

MIX

di Massimo Tacchini (Modena)

Sono uno studente 22enne che frequenta l'ultimo anno di ingegneria, e nel farvi i miei più sinceri complimenti per la vostra ottima rivista, vi dirò che fin dal primo numero la seguo con vivo interesse, in quanto lascia spazio non solo ai personal veri e propri, ma anche a quelli che si sono travestiti da calcolatrici: parlo naturalmente dell'HP 41 che io, come molti oggi, ho la fortuna di possedere.

Solo ora però, ho trovato quella briciola di coraggio necessaria per spedirvi un mio programma; la scelta non si presentava delle più semplici, tuttavia, dopo aver eliminato quei programmi che potevano avere un ambito specifico e di conseguenza interessare ben pochi lettori, non erano rimasti che i giochi; e per non annoiarvi con i soliti giochi più o meno machiavellici ho pensato di inviarne uno particolarmente semplice, ma "condito" in modo talmente ricco da farlo, spero, apprezzare.

Ma incominciamo a parlarne.

Il gioco. Le regole

Il gioco "MIX", noto forse con altri nomi, consiste nell'ordinare in fila crescente

le N cifre di un numero, cifre che sono una diversa dall'altra e comprese tra 1 ed N, con N che al max può valere 9; unica mossa a disposizione è quella che permette di invertire l'ordine delle prime P cifre del numero.

Un facile esempio chiarirà meglio: supponiamo di aver dimensionato N=5 e che il calcolatore abbia proposto come numero da risolvere 45321; se ora noi impostiamo 2 nel calcolatore, esso invertirà le prime 2 cifre fornendo il nuovo numero 54321; impostando 5, esso invertirà tutto il numero, restituendo 12345 che è il nostro punto di arrivo.

Come si vede il gioco è quasi banale, tuttavia, poiché è più bravo naturalmente chi impiega meno mosse, la ricerca del percorso più breve per giungere alla soluzione favorisce la nascita di diverse tattiche.

Il programma lavora su numeri al max di nove cifre, ma nulla vieta di ampliarne di una cifra i numeri con l'aggiunta dello zero, o addirittura di passare alle lettere dell'alfabeto; in quest'ultimo caso lo scopo del gioco sarebbe quello di mescolare le lettere fino a porle in ordine alfabetico o in modo da formare una parola; le modifiche renderebbero il programma più lungo e un po' più lento. Ma ne riparleremo.

Come si gioca

Dopo aver caricato il programma in memoria e aver introdotto un seme nel registro 00, iniziate con XEQ MIX; la prima cosa che il calcolatore vi chiede è il numero dei giocatori (player): il max possibile, lo dico solo per dovere, è 119 con la CV e 23 con la C se fornita di almeno un modulo RAM; premuto R/S se la memoria dati così ripartita è insufficiente appare la scritta "SIZE nnn" ed è necessario eseguire la funzione SIZE nnn prima di continuare; quei fortunati (io no purtroppo) che posseggono il modulo di estensione delle funzioni possono impostare direttamente nel programma la funzione PSIZE.

Al punto successivo l'HP chiede i nomi dei giocatori, uno alla volta; superfluo dire, forse, che tali nomi devono essere al max di 6 lettere, che è appunto il max memorizzabile in un registro.

Proseguendo compare la richiesta di dimensionare N, cioè sostanzialmente da quante cifre, deve essere composto il numero; l'ultima richiesta è "CASUAL ? Y/N": se rispondete "Y" (yes) l'HP ricerca autonomamente e casualmente il numero, se rispondete "N" (no) compare il simbolo "f", il quale sta ad indicare che potete impostare un numero a vostro piacimento purché naturalmente sia di N cifre una diversa dall'altra e tale che, se avete ad es. dimensionato N=6, non compaiano mai il 7, l'8 e il 9.

Ora si entra nel vivo del gioco: l'HP mediante una scritta alfabetica e un segnale acustico avverte il primo giocatore che è

Tasti	Visore	Commenti
	seme STO 00	
XEQ MIX	N OF PLAYERS	impostare n. giocatori
2 R/S	NAME OF	
	PLAYER 1	impostare nomi giocatori
MARIO R/S	PLAYER 2	
LUCA R/S	N?	dimensionare il numero
6 R/S	CASUAL ? Y/N	decidere il criterio di scelta del numero (casuale -Y- oppure no -N-)
Y R/S	MARIO	invita al gioco Mario
	6 3 4 5 2 1	propone il numero
2 R/S	3 6 4 5 2 1	
4 R/S	5 4 6 3 2 1	
2 R/S	4 5 6 3 2 1	
3 R/S	6 5 4 3 2 1	
6 R/S	STEPS : 5	n. mosse impiegate da Mario
	LUCA	invita al gioco Luca
	6 3 4 5 2 1	ripropone il numero
4 R/S	5 4 3 6 2 1	
3 R/S	3 4 5 6 2 1	
4 R/S	6 5 4 3 2 1	
6 R/S	STEPS : 4	n. mosse impiegate da Luca
	RESULT	
	LUCA 4	classifica finale
	MARIO 5	
	N ?	ricomincia per una nuova gara tra Mario e Luca

Figura 1

Programma Mix				
01 *LBL "MIX"	51 R	101 RDN	151 *LBL 00	201 RCL 14
02 FIX 9	52 *LBL 01	102 ISG Y	152 *Y?Y?	202 INT
03 CF 24	53 STO IND Y	103 GTO 04	153 GTO 06	203 Z
04 *N OF PLAYERS	54 ISG Y	104 RCL 10	154 I	204 +
05 PROMPT	55 GTO 01	105 GTO 04	155 RCL 01	205 RCL 15
06 Z	56 *Y*	106 *LBL 03	156 *X?Y?	206 FRC
07 *	57 STO Y	107 RCL 13	157 GTO 05	207 +
08 STO 14	58 RDN	108 STO IND Z	158 RCL 10	208 STO 15
09 17	59 *CASUAL ? Y/N*	109 BSE 13	159 STO Z	209 ISG 14
10 *	60 STOP	110 GTO 03	160 ISG Z	210 RCL IND 14
11 SF 25	61 RSTO X	111 RCL 12	161 *LBL 07	211 *LBL 17
12 *SIZE *	62 ROFF	112 STO 13	162 RDN	212 ISG 15
13 ARCL X	63 *Y?Y?	113 BSE 13	163 ISG X	213 RCL IND 15
14 I	64 GTO 03	114 0	164 *LBL 00	214 *Y?Y?
15 -	65 *Y*	115 *LBL 13	165 RCL IND X	215 *X?Y?
16 RCL IND X	66 PROMPT	116 RCL IND 14	166 *Y?Y?	216 GTO 00
17 FCIO 25	67 STO 11	117 10	167 GTO 05	217 STO IND 15
18 PROMPT	68 *LBL 02	118 RCL 13	168 ISG Z	218 RDN
19 XEQ 12	69 10	119 *Y*	169 GTO 07	219 STO IND 14
20 *NAME OF*	70 /	120 *	170 BEEP	220 RCL 14
21 *VIEW	71 ENTER*	121 +	171 *STEPS *	221 I
22 PSE	72 FRC	122 BSE 13	172 ARCL IND 15	222 -
23 I	73 -	123 *LBL 00	173 *VIEW	223 RCL 15
24 RDN	74 STO 14	124 ISG 14	174 ISG 15	224 I
25 CLB	75 LASTX	125 GTO 11	175 GTO 00	225 -
26 *LBL 13	76 10	126 STO 11	176 PSE	226 RCL IND X
27 *PLAYER *	77 *	127 *LBL 14	177 *RESULT*	227 *X? IND Z
28 ARCL X	78 STO IND 13	128 CLR	178 *VIEW	228 STO IND Y
29 STOP	79 RCL 14	129 ARCL IND 15	179 XEQ 12	229 RCL IND 14
30 STO IND 15	80 BSE 13	130 *VIEW	180 RCL 15	230 *LBL 00
31 ISG X	81 GTO 02	131 *TONE 9	181 *002	231 ISG 15
32 *LBL 00	82 GTO 14	132 PSE	182 -	232 GTO 17
33 ISG 15	83 *LBL 03	133 ISG 15	183 STO 14	233 ISG 14
34 0	84 RCL 00	134 *LBL 05	184 GTO 16	234 GTO 16
35 STO IND 15	85 RES	135 RCL 10	185 *LBL 00	235 XEQ 12
36 RDN	86 9821	136 CLR	186 RCL 12	236 *LBL 15
37 ISG 15	87 *	137 *LBL 09	187 STO 13	237 CLR
38 GTO 12	88 *211327	138 ARCL IND X	188 RCL 11	238 ARCL IND 15
39 ROFF	89 +	139 ISG X	189 GTO 02	239 *Y*
40 *LBL 05	90 FRC	140 GTO 09	190 *LBL 12	240 ISG 15
41 XEQ 12	91 STO 00	141 I	191 RCL 16	241 0
42 *N ?*	92 RCL 12	142 PROMPT	192 16	242 *X? IND 15
43 PROMPT	93 *	143 ISG IND 15	193 +	243 ARCL X
44 STO 12	94 INT	144 *LBL 06	194 I E3	244 *VIEW
45 STO 13	95 RCL 10	145 RCL IND X	195 /	245 PSE
46 I E3	96 +	146 *X? IND Z	196 17	246 PSE
47 /	97 *LBL 04	147 STO IND Y	197 +	247 ISG 15
48 ISG X	98 RCL IND X	148 RDN	198 STO 15	248 GTO 15
49 STO 10	99 *X?*	149 BSE X	199 RTN	249 GTO 00
50 STO 14	100 GTO 03	150 ISG Y	200 *LBL 16	250 *END.

il suo turno di gioco e visualizza il numero (da lei o da voi scelto) che sarà lo stesso per

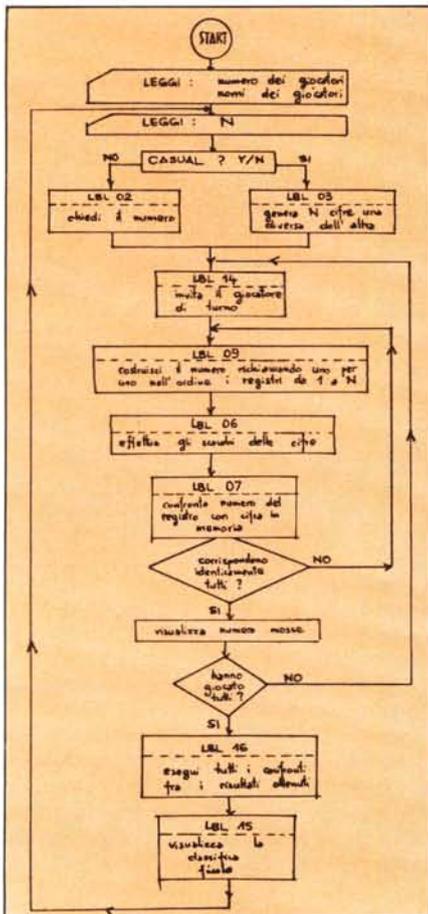


Diagramma di flusso del programma Mix

tutti i giocatori, i quali hanno così modo di confrontare la loro abilità.

Alla comparsa del numero, basta impostare la quantità di cifre che si vuole invertire e premere R/S; l'operazione si ripete fino a che non si sono ordinate le cifre da 1 a N, dopo di che l'HP fornisce il numero "mm" di mosse impiegate secondo il messaggio "STEPS: mm" accompagnato da un BEEP, ed invita il secondo giocatore ripresentando il numero di partenza.

Dopo che tutti i giocatori hanno esaurito il loro turno, in pochi secondi l'HP fornirà (sorpresa!) la classifica finale in ordine crescente a partire dal giocatore che ha impiegato meno mosse; quindi si riparte dalla richiesta di dimensionamento N del numero, per una nuova gara fra gli stessi giocatori.

Per dissipare gli ultimi dubbi vedi un esempio riportato nella figura 1.

Registri di memoria

Il programma occupa 64 registri (esattamente due schede magnetiche); in più vi sono 17 registri di memoria utilizzati come segue:

00	numero casuale
01 ÷ 09	cifre del numero da ordinare
11	numero da ordinare
12 e 13	N
16	numero dei giocatori

10, 14 e 15 contatori

Inoltre, dal registro 17 in poi sono occupati due registri di memoria per ogni giocatore, uno per il nome e l'altro per il numero delle mosse.

Detto quindi "gg" il numero dei giocatori, sono necessari complessivamente $64 + 17 + 2 \text{ gg}$ registri, condizione che rende indispensabile un modulo RAM per chi possiede il modello C.

Il programma

Analizziamo ora le varie parti del programma; sorvoliamo sull'input iniziale dei dati che, dovete ammetterlo, malgrado non possa più meravigliare perché le qualità dell'HP41 sono ormai arcinote, è sufficientemente curato (scusate la modestia).

Passiamo direttamente al momento della richiesta "CASUAL? Y/N": come si comporta il programma a seconda della risposta? Se rispondete N (no) appare "f" e dovete impostare il numero da voi scelto il quale viene memorizzato nel registro 11 e scomposto cifra per cifra nella successiva LBL 02, mediante l'ausilio del classico metodo della divisione per 10 e della funzione FRC; se rispondete Y (yes) il programma si porta alla LBL 03 (passo 83) dove vengono scelte le N cifre diverse; il problema è proprio questo: scegliere nel minor tempo possibile cifre tutte diverse.

Prima di continuare, osserviamo che se N è il numero max di cifre del numero, N è anche la cifra più grande che vi compare, poiché le cifre devono essere N e consecutive a partire dall'1.

Se ci si affida ad una generazione completamente casuale per la determinazione del numero che deve essere fatta, ovviamente, cifra per cifra, è facile capire come per ottenere N cifre di cui nessuna ripetuta si possa aspettare anche molto tempo; ho scelto allora, senza peraltro inventare nulla, una generazione pseudo-casuale nella quale si generano non le cifre del numero ma la loro posizione nel numero stesso.

Così, opportunamente azzerati i registri di memoria dall'1 all'N nei quali andranno poste le N cifre del numero, basta sorteggiare un registro di memoria e posizionarvi la cifra N; poi basta sorteggiare un altro registro e posizionarvi N-1 e, e così via fino ad N=1.

Per evitare ripetizioni, detto "R" il registro sorteggiato, basta ogni volta eseguire il test $RCL R X = 0?$

Se la risposta è negativa, vuol dire che nel registro è già stata posizionata una cifra e quindi tale registro non è più utilizzabile; basta allora incrementare $R \leftarrow R + 1$ ed eseguire di nuovo il test $RCL R X = 0?$ fino a che non si trova la risposta affermativa; questa risposta indica che il registro non è ancora stato utilizzato e vi può essere memorizzata la cifra corrente.

Così per l'ultima cifra da posizionare, che è sempre l'1 basta cercare l'unico registro vuoto e questo richiede, nella peggiore delle ipotesi, N interazioni. In questo modo è possibile giungere al risultato in un tempo relativamente breve ($15 \div 20$ secon-

di per N=9).

Mi rendo conto di avere dilungato un po' la spiegazione, ma, come sapete, descrivere a parole è più lungo che capire o immaginare; tuttavia l'ho fatto con l'unico scopo di illustrare abbastanza profondamente le tecniche usate per dar modo a chi legge di svolgere le proprie critiche ed individuare possibili errori. D'altra parte, penso che sia una cosa risaputa che i giochi sono un ottimo allenamento per imparare a programmare, in quanto nascono situazioni e problemi spesso non semplici, e la cui risoluzione può essere riutilizzata per programmi ben più seri. Ma continuiamo.

Un altro punto interessante è alla LBL 06 (passo 144) dove, dopo aver posizionato il numero 1 in Y e il numero P corrispondente alle prime cifre da invertire in X, si scambiano i contenuti dei registri 1 e P poi, mediante DSE X e ISG Y, dei registri 2 e P-1, e così via fino a che $Y \geq X$; ecco perché il numero viene precedentemente scomposto cifra per cifra; è possibile così con pochissime istruzioni (e quindi pochissimo tempo) invertire l'ordine delle prime P cifre; non solo, ma poiché le N cifre sono memorizzate nei primi N registri a partire dallo 01, il gioco è finito se e solo se nel registro 01 c'è la cifra 1, nello 02 la cifra 2 e così via; in tal modo se da un lato si spreca alcuni registri quando $N < 9$, dall'altro è facilissimo verificare se il gioco è terminato: basta posizionare nella catasta il numero 1 che fa da contatore, eseguire RCL IND X poi il test $X \neq Y?$; se sì il numero non comincia per 1 e il gioco non è sicuramente finito, se no basta eseguire RDN ISG X (e l'1 è diventato 2) e ripetere il test dopo RCL IND X; se tutti i registri e le cifre coincidono il gioco è terminato e viene visualizzato il numero di mosse impiegato.

Infine un breve cenno all'ultima parte del programma, dalla LBL 16 (passo 200) in poi dove cioè si determina la classifica finale la quale, nel caso di parità, privilegia chi ha giocato per primo premiando, come è giusto, chi per primo ha raggiunto un certo risultato.

Il metodo di ordinamento usato è certamente uno dei più lunghi ma più semplici; in questo modo si hanno non troppe istruzioni e poiché il numero dei giocatori è nella maggior parte dei casi molto esiguo, l'attesa dei risultati è breve.

Individuato il risultato del 1° giocatore (RCL 18) lo si confronta con tutti gli altri facendo uno scambio (sia di punti che di nomi ovviamente) ogni volta che si trova un punteggio minore. Alla fine del primo ciclo si è sicuri di aver determinato il vincitore, cioè colui che ha il numero più basso di mosse; si passa al risultato posizionato nella seconda memoria (RCL 20) e lo si confronta coi successivi ripetendo le stesse precedenti operazioni; e così via fino ad esaurire tutti i risultati.

Suggerimenti e modifiche

Alcuni lettori avranno certamente una

CONTABILITA' GENERALE E IVA

CONTABILITA' SEMPLIFICATA

FATTURAZIONE

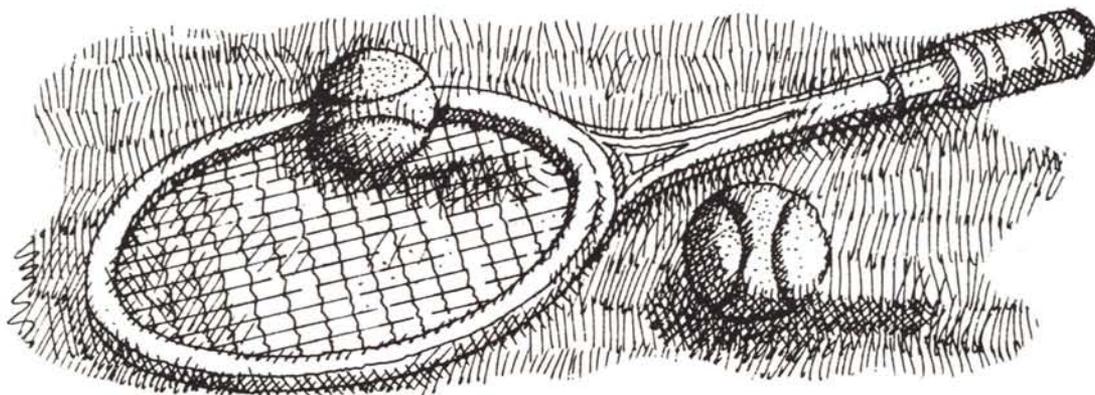
WORD PROCESSING

GESTIONE DEL PERSONALE

GESTIONE ORDINI
DA CLIENTI A FORNITORI

MAGAZZINO
CONTABILE E FISCALE

liberati dai problemi, dalli al Computer!



Microcomputer **B2** Buffetti

Nato con "Forte personalità" e tanta esperienza in problemi d'ufficio

B2 un microcomputer potente
La moderna tecnologia consente, attraverso i processi di miniaturizzazione, di abbattere i costi di fabbricazione. La potenza dei minicomputers di ieri, è oggi presente nel nostro micro a metà prezzo.

B2 un microcomputer affidabile
Un manciata di chips componenti basta oggi per costruire un computer intero. Meno componenti, meno possibilità di guasti. L'intero progetto è finalizzato ad ottenere un prodotto di qualità professionale senza compromessi.

B2 un microcomputer che cresce
E' l'inizio di una famiglia di sistemi via via più potenti e compatibili. Le vostre esigenze cresceranno con la Vostra azienda e B2 crescerà per adeguarsi.



MC **Desidero ricevere:**

Materiale illustrativo sul vostro nuovo Microcomputer B2

La visita di un vostro funzionario commerciale

SIG. _____ N° _____

VIA _____ CITTA' _____

TELEFONO _____

**INVIARE A BUFFETTI DATA S.P.A.
P.LE VITTORIO BOTTEGO, 51/ROMA**

Microcomputer Made in Italy per l'ufficio italiano

Roma, P.le V. Bottego, 51
tel. (06) 5758343
Milano, Foro Buonaparte, 53
tel. (02) 870578/874325

Buffetti data

Napoli, Viale Kennedy, 395
tel. (081) 7602585
Firenze, Viale Pier Capponi, 42
tel. (055) 572923