



a cura di Paolo Galassetti

MGP megaprint con buffer caratteri

Ho realizzato un buffer di 24 caratteri (l'intero registro ALPHA) per il megaprint. Ho dovuto cambiare il nome della LBL SP in LBL "*".

MGP impiega circa un minuto per ogni lettera da stampare. A patto di utilizzare un carattere alfanumerico di tastiera, p.es. 0, (anziché un carattere speciale) l'istruzione XEQ IND Y all'interno del loop LBL9... GTO9, può essere sostituita da una più veloce ARCL IND Y.

Su quest'idea è costruito PGM, che impiega meno di 40 secondi per lettera, con un risparmio di oltre il 33%. PGM occupa 7 registri in più, ma in compenso è 21 bytes più corto. Mi sembra che il gioco valga la candela. In PGM tutti i registri usati in MGP sono stati shiftati di 7.

Corrado Cavallini - Padova

GLRGX Clear Registers as directed by X

Manca nell'HP-41 una istruzione che permetta di cancellare solo alcuni dei registri di memoria. L'istruzione CLRG cancella tutti i registri. L'istruzione CLΣ cancella un gruppo di sei registri a partire da quello indicato con ΣREG. La routine proposta sopperisce allo scopo, utilizzando il contenuto del registro X come parametro di controllo, analogamente alle istruzioni WDTAX, RDTAX, PRREGX. La routine utilizza due registri della catasta. X deve contenere un numero della forma aaa.bbb, con aaa primo registro da pulire, bbb l'ultimo.

Corrado Cavallini - Padova

INVIATECI I VOSTRI PROGRAMMI!

Se, qualunque sia la vostra macchina, avete realizzato programmi o routine che ritenete possano interessare altri lettori, inviateceli. Saranno esaminati e, se pubblicati, ricompensati con valutazioni approssimativamente tra le 30 e le 100.000 lire, secondo la complessità, la genialità, l'originalità e la presentazione del materiale e della documentazione (listati, diagrammi, commenti ecc.). Per ragioni organizzative non possiamo impegnarci, salvo eventuali accordi presi prima dell'invio, alla restituzione dei materiali, che resteranno di proprietà della redazione che si impegna a non divulgarli (se non tramite la rivista) senza l'autorizzazione dei rispettivi autori.

Evidentemente qualcuno ha accolto la mia proposta apparsa sul n° 3 di MCmicrocomputer, commentando il programma "Megaprint" di Stefano Tendon, circa la possibilità di realizzare un buffer che non costringesse ad attendere il termine dell'operazione di stampa di un carattere prima di poterne impostare uno nuovo.

È così che l'ing. Corrado Cavallini, un nostro lettore, ci ha inviato due programmi che, pur non svolgendo le funzioni di un vero e proprio buffer, servono a stampare in formato "Megaprint" una stringa di un massimo di ventiquattro caratteri impostata nel registro ALPHA; tale capacità risulta sufficiente praticamente in tutti i casi.

Il programma "MGP" è piuttosto semplice concettualmente: la stringa viene scomposta nei singoli caratteri che vengono man mano utilizzati per richiamare indirettamente le routine del programma di codifica presentato sul n° 3 (le sequenze LBL *A, LBL *B, ..., LBL *Z); quest'ultimo programma è rimasto invariato nella versione presentata in questo numero. È

interessante il modo in cui viene operata la scomposizione della stringa e il successivo richiamo indiretto. Vediamo passo per passo cosa succede, per esempio scomponendo la parola test (il contenuto della stringa ALPHA, ai passi 12 ÷ 18, viene memorizzato in 4 registri, R07, R08, R09, R10, contenenti ciascuno 6 caratteri):

- 25 "... " Pone nel registro ALPHA 4 blank e un asterisco
- 26 ARCL Aggiunge ad ALPHA il contenuto del registro R07
- IND 11 "TEST" (i primi 6 caratteri)
- 27 ASTO X Memorizza in X i primi 6 caratteri di ALPHA "...T"
- 28 ASHF Toglie i primi 6 caratteri di ALPHA che quindi diventa "EST".
- 29 ASTO Memorizza in R07 la parola mancante del primo carattere
- IND 11 RICHAMA in ALPHA il contenuto del registro R07
- 30 "... " Pone in ALPHA due blank
- 31 ARCL X Richiama in ALPHA il contenuto del registro R07

Buffer per Megaprint - MGP

01*LBL *MGP*	30 " "	59 ST/ 04	88*LBL 01	118 1
02*LBL A	31 ARCL X	60 1.009	89 2	119 SKPCHR
03 ADV	32 ASHF	61 STO 05	90 SKPCHR	120 X<>Y
04 ADV	33 ASTO X	62*LBL 08	91 RCL 00	121 ACCHR
05 SF 12	34 "*"	63 1.004	92 ACCHR	122 RTN
06 31	35 ASTO Y	64 STO 06	93 RTN	123*LBL 06
07 STO 00	36 X=Y?	65*LBL 09	94*LBL 02	124 RCL 00
08 "WORDS ?"	37 GTO 12	66 RCL IND 06	95 1	125 ACCHR
09 AOH	38 XEQ IND X	67 INT	96 SKPCHR	126 ACCHR
10 STOP	39 ISG 12	68 LASTX	97 RCL 00	127 1
11 AOFF	40 GTO 11	69 FRC	98 ACCHR	128 SKPCHR
12 ASTO 07	41 ISG 11	70 10	99 1	129 RTN
13 ASHF	42 GTO 10	71 *	100 SKPCHR	130*LBL 07
14 ASTO 08	43*LBL 12	72 STO IND 06	101 RTN	131 RCL 00
15 ASHF	44 CF 12	73 XEQ IND Y	102*LBL 03	132 ACCHR
16 ASTO 09	45 BEEP	74 ISG 06	103 1	133 ACCHR
17 ASHF	46 RTN	75 GTO 09	104 SKPCHR	134 ACCHR
18 ASTO 10	47*LBL *P*	76 PRBUF	105 RCL 00	135 RTN
19 7.010	48 STO 01	77 ISG 05	106 ACCHR	136*LBL "*" "
20 STO 11	49 RDN	78 GTO 08	107 ACCHR	137 ADV
21*LBL 10	50 STO 02	79 FS? 00	108 RTN	138 ADV
22 1.006	51 RDN	80 RTN	109*LBL 04	139 ADV
23 STO 12	52 STO 03	81 ADV	110 RCL 00	140 ADV
24*LBL 11	53 RDN	82 ADV	111 ACCHR	141 ADV
25 " "	54 STO 04	83 RTN	112 2	142 ADV
26 ARCL IND 11	55 1 E6	84*LBL 00	113 SKPCHR	143 ADV
27 ASTO X	56 ST/ 01	85 3	114 RTN	144 ADV
28 ASHF	57 ST/ 02	86 SKPCHR	115*LBL 05	145 ADV
29 ASTO IND 11	58 ST/ 03	87 RTN	116 RCL 00	146 RTN
			117 ACCHR	147 .END.

Buffer per Megaprint - PGM			
01*LBL "PGM"	19 " 0"	37 STO 19	55 ISG 18
02*LBL A	20 ASTO 01	38*LBL 11	56 GTO 10
03 ADV	21 " 0 "	39 " *	57*LBL 12
04 ADV	22 ASTO 02	40 ARCL IND 18	58 CF 12
05 SF 12	23 " 00"	41 ASTO X	59 BEEP
06 *WORDS ?	24 ASTO 03	42 ASHF	60 RTH
07 AGN	25 " 0 "	43 ASTO IND 18	61*LBL "P"
08 STOP	26 ASTO 04	44 " "	62 STO 08
09 AOFF	27 " 0 0"	45 ARCL X	63 RDN
10 ASTO 14	28 ASTO 05	46 ASHF	64 STO 09
11 ASHF	29 "00 "	47 ASTO X	65 RDN
12 ASTO 15	30 ASTO 06	48 "**"	66 STO 10
13 ASHF	31 "000"	49 ASTO Y	67 RDN
14 ASTO 16	32 ASTO 07	50 X=Y?	68 STO 11
15 ASHF	33 14.017	51 GTO 12	69 1 E0
16 ASTO 17	34 STO 10	52 XE0 IND X	70 ST/ 08
17 " "	35*LBL 10	53 ISG 19	71 ST/ 09
18 ASTO 00	36 1.006	54 GTO 11	72 ST/ 10
			73 ST/ 11
			74 1.009
			75 STO 12
			76*LBL 08
			77 8.011
			78 STO 13
			79 CLA
			80*LBL 09
			81 RCL IND 13
			82 INT
			83 LASTX
			84 FRC
			85 10
			86 *
			87 STO IND 13
			88 ARCL IND Y
			89 ISG 13
			90 GTO 09
			91 ACA
			92 PRUF
			93 ISG 12
			94 GTO 08
			95 FS? 00
			96 RTN
			97 ADV
			98 ADV
			99 RTN
			100*LBL "+ "
			101 ADV
			102 ADV
			103 ADV
			104 ADV
			105 ADV
			106 ADV
			107 ADV
			108 ADV
			109 ADV
			110 RTN
			111 END

contenuto di X (vedi passo 27); ora ALPHA contiene "...T" (6 blank)

32 ASHF Toglie i 6 blank iniziali dal registro ALPHA

A questo punto nel registro ALPHA si trovano due caratteri: un asterisco, seguito dal primo carattere isolato dalla stringa; essi vengono utilizzati per richiamare indirettamente la routine del programma di codifica relativa al carattere da stampare. L'operazione, controllata da un contatore, viene ripetuta 6 volte finché viene esaurito il contenuto del primo registro (R07), dopodiché il loop si ripete per R08, R09, R10. Il programma si arresta dopo l'ultimo carattere impostato inizialmente in ALPHA.

La sequenza LBL "P" è rimasta pressoché identica alla versione originale, a parte gli RCL00 (1 byte) usati per porre in X il numero 31 anziché riscriverlo ogni volta (2 byte).

Il programma "PGM" differisce da "MGP" nel sistema usato per la preparazione delle righe da stampare; in "MGP" l'operazione avviene "caricando" il buffer di riga con caratteri 31 (del set della stampante) alternati a spazi vuoti, per mezzo delle istruzioni "ACCHR" e "SKPCHR", mentre nel programma "PGM" vengono utilizzati gruppi di "0" e blank posti nei registri da R00 a R07.

Dello stesso Cavallini, pubblichiamo anche la routine "CLRGX", un programma che non richiede altri commenti.

Cancela Registri: GLRGX

01*LBL "CLRGX"	05 ISG Y
02 0	06 GTO 00
03*LBL 00	07 RTN
04 STO IND Y	08 END

Megaprint			
01*LBL "A"	28 77444 E4	55 RCL T	82*LBL "H"
02 467333764	29 77111 E4	56 GTO "P"	83 774000077
03 770000077	30 776666666	57*LBL "J"	84 773764077
04 773333377	31 GTO "P"	58 3337733	85 770013777
05 770000077	32*LBL "F"	59 7700	86 770000077
06 GTO "P"	33 773333333	60 660007700	87 GTO "P"
07*LBL "B"	34 77444 E4	61 137673100	88*LBL "0"
08 773333764	35 77111 E4	62 GTO "P"	89 467333764
09 774444473	36 77 E7	63*LBL "K"	90 770000077
10 771111176	37 GTO "P"	64 770004673	91 ENTER†
11 776666731	38*LBL "G"	65 774673100	92 ENTER†
12 GTO "P"	39 467333764	66 771376400	93 137666731
13*LBL "C"	40 770000011	67 770001376	94 GTO "P"
14 467333764	41 770003377	68 GTO "P"	95*LBL "P"
15 770000011	42 137666731	69*LBL "L"	96 773333764
16 770000044	43 GTO "P"	70 77 E7	97 770000477
17 137666731	44*LBL "H"	71 ENTER†	98 773333310
18 GTO "P"	45 770000077	72 ENTER†	99 77 E7
19*LBL "D"	46 774444477	73 ENTER†	100 GTO "P"
20 773333764	47 771111177	74 776666666	101*LBL "0"
21 770000077	48 RCL 2	75 GTO "P"	102 467333764
22 ENTER†	49 GTO "P"	76*LBL "M"	103 770000077
23 ENTER†	50*LBL "I"	77 764000467	104 770066477
24 776666731	51 77777770	78 773767377	105 137667376
25 GTO "P"	52 777 E3	79 770010077	106 GTO "P"
26*LBL "E"	53 ENTER†	80 770000077	107*LBL "R"
27 773333333	54 ENTER†	81 GTO "P"	108 773337640
			109 770004770
			110 773377500
			111 770001376
			112 GTO "P"
			113*LBL "S"
			114 467333764
			115 376444411
			116 441111376
			117 137666731
			118 GTO "P"
			119*LBL "T"
			120 777777777
			121 777 E3
			122 ENTER†
			123 ENTER†
			124 GTO "P"
			125*LBL "U"
			126 770000077
			127 ENTER†
			128 ENTER†
			129 ENTER†
			130 137666731
			131 GTO "P"
			132*LBL "V"
			133 770000077
			134 177404771
			135 3767300
			136 171 E3
			137 GTO "P"
			138*LBL "W"
			139 770000077
			140 770060077
			141 774777477
			142 37303730
			143 GTO "P"
			144*LBL "X"
			145 774000477
			146 37646730
			147 67313760
			148 771000177
			149 GTO "P"
			150*LBL "Y"
			151 774000477
			152 37606730
			153 1777 E0
			154 777 E3
			155 GTO "P"
			156*LBL "Z"
			157 333333377
			158 46310
			159 4631 E4
			160 776666666
			161 GTO "P"
			162 .END.

A proposito di Bytes.....

Sul numero 2 di "MCmicrocomputer", in questa rubrica, è riportata la tabella dei codici usati dalla 41 per memorizzare le varie istruzioni di un programma; ho inoltre cercato di descrivere il modo in cui tali codici vengono utilizzati a seconda dei casi (istruzioni da 1 byte, istruzioni da 2, 3 e così via). La cosa potrebbe sembrare fine a se stessa, poiché conoscere il linguaggio usato internamente dalla macchina ma non potersene servire servirebbe tutt'al più a soddisfare la curiosità di sapere "quello che succede dentro" e basta. Wickes, autore del libro "Synthetic Programming", indica il modo di manipolare i singoli byte per mezzo di una importantissima funzione: il "BYTE JUMPER". La cosa è interessante però, come anche Merelli affermava nella recensione del libro, il tutto appare di una certa difficoltà, a cominciare dal modo di ottenere il prezioso "Byte Jumper". Wickes indica, nel suo libro, come ottenere tale funzione: troppo laborioso, e poi in un primo tempo non era applicabile alla 41CV; in seguito, lo stesso autore pubblicava un'appendice in cui era indicato il modo di ottenere il Byte Jumper anche sulla 41CV, ma ancora il procedimento, sebbene più semplice del primo, scoraggiava i "quarantunisti" senza molto tempo a disposizione. Il desiderio di diventare anch'io "Synthetic programmer" si risvegliò con una telefonata che tempo fa giunse in redazione, proveniente dalla Svizzera; un nostro lettore, il signor Wyss Christian, mi suggeriva un modo abbastanza semplice per ottenere il Byte Jumper sia sulla 41C che sulla 41CV. Passo subito ad elencare le operazioni da compiere:

- 1) Provocate un "MEMORY LOST" accendendo la macchina mentre premete il tasto ← (backarrow)
- 2) Disponete la macchina in "SIZE 062" se avete la 41C senza moduli aggiuntivi di memoria, "SIZE 318" se avete la 41CV
- 3) Assegnate la funzione "DEL" al tasto "LN" e la funzione "ENTER†" al tasto "ENTER†" (sembra inutile ma bisogna farlo!); attivate il modo "USER"
- 4) Premete i seguenti tasti, senza sbagliare:

TASTI	DISPLAY
PRGM	00REG 00
←(backarrow)	00 REG 00
CAT I e immediatamente	
R/S	.END. REG 00
DEL 001 (usando il tasto LN)	4094 poi .END. REG 00
DEL 001	4093 DEC
GTO. 001	01†
GTO. 002	02 LBL 03
DEL 002	01†
ALPHA "A" ALPHA	00† A
GTO..	02 REG 00

A questo punto, al tasto "ENTER†" è assegnata una nuova funzione che sul display compare come "XROM 05,01": è il Byte Jumper. Se avete sbagliato a premere qualche tasto, ricominciate daccapo; se la macchina va in "CRASH" (cioè si blocca senza possibilità neanche di spengerla col tasto "ON") è sufficiente rimuovere le batterie e reinserirle poco dopo. Tutto ciò non è affatto dannoso per la 41, sebbene vada in "CRASH" abbastanza spesso e a volte sia necessario lasciarla tutta la notte senza batterie. A partire dal prossimo numero, impareremo ad usare questa eccezionale funzione per ottenere i risultati più strani e impensati....

P.G.