0CAE
43 / 0CC6
21 / 0C6F              (Switch values)
80 0C.

```

```

After a Reset:
M 0C6F
80 0C

```

Yours faithfully,

D. Walker.
Windsor.

PIO HELP NEEDED
=====

Dear INMC,

Could you include some programming hints on the use of the PIO on Nascom 1 in some future issue? One useful example would be a traffic lights program driving a LED display via the PIO. There is a lot of