

## Esami di guida

di Guido Zualdi - Perugia

Con questi programmi la PC-1211 è in grado di memorizzare tutte le risposte dei questionari per gli esami di guida. L'allievo poi, con il questionario sottomano, può selezionare la pagina che desidera e cominciare a introdurre le risposte: la macchina controllerà la loro esattezza, segnalerà quelle errate e terrà il conto del numero di pagine fatte e di errori commessi.

La memorizzazione delle risposte è stata compattata in modo da fare entrare in un solo registro di memoria tutte le 18 risposte di una pagina. Ogni risposta è composta da un numero compreso fra 1 e 3, al quale la macchina sottrae 1, in modo da ottenerlo compreso fra 0 e 2. Ogni cifra del registro

### Guida - Programma Inizializzazione

```
10: CLEAR : INPUT # "DATA1"; A(26)
20 : T=26: CHAIN "DRIVE", 00
```

contiene due risposte: la cifra è formata dal numero della prima più il numero della seconda moltiplicato per 3. Rimane, tra l'altro, una cifra non utilizzata, che consentirebbe la memorizzazione di altre due risposte.

Il programma "INPUT" serve per creare il file con tutti i dati delle risposte in forma compattata, e deve essere eseguito solo all'inizio. Per utilizzarlo occorre inserire il numero della pagina da caricare e premere SHFT X. La macchina comincerà

e chiedere le varie risposte, per ognuna delle quali occorre introdurre un numero compreso fra 1 e 3. Dopo la domanda Z la calcolatrice entrerà in elaborazione per alcuni secondi e si predisporrà per la pagina successiva. In caso di errori di immissione è possibile fermare la macchina con il BRE-AK, scrivere la lettera della domanda errata e premere SHFT =. Dopo aver corretto la risposta è possibile continuare inserendo le altre o, se sono già state inserite, premendo SHFT SPC. In questo caso la macchina parte direttamente con la codifica delle risposte nel registro.

Se a qualche domanda non è stata data risposta (il registro relativo si trova a zero), la calcolatrice se ne accorge e prima di iniziare la codifica delle risposte torna a chiederla. Dopo averla fornita e premuto ENTER, è possibile tornare in codifica sempre con SHFT SPC.

Dopo il caricamento della centesima pagina, o in qualunque momento premendo SHFT K, la macchina chiederà se si vogliono registrare su cassetta i dati introdotti.

Il programma "DRIVE" serve invece per far esercitare l'allievo ed è di uso abbastanza simile: la pagina dalla quale si vuole incominciare va inserita con SHFT X o SHFT V. La differenza tra questi due comandi consiste nel fatto che SHFT X rimette a zero i contatori delle pagine fatte e degli errori commessi, mentre SHFT V li lascia invariati. Le regole per l'immissione delle risposte sono le stesse del programma "INPUT", ed anche i comandi SHFT = e SHFT SPC hanno le stesse funzioni.

Completata una pagina, la macchina controlla per prima cosa se è stata data risposta a tutte le domande, poi esegue un controllo globale dell'esattezza delle risposte, codificandole e controllando se il risultato è uguale a quello memorizzato nel registro. Se il test dà esito negativo, la macchina comincia a controllare risposta per risposta, evidenziando quelle errate con un messaggio di errore di questo tipo:

1. /R. 2./ES. 3.

ad indicare che alla domanda numero 1 è stata data risposta 2, mentre quella esatta era tre.

I tempi di elaborazione sono di circa 20 secondi se non vengono trovati errori, e di circa 30 secondi in caso contrario. Non si tratta di un tempo di risposta fulmineo, ma comunque dopo un po' ci si può fare l'abitudine, e può anche essere usato per riposarsi dopo l'enorme "sforzo mentale" necessario per rispondere.

Dopo ogni pagina la macchina segnala il numero di pagine fatte e di errori commessi. La macchina può essere fermata e spenta in qualsiasi momento, alla riaccensione è possibile ripartire dalla stessa pagina premendo SHFT =.

Il caricamento di questi programmi deve essere fatto con estrema cautela, in quanto "DRIVE" occupa tutto lo spazio disponibile, non uno step di meno, ed è stato notevolmente compattato con messaggi alfanumerici piuttosto crittografici, virgolette e parentesi tonde non chiuse, ecc. Per questo bisogna fare attenzione a non inserire assolutamente nulla che non sia sul listato, fare attenzione alla linea 70 e al termine

### Drive - Programma principale

```
2:Y=0+1:PRINT Y: /D. "A(Y)"/ES. ";U:RETURN
5:" IF ABCDEFHIJKLINSOPQRGOTO 30
10:FOR Y=1TO 18:IF A(Y)=0THEN 5Y:105
20:NEXT Y
30:S=0:FOR Y=1TO 9:S=S+10-(9-Y)*(A(2Y-1)+3A(2Y)-4):NEXT Y:IF S=A(T)
  REEP 3:GOTO 70
40:S=A(T)/ES:FOR Y=1TO 18:U=INT 105:S=105-U:V=1+INT (U/3)+U-2V+4:
  IF A(Y)<=UOSUB 2
50:Y=Y+1:IF A(Y)<=U:LET U=UOSUB 2
60:NEXT Y
70:X=X+1:T=T+1:IF T>12:PRINT "INPUT END -ERR.":M:END
80:FOR Y=1TO 18:A(Y)=0:NEXT Y
90:PRINT X:"PAG. * ";M:"ERR.
100:"PRINT T-25:" PAG.
110:"A"INPUT "A":A
115:"B"INPUT "B":B
120:"C"INPUT "C":C
125:"D"INPUT "D":D
130:"E"INPUT "E":E
135:"F"INPUT "F":F
140:"G"INPUT "G":G
145:"H"INPUT "H":H
150:"I"INPUT "I":I
155:"REPEAT 1:INPUT "M":J
160:"N"INPUT "N":K
165:"O"INPUT "O":L
170:"P"INPUT "P":M
175:"Q"INPUT "Q":N
180:"R"INPUT "R":O
185:"T"INPUT "T":P:ERR
190:"U"INPUT "U":Q
195:"Z"INPUT "Z":R
200:GOTO 5
210:"A"HEAD U:GOTO U
220:"X"HEAD T:X=0:Y=0
230:"V"HEAD T:IF (T<1)+(T>X)END
240:T=T+25:GOTO 80
MEM:782 STEPS 99 MEMORIES
```

### Input - Immissione risposte questionari

```
10:"X"HEAD Y:IF (Y<1)+(Y>26)EEP 1:END
20:FOR X=1TO 18:A(X)=0:NEXT X
30:"REPEAT 1:PRINT "PAGINA ";Y
40:"A"INPUT "A":A
50:"B"INPUT "B":B
60:"C"INPUT "C":C
70:"D"INPUT "D":D
80:"E"INPUT "E":E
90:"F"INPUT "F":F
100:"G"INPUT "G":G
110:"H"INPUT "H":H
120:"I"INPUT "I":I
130:"REPEAT 1:INPUT "M":J
140:"N"INPUT "N":K
150:"O"INPUT "O":L
160:"P"INPUT "P":M
170:"Q"INPUT "Q":N
180:"R"INPUT "R":O
190:"T"INPUT "T":P
200:"U"INPUT "U":Q
210:"Z"INPUT "Z":R
220:" " IF ABCDEFHIJKLINSOPQRGOTO 250
230:FOR X=1TO 18:IF A(X)=0:LET X=10X+30
240:NEXT X:GOTO X
250:REPEAT 1:U=Y+29:A(U)=0:FOR X=1TO 17 STEP 2:A(U)=10A(U)+A(X)+3A(X+1)=0:
  A(X)=0:A(X+1)=0:NEXT X
260:Y=Y+1:IF Y<=26:GOTO 30
270:REPEAT 2:PRINT "FINE DELLE PAGINE
280:"M"INPUT "REGISTRAZIONE? (SI/NO) ";S:IF S#="NO"END
290:IF S#="SI"GOTO 310
300:GOTO 260
310:PRINT "PREPARA NASTRO (ENTER)":PRINT # "DATA1":A(30):END
400:"A"HEAD U:GOTO U
MEM:831 STEPS 103 MEMORIES
```

Le due parole sottolineate della riga 70 sono due "tokens", inseriti al posto delle parole corrispondenti per risparmiare spazio. Effettuare molta attenzione in sede di copia a non aggiungere assolutamente nulla, altrimenti la memoria non è sufficiente.

controllare che siano rimasti sempre esattamente 792 step.

Questa raccomandazione vale anche per "INPUT", per il quale non ci sarebbero particolari problemi di spazio, tuttavia in questo programma l'area dati è stata posta negli ultimi 100 registri disponibili, per fare in modo che in fase di registrazione venissero salvati solo quelli e non altri, considerato che nell'istruzione PRINT è possibile specificare il registro di inizio, ma non quello di fine. Attenzione quindi a non allungare neanche questo programma.

Il programma "GUIDA" infine è solo una piccola routine di caricamento del file dati (con il nome "DATA1") e del programma DRIVE, con autostart di quest'ultimo a partire dalla prima pagina. Ovviamente per poterlo usare è necessario che i file "GUIDA", "DATA1" e "DRIVE" risiedano sequenzialmente su nastro, e possibilmente siano registrati molto vicini, in modo da avere brevi tempi di caricamento e minori possibilità di errori di ricerca.

## REMARKS

### "DRIVE"

2	Visualizzazione errore
5	Controlla se tutte le domande hanno avuto risposta
10-20	Scopre la prima domanda senza risposta e torna a chiederla
30	Controllo globale esattezza risposte
40-60	Controllo singolo esattezza risposte
80	Azzeramento registri risposte (A-R)
110-195	Input risposte

### "INPUT"

20	Azzeramento registri risposte (A-R)
40-210	Input risposte
220-240	Controllo se a tutte le domande è stata data risposta
250	Composizione del registro di memoria con le 18 risposte codificate.
280-310	Registrazione su nastro

## Uso delle variabili

### "DRIVE"

A-R	Risposte
S	Variabile di comodo
T	Puntatore al registro contenente le risposte della pagina corrente. T-25 dà il numero della pagina corrente.
U,V	Variabili di comodo
W	Contatore errori commessi
X	Contatore numero pagine fatte
Y	Contatore ad uso generale
Z-A(125)	Area dati

### "INPUT"

dA-R	Risposte
SS	SI-NO
U	Puntatore al registro contenente le risposte della pagina corrente.
X	Contatore ad uso generale
Y	Numero pagina corrente
A(30)-A(129)	Area dati

## Programmazione sintetica sulla PC-1211

di Guido Zualdi - Perugia

Premetto che tutto quello che seguirà è stato provato solo sulla mia calcolatrice, un modello appartenente alla prima serie di PC-1211, quelle sprovviste dell'istruzione CLOAD 1, e per questo non posso garantire che tutto funzioni anche sugli altri esemplari.

Sul numero 7 di MC sono stati pubblicati i codici esadecimali della PC, ed in fondo si suggerisce quel truccetto per fare apparire SALVE sul display. Sulla mia calcolatrice è possibile anche far riapparire la linea di programma cancellata: basta dare il comando LIST 10. Solo questo comando è in grado di farla riapparire, le frecce o il LIST senza numero di riga non hanno, giustamente, alcun effetto. LIST seguita da un numero di riga, invece, ignora totalmente il fatto che non vi siano programmi in memoria (o meglio che la memoria sia tutta libera per i registri) e comincia a cercare la linea richiesta, o perlomeno qualcosa che gli assomigli. Se la trova, la visualizza, se trova qualcosa che assomiglia ad una linea con un numero superiore segnala errore (linea richiesta non trovata), se infine non trova assolutamente niente continua a cercare entrando in un loop infinito, e l'unico sistema per sbloccarla è l'ALL RESET. Preciso infine che tutto ciò avviene solo se non vi è alcun programma in memoria: basta una sola linea e tutto procede normalmente.

Un'altra particolarità di questo strano LIST forzato è che ora la macchina non sa dove finisce il programma, e quindi, una volta entrati nel modo LIST, è possibile con i tasti ↓ e ↑ spostarsi per tutta la memoria, ovviamente con alcune piccole limitazioni, ovvero la freccia ↓ consente di avanzare praticamente byte per byte, mentre la freccia ↑ consente di tornare indietro solo alla prima "vera" (ce ne sono anche di apparenti) linea di programma.

Ultima caratteristica di questa specie di MONITOR (visto che consente di analizzare l'intera memoria) è che se un'area di memoria contiene qualcosa, questo qualcosa viene effettivamente visualizzato, se invece non contiene niente (presumibilmente degli zeri binari o perlomeno dei Nop) viene visualizzato una specie di "numero di riga" esadecimale (visualizzato con le solite convenzioni dalla PC) seguito da un "8000" e con il simbolo ":" classico del LIST. Questo fantomatico "numero di riga" sembrerebbe l'indirizzo fisico esadecimale della prima cella di memoria visualizzata. In questo modo ho potuto ricavare la mappa della memoria della calcolatrice. Osservandola si possono notare due cose: l'area di memoria RESERVE è di 96 celle, non 48, e quella di programma è lunga 2848 celle, e non 1424, il che significa che le celle sono di 4 bit ciascuna (e quindi dovrebbe esserlo anche il processore). Altra amara constatazione è la duplicazione de-

gli indirizzi nell'ultima parte: sapere che forse sarebbe stato possibile aggiungere altri 512 step (1024 celle) e che ciò non è stato fatto è un po' deprimente.

Comunque ora è possibile generare le prime istruzioni (o volendo è possibile chiamarli anche "caratteri" o "codici") sintetiche.

Si voglia generare il carattere 6B(√). Dalla vostra tabella si nota che il suo inverso, B6, è l'istruzione CSAVE.

Eseguire:

Modo PRO

NEW (enter)

10EEEEEEEEEEEEEE (enter) devono essere almeno 14 E.

10 (enter)

AS(203) = EEEEECSAVE (enter) la PC segnala errore richiama la linea ed inserire le virgolette " dopo il segno =.

LIST 10 (enter)

comparirà: 10:EEEEEE√EEEEEO

Il carattere √ è l'istruzione sintetica. Volendo generarne altre, è possibile spostare tale carattere con il DEL all'inizio della riga, o perlomeno nei primi 6 caratteri, e ripristinare la serie di E in modo che la linea occupi sempre almeno 17 step di memoria, in modo da coprire tutta l'area riservata alla memoria A(203). L'errore nell'assegnazione ad AS(203) è necessario per permettere alla macchina di "tokenizzare" l'istruzione CSAVE. Infine ho usato la lettera E come carattere per riempire, perché la E ha codice 55 e quindi non varia se invertita. C'è un'istruzione che non possiede inverso (o meglio lo possiede, ma è un Nop, e quindi non può essere introdotto da tastiera), ma che comunque può essere ugualmente generata: è il codice 50 (—). Per ottenerlo occorre sostituire l'assegnazione alla memoria 203 con A(203) = 5.505050505E 10 e poi, dopo LIST 10, premere un volta ↓.

Ora diamo un'occhiata alla memoria RESERVE: è situata nelle locazioni 00-5F, però per poterla listare occorre percorrere prima tutto il resto della memoria e fare molta attenzione a non oltrepassarla, altrimenti bisogna ricominciare tutto da capo. Una volta raggiunta, però, può essere trasferita nella memoria di programma ed analizzata: basta aggiungere in cima un numero di riga e premere ENTER. Si scopre così che quando viene premuto un tasto definibile la macchina cerca nella memoria RESERVE un carattere particolare (un prefisso) e, se lo trova, considera come messaggio associato al tasto tutto quello che lo segue fino ad un Nop o ad un altro carattere di prefisso. Questi prefissi risiedono tutti nelle ultime due righe della tabella, cioè il loro codice esadecimale è del tipo Ex o Fx. Ora anche questi prefissi sono caratteri sintetici, ma non solo, è possibile generare anche gli inversi di questi caratteri: ecco quindi che si allarga il numero di caratteri generabili.

Bisogna dire però che qui cominciano a nascere i problemi: alcuni di questi prefissi

sono di tipo non stampabile, si comportano cioè come i caratteri 85, 86, 96, 97. Nella mia macchina questi caratteri a volte si limitano a modificare gli indicatori del display, a volte invece bloccano il visualizzatore, che rimane "congelato" fino a che, agendo sulle frecce del cursore, non si sposta il carattere "incriminato" fuori della finestra del visualizzatore. A volte invece essi possono provocare lo spegnimento della macchina e il suo blocco totale alla riaccensione. I diversi comportamenti della macchina dipendono da fattori per ora sconosciuti e, almeno per il momento, imprevedibili: neanche l'ALL RESET è in grado di riportare questi fattori in uno stato fisso e quindi, volendo lavorare con questi caratteri, bisogna aspettare che la calcolatrice "ne abbia voglia" e si metta a fare la brava.

Altri codici prefisso hanno un altro effetto, fastidioso ma non catastrofico: se vengono inseriti all'interno di una linea di programma, la rendono incancellabile con il metodo classico (numero di riga ed ENTER). Anche modificando la linea, ci si ritrova dopo in memoria sia la vecchia che la nuova versione, entrambe incancellabili.

Per convincere la macchina a cancellare la linea è necessario usare il registratore: bisogna iniziare il caricamento di un programma qualsiasi e poi, non appena la macchina ha letto il nome (il primo blocco corto seguito subito dopo dal primo blocco di dati) interrompere il caricamento con

il BREAK. Questo perché la macchina, dopo aver letto il nome, si prepara per caricare il nuovo programma mettendo a zero (o perlomeno a qualcosa di simile) i puntatori del BASIC. Arrestando la macchina a questo momento, la memoria non verrà alterata, in quanto ancora non è stato caricato nulla, però per il BASIC non ci sarà più alcun programma in memoria.

Tra l'altro con questo metodo, molta pazienza ed un uso intensivo dell'ALL RESET è possibile costruirsi una riga di programma contenente prefissi e istruzioni sintetiche. Questa linea può poi essere registrata su nastro e poi caricata nella memoria RESERVE (la macchina non fa distinzione fra programmi e file RESERVE e se non ci sono problemi di lunghezza qualsiasi registrazione può essere caricata in qualsiasi memoria). In tale modo è possibile avere direttamente in tastiera le istruzioni che interessano.

Ed ora un piccolo sguardo ad alcuni caratteri sintetici:

Codici 6B-6F (6C escluso): possono essere considerati come una estensione delle lettere: infatti ad esempio il simbolo  $\sqrt{\quad}$  può essere considerato come la 27ª lettera dell'alfabeto della PC, in quanto indirizza la memoria A (27). Istruzioni del tipo  $\sqrt{\quad} = \emptyset$  o PRINT  $\sqrt{\quad}$  sono perfettamente legali ed influenzano la memoria 27. Gli altri codici si comportano nello stesso modo. Unico problema è il codice 6C, che

dovrebbe esistere, ma non possiede inverso e quindi non è generabile.

Codice 50 (—): appresenta anch'esso un registro di memoria, ma con caratteristiche tutte sue. Utilizzandolo in calcoli da tastiera sembra un mini-registro con 4 cifre significative o una mini-stringa da 3 caratteri, ma se usato in un programma dopo un po' si "arricchisce" di cifre sbucate da chissà dove. Inoltre i suoi 4 byte "funzionanti" sono in comproprietà con la variabile V, cosicché modificando una, cambia anche l'altra. La mia ipotesi è che — occupi 4 byte della variabile V e poi altri 4 byte ad uso interno, byte che vengono rimessi a zero ogni volta che la macchina torna in modo attesa comandi. Per questa ragione esso sembra avere solo 4 cifre significative, le altre vengono sempre cambiate con zero. In ogni caso, esso può essere usato per troncatura (uso alquanto bislacco) il contenuto di V alle prime 4 cifre, senza modificare l'esponente, è sufficiente porlo uguale a zero.

Codici 3A-3F: appartengono alla riga dei simboli di operazione ed hanno le stesse loro proprietà, ovvero richiedono un'operazione prima ed uno dopo di essi. Inoltre, se immessi via tasto definibile subito dopo aver effettuato un calcolo, ricopiano anch'essi nel buffer di riga il risultato dell'operazione precedente. Solo un paio di essi sembrano avere una precisa (per modo di dire) funzione:

3B (SPC) effettua un'operazione scon-

PC-1211 - Codici esadecimali

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	Nop <sup>(1)</sup>	Nop <sup>(1)</sup>	Nop <sup>(1)</sup>	Nop <sup>(1)</sup>	Nop <sup>(1)</sup>	Nop <sup>(1)</sup>	Nop <sup>(1)</sup>	Nop <sup>(1)</sup>	Nop <sup>(1)</sup>	Nop <sup>(1)</sup>	Nop <sup>(1)</sup>	Nop <sup>(1)</sup>	Nop <sup>(1)</sup>	Nop <sup>(1)</sup>	Nop <sup>(1)</sup>	Nop <sup>(1)</sup>
1	Ins	SPC	.	?		#	%	¥	\$	π	$\sqrt{\quad}$	.			?	?
2	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
3	{	}	<	>	=	+	-	*	/	Λ	Ins	SPC	.	?		#
4	0	1	2	3	4	5	6	7	8	9	.	E	%	¥	\$	π
5	-	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
6	P	Q	R	S	T	U	V	W	(2)	Y	Z	$\sqrt{\quad}$	.			?
7	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
8	STEP <sup>(1)</sup>	THEN <sup>(1)</sup>	>= <sup>(1)</sup>	<= <sup>(1)</sup>	<> <sup>(1)</sup>	(3)	(3)	RUN <sup>(1)</sup>	NEW <sup>(1)</sup>	MEM <sup>(1)</sup>	LIST <sup>(1)</sup>	CONT <sup>(2)</sup>	DEBUG <sup>(2)</sup>	CSAVE <sup>(1)</sup>	CLOAD <sup>(1)</sup>	RUN <sup>(1)</sup>
9	TO	STEP	THEN	> <sup>(1)</sup>	< <sup>(1)</sup>	<> <sup>(1)</sup>	(3)	(3)	RUN <sup>(1)</sup>	NEW <sup>(1)</sup>	MEM <sup>(1)</sup>	LIST <sup>(2)</sup>	CONT <sup>(2)</sup>	DEBUG <sup>(1)</sup>		
A	SIN	COS	TAN	ASN	ACS	ATN	EXP	LN	LOG	INT	ABS	SGN	DEG	DMS	STEPS <sup>(1)</sup>	MEMOR <sup>(1)</sup>
B	RUN	NEW	MEM	LIST	CONT	DEBUG	CSAVE	CLOAD			THEN <sup>(1)</sup>			<> <sup>(1)</sup>		
C	GRAD	PRINT	INPUT	RADIAN	DEGREE	CLEAR					(3)			(3)	(3)	
D	IF	FOR	LET	REM	END	NEXT	STOP	GOTO	GOSUB	CHAIN	PAUSE	BEEP	AREAD	USING	RETURN	
E		THEN <sup>(1)</sup> (A)	>= <sup>(1)</sup> (B)	<= <sup>(1)</sup> (C)	<> <sup>(1)</sup> (D)	(3)	(3)	RUN <sup>(1)</sup> (G)	NEW <sup>(1)</sup> (H)	LIST <sup>(1)</sup> (J)	CONT <sup>(1)</sup> (K)	DEBUG <sup>(1)</sup> (L)	CSAVE <sup>(1)</sup> (M)	CLOAD <sup>(1)</sup> (N)		
F		THEN <sup>(1)</sup> (SPC)	<= <sup>(1)</sup> (S)	<> <sup>(1)</sup> (=)	LIST <sup>(1)</sup>	(3)	(3)	NEW <sup>(1)</sup> (X)		LIST <sup>(1)</sup> (Z)						

- (1) Carattere sintetico
- (2) Istruzione presente nella tabella di MC, ma non ricavabile con i metodi suddetti.
- (3) Carattere non stampabile

La tabella dei codici esadecimali della PC-1211

sciuta, ma perlomeno abbastanza definita, che dovrebbe avere qualche parentela con gli elevamenti a potenza: ha un tempo di calcolo abbastanza simile ed inoltre, se il secondo operando è uguale a due, fornisce in uscita sempre uno, come accade per gli elevamenti a potenza se il secondo operando è zero.

3F (#) non effettua una vera e propria operazione matematica, ma influisce sullo stato della macchina, eseguendo una specie di reset parziale: spegne tutti gli indicatori del display, pone la calcolatrice in un modo indefinito, privo di qualsiasi caratteristica particolare degli altri modi di funzionamento, ed infine visualizza zero. Se incontrata in un programma, ne arresta l'esecuzione senza più possibilità di riprenderla con il CONT.

Gli altri codici di questo gruppo non hanno funzioni particolari, tuttavia i codici 3A, 3B e 3C, insieme al codice 50, possono essere usati nelle istruzioni di PRINT per scrivere (solo su display) caratteri che normalmente non si possono inserire (□ " —). Lo spazio sintetico poi può essere usato nei REM., in quanto in fase di tokenizzazione non viene soppresso.

Codici AE-AF: non hanno funzioni particolari, vengono solo rappresentati con le parole STEPS e MEMORIES (quelle che compaiono nei MEM). MEMORIES è tra l'altro l'unica parola lunga ben nove lettere, spazio incluso, e quando viene visualizzata può creare ogni tanto dei problemi di scrolling orizzontale. Da qualche parte dovrebbe anche esserci la parola BREAK IN, ma non sono riuscito a trovarla.

Codici 4C-4F: sono gli stessi che compaiono durante la visualizzazione dei numeri esadecimali, e sono ovviamente diversi dai loro corrispondenti % ¥ \$ e π. Da notare che questi caratteri, se fatti valutare singolarmente (introducendo il simbolo e premendo ENTER) forniscono in uscita il loro valore in decimale, ma non hanno gli stessi effetti se usati in calcoli. Da notare anche che questi quattro caratteri rappresentano le cifre C-F esadecimali, e forniscono in uscita il loro valore decimale. La rappresentazione delle cifre A e B è invece affidata ai caratteri . ed E, che non sono sintetici, e quindi svolgono la loro particolare funzione.

Come accennavo all'inizio, non so se il sistema di generazione funziona su tutte le PC, dopo tutto un buco così grosso dovrebbe essere stato scoperto in fretta dai

progettisti (a meno che non fosse intenzionale). Per questo voglio accennare ad un altro buco della mia PC, questa volta molto più nascosto e difficile da scoprire, e quindi più difficile da eliminare:

si assegni ad un tasto definibile una sequenza abbastanza lunga di caratteri qualsiasi, possibilmente diversi fra loro. Si entri poi in modo PRO e si inserisca la seguente linea:

```
(1234567890123456789012345678
90123456789012345678901234567
89012345678901234567 + (enter)
```

L'ordine delle cifre non è importante, quello che invece è importante è il numero di caratteri immessi, che deve essere 79, e l'ultimo carattere, che deve provocare un errore.

Premuto ENTER, la macchina segnalerà questo errore (operazione aritmetica non terminata). Ora premendo la freccia a sinistra la linea NON riapparirà. Tenendo la freccia premuta per un po' compariranno sequenze di caratteri strani. A questo punto inserire la sequenza di caratteri memorizzata nel tasto definibile: si vedrà la parte sinistra del visualizzatore entrare in un modo grafico.

Volendo è possibile ripetere l'operazione più volte, alternando spostamenti del cursore con le frecce (spostamenti comunque invisibili) e inserimenti della sequenza. Alla fine comunque si tenga premuta la freccia sinistra fino a che il cursore non riappare, fermo, alla sinistra del display. Ora la linea prima introdotta è riapparsa, ma perlomeno nella prima parte non contiene più le cifre che erano state introdotte, ma vari caratteri, alcuni dei quali sintetici. Procedendo avanti con la freccia destra ogni tanto il cursore si bloccherà come se fosse arrivato alla fine della linea, anche se dopo di esso compaiono altri caratteri, in tale caso è sufficiente premere una o più volte lo spazio e poi continuare. Anche in questo modo è possibile generare delle istruzioni sintetiche, ma non sono mai riuscito a capire come è possibile generare i caratteri desiderati e non altri. Devo comunque dire che "normalmente" la calcolatrice si comporta in questo modo, ma anche qui, anche se più raramente, ogni tanto bisogna fare i conti con gli umori della macchina, e sperare che quella mattina si sia svegliata bene.

Il file SINTRES va caricato nella memoria RESERVE ed assegna ai tasti definibili

alcuni caratteri sintetici:

Tasto	Carattere	Codice
A	✓	6B
S	:	6D
D	:	6E
F	?	6F
G	%	4C
H	¥	4D
J	\$	4E
K	π	4F
L	STEPS	AE
=	MEMORIES	AF
Z	Non definito	
X	—	50
C	#	3F
V	!	3E
B	?	3D
N	"	3C
M	INS	3A
SPC	SPC	3B

Questo file può essere anche caricato della memoria di programma, al fine di esaminarne la struttura (attenzione però ai prefissi non stampabili). Al tasto Z non è stato assegnato alcun carattere, però il prefisso relativo è stato volutamente inserito nel file. In questo modo, premendo SHFT Z, la macchina trova il prefisso, ma non trovando nessun messaggio dopo di esso, ma semplicemente... un altro prefisso, si limita a... cancellare il visualizzatore, eseguendo però non un CL, ma un CA: ovvero l'eventuale programma in esecuzione viene interrotto senza più possibilità di eseguire il CONT.

Il prefisso del tasto Z è il primo del file, volendo assegnare anche a quest'ultimo un carattere sintetico basta eseguire quanto segue:

eseguire il LIST sintetico ed arrivare a listare la memoria RESERVE. Deve comparire: πSTEP□LIST RUN %NEW ¥;

cancellare i primi tre caratteri e sostituirli con un numero di riga. Inserire poi dopo LIST una serie da almeno dodici E. Infine memorizzare la linea con ENTER e ricancellarla con il trucco del registratore. Fare attenzione a non spostare troppo a destra il cursore per evitare la comparsa nel display di prefissi non stampabili.

A questo punto l'istruzione può essere generata con la solita assegnazione ad A\$(203). Al termine listare la linea, eliminare le E e la lettera O formatasi e rimemorizzare la linea. Registrare poi il tutto su nastro e ricaricare nella memoria RESERVE.

Con questo sistema è stato generato l'intero file SINTRES: sono stati prima assegnati ad ogni tasto definibile le lettere corrispondenti (ad es. a SHFT A è stata assegnata la lettera A), in modo da creare tutti i prefissi. In seguito la memoria RESERVE è stata trasferita nella memoria di programma ed è stato possibile sostituire le lettere con i caratteri sintetici.

Per costruire il tutto ho impiegato circa due giorni, tenendo in conto gli innumerevoli blocchi della macchina e gli altrettanto innumerevoli salvataggi del file incompleto per evitare di perdere tutto a seguito di un blocco.

## PC-1211 mappa di memoria

### Indirizzi (hex.)

000-05F	Memoria RESERVE
060-B7F	Memoria di programma
B80-BBF	Area riservata alle variabili Z, Y, X, W: 16 celle per ciascuna, a cominciare dalla Z.
BC0-BFF	Area ad uso interno.
COO-FFF	Contiene gli stessi dati dell'area 800-BFF. Si tratta probabilmente della stessa memoria che viene letta due volte sotto indirizzi diversi. Probabilmente il processore della PC avrebbe potuto indirizzare altra RAM mappata sotto questi indirizzi, che però non è stata installata (peccato!).