

PLAY ON TAPE

COMPUTER

manuale 9

**VIC 20 - C. 16
PLUS/4**



**LINGUAGGIO
MACCHINA**

- Status register
- Istruzioni aritmetiche

**PSEUDOCODICE:
INTELLIGENZA
ARTIFICIALE**

Intelligenza artificiale: questo l'argomento trattato nella lezione di pseudocodice che appare in questo numero. L'importanza e l'attualità dell'argomento sono note a tutti: l'intelligenza artificiale è fondamento basilare della cultura informatica. La lezione di Linguaggio/Macchina tratta dello Status Register e dei Flags per affrontare le istruzioni aritmetiche della CPU e per conoscere le basi necessarie per un apprendimento più completo di nuove istruzioni.

Sulle tracce del suono: pubblichiamo la seconda parte della monografia (importantissima!) sulle capacità sonore del vostro computer.

PLAY ON TAPE Computer n. 9: un numero fondamentale, com'è evidente. Nella cassetta troviamo come sempre programmi-giochi che uniscono l'utile al dilettevole, che aiutano ad approfondire divertendo i concetti della scienza informatica contenuti nelle pagine del manuale.

Dal lato del Vic 20 i programmi sono: Ippodromo (vi farà provare il divertimento di chi assiste a una gara ippica e il brivido, innocentissimo, della scommessa), Vic Vic Bang (divertente e autoironica roulette russa), Le tre carte (per misurare fortuna e colpo d'occhio), Power (videogioco spaziale) e Conversioni (un programma utilissimo a tutti, studenti e non).

Dal lato C.16 e PLUS/4 la cassetta presenta: Vocabolario (per avere un pratico e favoloso dizionario poliglotta a computer), Istogrammi 3D (per rappresentare graficamente gruppi di dati e informazioni), Chitarra facile (un metodo estremamente pratico per imparare a suonare il più popolare strumento a corda), Filetto (una bella versione del noto gioco di abilità) e Parol (un gioco davvero divertente che può animare le vostre serate).

Nel prossimo numero di PLAY ON TAPE (il fatidico numero 10 che completa la prima ideale raccolta necessaria ad ogni appassionato Vic 20, C.16 e PLUS/4), dovremmo superare noi stessi, anche e soprattutto grazie alle segnalazioni, idee e le richieste dei nostri lettori.

Arrivederci a presto.

Editoriale VIDEO - Direttore: Antonio Lucarella - Coordinamento tecnico: Roberto Treppiedi - Hanno collaborato: Franco Longoni, Massimo Cellini, Fabrizio Iotti, Maria Russino, Aldo Campanozzi, Silvana Buttà, Cinzia Agiman, Domenico Cellamare, Roberta Di Pietro, Alessandro Vallone, Arnaldo Restelli, Dino Ticli, Igino Zaffaina, Giovanna Zampella - In copertina: disegno di Aldo Campanozzi - Stampa: Color Graf Milano - Fotocomposizione: ERREGI Milano - reg. del Trib. di Milano n. 518 del 10/11/84 - Stampato in Italia

PSEUDOCODICE E INTELLIGENZA ARTIFICIALE

INTRODUZIONE

In questo nono articolo, tenteremo di illustrare la potenza di progettazione dello pseudocodice applicato ai problemi più "difficili" che conosciamo: il campo di indagine della "INTELLIGENZA ARTIFICIALE".

Lo pseudocodice serve per progettare programmi di qualsiasi natura. Il vantaggio principale consiste nell'offrire uno strumento di analisi dei problemi, oltre che come mezzo di espressione per le soluzioni trovate. Con lo pseudocodice possiamo affrontare problemi che richiedono per la soluzione sia artifici di programmazione (cioè algoritmi) sia organizzazione dei dati (strutture dati). Nello scorso articolo abbiamo visto che anche la programmazione con files è dominabile attraverso lo pseudocodice. Come al solito, oltre a presentare molti esempi e a spiegare i termini base, ci proponiamo di risolvere con lo pseudocodice ed il BASIC un minuscolo problema di Intelligenza Artificiale: il gioco perfetto del PUNTO E CROCE. In inglese, questo famoso gioco dei 9 quadrati, si chiama "TIC-TAC-TOE", ed è stato studiato proprio perché offre lo spunto per interessanti osservazioni nel campo della teoria dei GIOCATORI AUTOMATICI, in cui rientra per esempio il Giocatore di Scacchi.

INTELLIGENZA ARTIFICIALE

Può sembrare strano, ma non esiste una definizione unica e completa di che cosa si intenda per Intelligenza Artificiale, per cui è senz'altro più facile fare esempi concreti di questo tipo di problema e darne un'idea intuitiva.

In inglese Intelligenza Artificiale si traduce "Artificial Intelligence" e perciò si usa la sigla internazionale A.I. come abbreviazione. Una proposta di definizione è comunque la seguente, che ha il vantaggio di mettere in luce dove sta la difficoltà di una definizione:

A.I. è lo studio di come programmare un computer in modo che faccia quelle cose che per il momento riescono meglio agli essere umani.

Da questa definizione, data da E. BICH (1982), risulta chiaro che l'A.I. si interessa di problemi che siano al limite della capacità di soluzione algoritmica, cioè attraverso un programma. Ciò equivale a dire che non appena per un problema si trova un buon algoritmo (per esempio un ottimo Giocatore di Scacchi), l'A.I. non ha più nulla da dire. E questo è proprio quello che intendiamo per INTELLIGENZA: quando un compito è così semplice che possiamo farlo senza problemi, non c'è più nulla di "intelligente" da scoprire.

Esamiano alcuni esempi di problemi che hanno attirato l'attenzione degli specialisti A.I.

Uno dei primi problemi studiati è stato "COME FAR GIOCARE IL COMPUTER AD UN GIOCO" e per questo furono scritti programmi che giocavano a scacchi, e addirittura imparavano a migliorare le proprie strategie di gioco ricordando le partite già giocate. I giochi sono particolarmente importanti per l'A.I. Infatti, sebbene le persone quando giocano dimostrino di saper applicare molta "intelligenza" (nella scelta delle mosse, della strategia, eccetera) i computers sono così veloci che possono competere per il semplice fatto che esplorano molte soluzioni in un tempo minimo scegliendo poi la migliore. Nel gioco degli scacchi, per esempio, un computer può prevedere fino a 10 o 15 mosse successive e questo gli permette di fare la scelta della mossa più astuta. Questo approccio è oggi però sorpassato. Può servire per prodotti commerciali, ma lo specialista A.I. studia tecniche alternative che gli permettano di escludere delle strade quando sia "ovvio" che non conducono a nulla. È così che agisce il grande campione di scacchi, che non perde certo tempo a esaminare ogni possibilità, anche la più stupida, prima di prendere una decisione, ma "vede" subito quelle tre o quattro alternative che si presentano come le più invitanti. È questa capacità di "comprendere" che interessa lo specialista A.I. e che questi tenta di riprodurre con un programma.

Oggi l'A.I. ha inglobato anche un altro problema oltre a quello di far giocare il computer: quello di rendere le macchine in grado di muoversi con intelligenza e trarre conclusioni da informazioni memorizzate.

Supponiamo per esempio, di leggere un articolo di giornale che riporta una notizia:

"Il cockerino della madre del Principe Carlo di Inghilterra è

IN EDICOLA

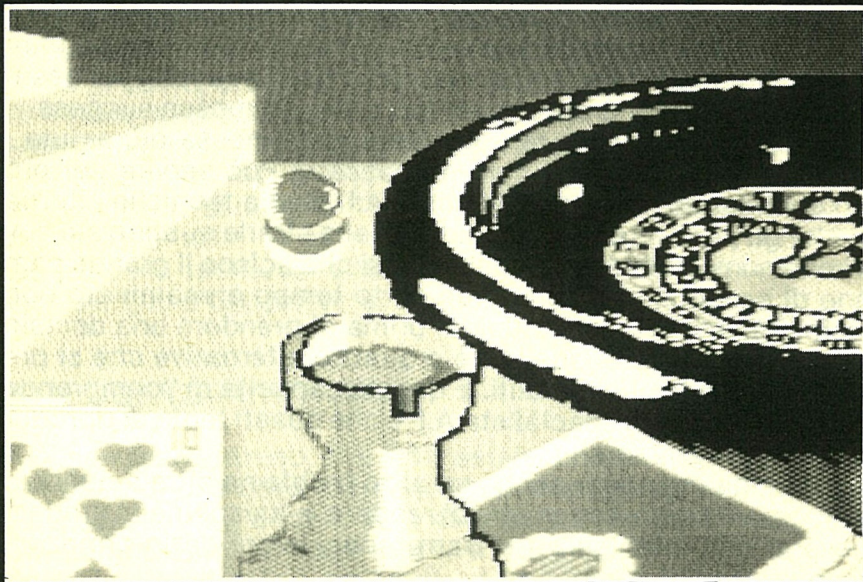
PER POSSESSORI DI

L. 8.000

CBM 64 - C.16 - PLUS/4

I GIOCHI DI SOCIETÀ

**SCACCHI - ROULETTE - POKER -
BRISCOLA - DAMA - TOMBOLA -
BLACK JACK - REVERSE**



EDITORIALE VIDEO

stato trovato abbandonato presso il parco di Hyde”

Vediamo cosa ne saprebbe ricavare un computer semplicemente memorizzando il brano secondo un programma di classe “normale” (cioè non A.I.) che si chiama “Information Retrieval”, e che si basa sul tener traccia delle parole del testo per rispondere a domande. Questo programma si usa per esempio in alcune sedi di giornali, per rintracciare notizie già memorizzate.

Un tale programma saprebbe rispondere alle seguenti domande:

- Un COCKERINO è stato trovato?
- Ad HYDE cosa è successo recentemente?
- C'è una notizia che riguarda sia la MADRE DI CARLO e il parco di HYDE?

A tutte queste domande il programma di Information Retrieval saprebbe rispondere, presentando la notizia scritta sopra. Infatti la notizia è espressa in modo che individuando le parole chiave si arrivi a selezionarla tra le altre notizie (le parole chiave sono scritte in maiuscolo nella loro formulazione). Un essere umano ricaverebbe, invece, un maggior numero di informazioni. Per esempio:

1. È stato perduto un cockerino?
2. La madre di Carlo di Inghilterra possiede un cane?
3. La madre di Carlo è ancora viva?

Alla prima domanda sappiamo rispondere perché se un cocker viene TROVATO noi sappiamo che precedentemente deve essere stato PERDUTO; alla seconda ci arriviamo perché sappiamo che un cockerino è un tipo di CANE, ed infine all'ultima domanda sapremmo rispondere (se la notizia è recente) osservando che non vi è scritto nulla che lasci presumere che questa signora sia morta. Tutte queste osservazioni umane dimostrano che il patrimonio di conoscenze che ci portiamo dentro non è fine a sè stesso, ovvero “nozionistico”, ma è un bagaglio di conoscenze che noi sappiamo SFRUTTARE CON INTELLIGENZA.

Quest'ultima e recente disciplina dell'A.I., si chiama costruzione di sistemi esperti, in inglese Expert Systems, e sta dando i più grandi contributi alla A.I., specialmente dal punto di vista commerciale.

I campi in cui gli Expert System si sono particolarmente distinti sono stati: l'analisi di immagini radar o radiografie, la diagnosi medica automatica, l'analisi di sostanze chimiche, la ricerca di guasti in apparecchiature complesse a partire dalla descrizione dei malfunzionamenti. Un problema è da sempre però la spina nel fianco dell'A.I.: da 20 anni si cerca di scrivere un programma

- 1. Riempire il secchio da 4 litri
- 2. Riempire il secchio da 3 litri
- 5. Svuotare il secchio da 4 litri completamente (a terra)
- 6. Svuotare il secchio da 3 litri completamente (a terra)
- 7. Versare acqua dal secchio da 3 litri in quello da 4 finché quest'ultimo è pieno.
- 8. Versare acqua dal secchio da 4 litri in quello da 3 finché quest'ultimo è pieno.
- 9. Versare tutta l'acqua dal secchio da 3 litri nel secchio da 4.
- 10. Versare tutta l'acqua dal secchio da 4 litri nel secchio da 3.

Le mosse possibili sono state numerate, in ordine crescente, saltando due numeri, per indicare che se ne potrebbero aggiungere anche altre. A questo punto risulta chiaro che ciò che manca è una STRATEGIA DI CONTROLLO che per ogni stato del sistema, a partire da (0,0) scelga tra le regole descritte sopra quella che fa avvicinare di più il sistema alla soluzione. Non abbiamo qui lo spazio per descrivere quale algoritmo permetterebbe di trovare la successione di mosse atte a risolvere il problema. Diamo invece un esempio di soluzione per dimostrare come il problema è risolubile e spingere il lettore ad analizzare la risposta per comprendere quale sia la tecnica con cui un programma potrebbe ragionare. Per inciso, questa soluzione è stata effettivamente trovata da un programma...

Secchio da 4 litri	Secchio da 3 litri	Regola applicata
0	0	
0	3	2
3	0	9
3	3	2
4	2	7
0	2	5
2	0	9

IL TIC-TAC-TOE

Per risolvere il problema del PUNTO E CROCE, cominciamo col descrivere la situazione in forma computerizzabile. Pensiamo alla scacchiera di gioco come ad un vettore di 9 elementi dove le posizioni della scacchiera siano le seguenti:

1	2	3
4	5	6
7	8	9

Ciascun elemento del vettore (variabile con indice) conterrà il valore 2 (per indicare che la corrispondente casella è vuota) 3 per indicare una croce (X), 5 per indicare un punto (O). Questa scelta di valore si giustificherà tra un attimo, quando osserveremo che possiamo individuare le righe e le colonne critiche (per O e per X) semplicemente sommando i valori delle righe e delle colonne. Poi instauriamo un contatore per contare i turni di gioco: siamo facilitati dal fatto che non vi sono più di 9 turni di gioco. Per essere chiari chiamiamo "il Giocatore" il nostro programma, il cui simbolo è il punto, e "l'avversario" il suo antagonista umano.

Quali sono le mosse che modificano la situazione? Ovviamente una mossa per il Giocatore consiste nel mettere un punto in una certa casella, diciamo la casella n. Questa mossa la indichiamo con la frase GO(n), che significa: occupa la casella n con il simbolo del punto.

Quale è la strategia di controllo? La strategia, che proponiamo è un insieme di 9 casi che guidano le mosse, uno per turno. Il Giocatore esegue le mosse pari o dispari a seconda che giochi per primo o per secondo. Assumiamo la convenzione che il Giocatore gioca sempre con il punto, ed il suo avversario sempre con la croce. Prima di descrivere la strategia di gioco, descriviamo la "funzione di scelta", che ad ogni turno esamina la situazione e decide quale è la casella da occupare con il punto. Chiamiamo questa funzione POSSWIN:

- POSSWIN restituisce 0 nella variabile WINS se nessun giocatore può vincere a questo turno; altrimenti restituisce nella variabile WINS la casella che costituisce la posizione di vittoria. La logica è che se l'avversario può vincere, allora bisogna bloccarlo; se invece può vincere il Giocatore, bisogna giocare quella casella. POSSWIN opera facendo il prodotto dei valori di ognuna delle 3 colonne, delle tre righe e delle due diagonali della tabella, secondo il loro contenuto numerico definito in base alle convenzioni indicate. Come potete facilmente verificare, quando uno di questi prodotti vale 18, allora su quella colonna, riga o diagonale esiste una possibilità di vittoria per X (cioè l'avversario umano), mentre se tale prodotto è 50, esiste

possibilità di vittoria per il Giocatore. La casella di vittoria è quella che vale 2 (il valore di casella vuota) tra le 3 del prodotto considerato.

Adesso siamo in grado di esprimere la strategia di gioco, mossa per mossa:

turno = 1

GO(1) (angolo alto a sinistra)

turno = 2

SE casella (5) è vuota ALLORA GO(5), ALTRIMENTI GO(1)

turno = 3

SE casella (9) è vuota ALLORA GO(9), ALTRIMENTI GO(3)

turno = 4

ESEGUI POSSWIN.

SE WINS non è 0 ALLORA

GO(WINS) (Cioè blocca l'avversario)

ALTRIMENTI

PROVA A FARE 2 PUNTI IN FILA (MAKE 2)

FINE_SE

turno = 5

ESEGUI POSSWIN.

SE WINS non è 0, ALLORA GO(WINS) (cioè vinci o blocca)

SE WINS è 0 ALLORA

SE casella (7) è vuota ALLORA GO(7)

ALTRIMENTI

GO(3)

FINE_SE (Qui tentiamo di fare una forza)

turno = 6

ESEGUI POSSWIN.

SE WINS non è 0, ALLORA GO(WINS) (cioè vinci o blocca)

SE WINS è 0 ALLORA

SE casella (7) è vuota ALLORA GO(7)

ALTRIMENTI

PROVA A FARE 2 PUNTI IN FILA (MAKE 2)

FINE_SE

turno = 7, 8 o 9

ESEGUI POSSWIN.

SE WINS non è 0, ALLORA GO(WINS) (cioè vinci o blocca)

SE WINS è 0 ALLORA SCEGLI UNA QUALUNQUE CASELLA LIBERA

Alle mosse 5 e 6 si nomina la funzione MAKE 2, il cui scopo è quello di tentare di mettere 2 PUNTI in fila per il Giocatore. Si può realizzare tentando prima di giocare nel centro, e poi di provare le caselle non di angolo.

LA REALIZZAZIONE IN BASIC

La realizzazione in BASIC della strategia di gioco è immediata, leggendo direttamente dalla rappresentazione delle regole del gioco che abbiamo espresso nel paragrafo precedente. Riportiamo in figura la traduzione in BASIC della STRATEGIA di gioco. Per quel che riguarda la funzione POSSWIN, possiamo ricavare lo pseudocodice a partire dalla descrizione della funzione fatta nel precedente paragrafo:

INIZIO. POSSWIN

Esegui prodotto per righe, per colonne e per le due diagonali, mettendo i 6 risultati in SR(1) SR(2) SR(3) SC(1) SC(2) SC(3) SD1 e SD2 rispettivamente.

SE almeno un prodotto vale 27

ALLORA

X ha vinto. STOP

FINE__SE

Trova il massimo (MAX) tra i valori 18 e 50 presenti in SR(1), SC(1) SD1 ed SD2.

SE MAX = 0 ALLORA WINS = 0: RITORNA

ALTRIMENTI

Scegli quale casella va occupata

FINE__SE

FINE.

La realizzazione BASIC di POSSWIN è presentata in figura. Il lettore può notare come con un ciclo innestato unico (istruzioni 750-820), si riescono ad ottenere tutti i prodotti che servono. Un altro trucco utilizzato spesso nella POSSWIN è quello di "leggere" la scacchiera, definita come BOARD(9), come se fosse una struttura a due dimensioni. In quest'ultimo caso, le coordinate I di riga e J di colonna vengono trasformate tramite la formula $L = 3*(I-1) + J$, per recuperare l'indice L della struttura ad 1 sola dimensione. Le istruzioni 760 e 770, invece, illustrano come si ottengono le coordinate degli elementi sulla diagonale 1 (cioè le posizioni 1,5 e 9 della BOARD) e sulla diagonale 2 (cioè le posizioni 3, 5 e 7 sulla BOARD) a partire dall'indice I che varia da 1 a 3. Le istruzioni 1020 fino alla 1200, scandiscono la sequenza di caselle che ha il valore critico più alto (18 o 50), e trovano la casella vuota della sequenza.

ALTRE OSSERVAZIONI

Il resto del programma non è di difficile comprensione. Diamo solo qualche suggerimento per rendere l'interpretazione più agevole.

- La scelta di chi gioca per primo è ottenuta impostando la variabile TCCA ad 1 o -1. Poiché ad ogni giro questa variabile viene moltiplicata per -1, otteniamo che vari da +1 a -1 a +1 a -1 e così via, a seconda del valore con cui è stata impostata la prima volta. Così il Giocatore farà solo le mosse pari o dispari.
- La vittoria di X si determina dal fatto che il valore MAX è 27, la vittoria del Giocatore è individuabile dal fatto che MAX vale 50.

COME DIVERTIRSI CON IL GIOCATORE

Il modo migliore per sfruttare le capacità di "intelligenza" del programma proposto è di giocare con esso molte partite, seguendo i ragionamenti del programma, che vengono emessi ad ogni turno. Osservate che la strategia presentata è migliorabile, soprattutto per la funzione Make 2. Una delle cose che il Giocatore non sa fare è PREVEDERE LE FORCHE. Sapendo questo dovrete intravedere che esiste una strategia di avversario che batte il nostro Giocatore!

A voi il compito di scoprire quale. Sapreste migliorare la funzione POSSWIN in modo che anche questa possibilità sia presa in considerazione?

Questo è un ottimo stimolo alla creazione di STRATEGIE alternative, che vi porterà a comprendere i problemi di generalità della soluzione di problemi di A.I.

Arrivederci.

```
5 PRINT "Q"
10 DIM BOARD(9),OR(3),SEG$(5)
20 SEG$(2)="." : SEG$(5)="0" : SEG$(3)="X"
30 FOR I=1 TO 3:FOR J=1 TO 3:L=3*(I 1)+(J:BOARD(L)=2:NEXT J:NEXT I
40 REM *****
50 PRINT " COMINCI TU O IO ";:INPUT TT$
55 TCCA=-1
56 IF TT$="IO" THEN TCCA=1
60 REM *****
70 PRINT " "
80 FOR M=1 TO 9
90 TCCA=TCCA*(-1)
100 PRINT "          TURNO      ":M
110 IF TCCA<0 THEN GOSUB 200:REM GIOCA TU
```

```

120 IF TCCA>0 THEN GOSUB 200:REM GIOCO IO
130 GOSUB 1330:REM STAMPA SITUAZIONE
140 IF MAX=50 THEN PRINT " HO VINTO0000 !!!!!":END
150 PRINT " "
160 NEXT M
170 REM *****
180 PRINT " PATTA "
190 END
200 REM *****
210 REM ***** GIOCA TU *****
220 REM *****
230 INPUT " DOVE SCRIVI LA CROCE ";TUKEN
240 IF BOARD(TUKEN)<>2 THEN PRINT"POSTO GIA' OCCUPATO!":GOTO 230
250 IF BOARD(TUKEN)=2 THEN BOARD(TUKEN)=3
260 RETURN
270 REM *****
280 REM ***** GIOCO IO *****
290 REM *****
300 IF M<>1 THEN GOTO 330
310 PLACE=1:GOSUB 640
320 RETURN
330 IF M<>2 THEN GOTO 370
340 PLACE=1:IF BOARD(5)=2 THEN PLACE=5
350 GOSUB 640:REM GO
360 RETURN
370 IF M<>3 THEN GOTO 410
380 PLACE=3:IF BOARD(9)=2 THEN PLACE=9
390 GOSUB 640:REM GO
400 RETURN
410 IF M<>4 THEN GOTO 450
420 GOSUB 700:REM POSSWIN
430 IF WINS=0 THEN GOTO 434
431 PLACE=WINS
432 GOSUB 640
433 GOTO 440
434 GOSUB 1230:REM MAKE2
440 RETURN
450 IF M<>5 THEN GOTO 510
460 GOSUB 700:REM POSSWIN
470 IF WINS<>0 THEN PLACE=WINS
480 IF WINS<>0 GOTO 490
482 PLACE=3
485 IF BOARD(7)=2 THEN PLACE=7
490 GOSUB 640:REM GO
500 RETURN
510 IF M<>6 THEN GOTO 550
520 GOSUB 700:REM POSSWIN
530 IF WINS=0 THEN GOTO 534
531 PLACE=WINS
532 GOSUB 640
533 GOTO 540
534 GOSUB 1230:REM MAKE2
540 RETURN
550 REM QUI SE TURNO E' 7 0 0 0
560 GOSUB 700:REM POSSWIN
570 FOR I=1 TO 3
580 IF BOARD(I)=2 THEN PLACE=I
590 NEXT I

```

```

600 IF WING<>0 THEN PLACE=WING
610 GOSUB 640
620 RETURN
630 REM *****
640 REM ***** GO (PLACE) *****
650 REM *****
660 BOARD(PLACE)=5
670 PRINT "SCRIVO 0 IN ":PLACE
680 RETURN
690 REM *****
700 REM ***** POSSWIN *****
710 REM *****
720 FOR I=1 TO 3:SR(I)=1:SC(I)=1:NEXT I
730 REM *****
740 SD1=1:SD2=1
750 FOR I=1 TO 3
760 FOR J=1 TO 3
770 SR(I)=BOARD(3*(I-1)+J)*SR(I)
780 SC(J)=BOARD(3*(I-1)+J)*SC(J)
790 NEXT J
800 NEXT I
810 SD1=BOARD(1)*BOARD(5)*BOARD(9)
820 SD2=BOARD(3)*BOARD(5)*BOARD(7)
830 REM *****
840 FOR I=1 TO 3
850 IF SR(I)=27 OR SC(I)=27 THEN G=1
860 NEXT I
870 IF SD1=27 OR SD2=27 THEN G=1
880 IF G=1 THEN PRINT "GASSP HAI VINTO !!!! !":STOP
890 K=0:MAX=0:WING=0
900 FOR I=1 TO 3
910 IF SR(I)=18 AND MAX<20 THEN MAX=18:K=1:IX=I
920 IF SR(I)=50 THEN MAX=50:K=1:IX=I
930 IF SC(I)=18 AND MAX<20 THEN MAX=18:K=2:IX=I
940 IF SC(I)=50 THEN MAX=50:K=2:IX=I
950 NEXT I
960 IF SD1=18 AND MAX<20 THEN MAX=18:K=3
970 IF SD1=50 THEN MAX=50:K=3
980 IF SD2=18 AND MAX<20 THEN MAX=18:K=4
990 IF SD2=50 THEN MAX=50:K=4
1000 IF MAX=0 THEN WING=0:GOTO 1220
1010 REM QUI SE VI E' POSSIBILITA' DI VITTORIA ..
1020 IF K<>1 THEN GOTO 1070:REM RIGA
1030 FOR H=1 TO 3
1040 L=3*(IX-1)+H:IF BOARD(L)=2 THEN WING=L
1050 NEXT H
1060 RETURN
1070 IF K<>2 THEN GOTO 1120:REM COLONNA
1080 FOR H=1 TO 3
1090 L=3*(H-1)+IX:IF BOARD(L)=2 THEN WING=L
1100 NEXT H
1110 RETURN
1120 IF K<>3 THEN GOTO 1170:REM DIAGONALE 1
1130 FOR H=1 TO 3
1140 L=1+4*(H-1):IF BOARD(L)=2 THEN WING=L
1150 NEXT H
1160 RETURN

```

```

1170 IF K<>4 THEN GOTO 1220:REM DIAGONALE 2
1180 FOR H=1 TO 3
1190 L=3+2*(H-1):IF BOARD(L)=2 THEN WING=L
1200 NEXT H
1210 RETURN
1220 RETURN
1230 REM *****
1240 REM ***** MAKE 2 *****
1250 REM *****
1260 IF BOARD(5)<>2 THEN GOTO 1280
1270 BOARD(5)=5:RETURN
1280 FOR I=1 TO 4
1290 IF BOARD(2*I)=2 THEN BOARD(2*I)=5:GOTO 1310
1300 NEXT I
1310 RETURN
1320 REM DISEGNA NUOVA SITUAZIONE
1322 PRINT "Q"
1330 PRINT " "
1340 FOR I=1 TO 3:PRINT " ";
1350 FOR J=1 TO 3:PRINT SEG$(BOARD(3*(I-1)+J));" ";NEXT J:PRINT
1360 NEXT I:PRINT " "
1370 RETURN
READY.

```

VIC PENDOLO

Questo divertente programma, che trasforma il vostro VIC 20 in un vero orologio a pendolo, può essere utilizzato solamente tramite l'applicazione del Super-expander, cioè di un'espansione di memoria.

Per attivare VIC PENDOLO inserite l'ora e premete RETURN: inserite poi i minuti. Sarà quindi sufficiente che premiate RETURN perché il vostro VIC-20 disegni sullo schermo del vostro televisore un orologio che segnerà l'ora esatta finché non deciderete di interrompere l'esecuzione del programma.

Se inserite un'ora negativa, il programma partirà con l'ora corrente dell'orologio interno del VIC-20.

Il programma scandirà il passare del tempo con una musichetta che vi ricorderà senza dubbio qualcosa...

N.B.: per ottenere una visualizzazione soddisfacente dell'orologio è necessario lanciare una prima volta l'esecuzione del programma. Attendete che l'orologio venga visualizzato, poi interrompete il programma con RUN/STOP e RESTORE e quindi date nuovamente il RUN.

```

10 DEFFNS(X)=SIN(2*PI/180*X)
20 DEFFNC(X)=COS(2*PI*X/180)
30 PRINT"*****"
40 PRINT" *VIC-PENDOLO*"
50 PRINT" *****"
70 PRINT"INTRODUCETE L'ORA":INPUTA$
75 H=VAL(A$):IFH<10THENA$="0"+A$
77 IFH<0THEN110
80 PRINT"INTRODUCETE I MINUTI":PRINT"PREMERE RETURN A"
85 PRINT" L'ORA ESATTA":
90 INPUTMN$:IFLEN(MN$)=1THENMN$="0"+MN$
92 A$=RIGHT$(A$,2)
95 TI$=A$+MN$+"00"
100 H=-3.5
110 GRAPHIC2
120 COLOR6,1,1,4
130 REM DISEGNA OROLOGIO
140 FORI=1TO12
150 X=10+INT(8*FNC(I*15-90))
160 Y=8-INT(9*FNS(I*15-90))
170 A$=STR$(I):IFI<9ANDI>5THENA$=" "+A$
180 CHARX,Y,A$
190 NEXTI
200 CIRCLE2,520,520,470,450
205 PAINT1,1,1

```

```

210 MN=-3: X=520: Y=X: XH=X: YH=X: XS=X: YS=X: S=-3
300 IFH=INT(H)ANDF=0THENPRINT" ■T603ECD02T7GT6G03DEC"
305 A#=MID$(TI$,3,2)
310 A=VAL(A#)
320 IFA=MNTHEN360
330 MN=A
335 DRAW,520,520TOX,Y
340 X=INT(300*FNS(A*3))+520
350 Y=-INT(300*FNC(A*3))+520
360 DRAW3,520,520TOX,Y
400 A#=LEFT$(TI$,3)
410 A=VAL(A#)/10
415 GOSUB1000
420 F=1: IFH=ATHEN460
430 H=A: F=0
435 DRAW0,520,520TOXH,YH
440 XH=INT(180*FNS(A*15))+520
450 YH=-INT(180*FNC(A*15))+520
460 DRAW3,520,520TOXH,YH
500 A=VAL(RIGHT$(TI$,2))
510 IFS=ATHEN300
520 S=A
530 DRAW0,520,520TOXS,YS
540 XS=INT(330*FNS(A*3))+520
560 YS=-INT(330*FNC(A*3))+520
570 DRAW2,520,520TOXS,YS
580 GOTO300
1000 B=A-INT(A): A=A-B
1010 B=B*10/6: A=A+B
1020 RETURN

```

STRUTTURA DEL PROGRAMMA:

Le linee 10 e 20 definiscono delle funzioni che calcolano il seno e coseno dell'argomento X.

Le linee da 30 a 50 eseguono la presentazione iniziale.

La linea 70 vi permette di inserire l'ora.

La linea 75 mette l'ora da voi inserita precedentemente, sotto forma di un numero di due cifre.

La linea 77 fa partire l'orologio con il valore attuale di TI\$ se avete inserito un valore negativo.

Le linee da 80 a 92 vi permettono di inserire i minuti.

La linea 95 regola l'orologio interno del VIC-20.

La linea 110 passa lo schermo nel modo in alta risoluzione.

La linea 120 determina i colori che saranno utilizzati.

Le linee da 140 a 190 costituiscono un ciclo che, ripetuto dodici volte, permette di determinare la posizione di ogni ora sul quadrante del pendolo (linee 150-160) e stampa nella giusta posizio-

ne questa ora (linee 170-180).

La linea 200 disegna il quadrante.

La linea 205 colora la superficie esterna dell'orologio in modo da avere un quadrante blu su fondo bianco.

La linea 300 genera la musica del pendolo all'inizio dell'esecuzione e poi a tutte le ore.

Le linee da 305 a 320 determinano se è necessario spostare la lancetta delle ore.

La linea 335 cancella la lancetta delle ore.

Le linee da 340 a 350 determinano la nuova posizione della lancetta delle ore.

La linea 360 disegna la lancetta delle ore nella nuova posizione.

Le linee da 400 a 415 determinano il numero di minuti dopo la conversione sessagesimale-decimale (sottoprogramma 1000).

La linea 420 rimanda al movimento della lancetta dei secondi se la lancetta dei minuti non deve muoversi.

La linea 435 cancella la lancetta dei minuti.

Le linee 440 e 450 calcolano la nuova posizione della lancetta dei minuti.

La linea 460 disegna la lancetta dei minuti nella nuova posizione.

Le linee da 500 a 520 determinano il numero di secondi.

La linea 510 rimanda al movimento della lancetta delle ore se la lancetta dei secondi non deve spostarsi.

La linea 530 cancella la lancetta dei secondi.

Le linee 540 e 560 calcolano la nuova posizione della lancetta dei secondi.

La linea 570 disegna la lancetta dei secondi nella nuova posizione.

La linea 580 rimanda al movimento della lancetta delle ore.

Il sottoprogramma da 1000 a 1020 permette di convertire un numero dato sotto forma sessagesimale (MM'.SS'') in notazione decimale (es: 15'30'' 15.5).

CARATTERI PARTICOLARI:

linea 30: SHIFT CLR, CRSR↓, CTRL 3, cinque spazi, * (undici volte).

linea 40: cinque spazi, *, *

linea 50: cinque spazi, * (undici volte)

linea 70: CRSR↓ (tre volte), CTRL 7

linea 80: CRSR↓ (due volte), CRSR↓, CTRL 9, CTRL 0

linea 85: CRSR↓

linea 300: CTRL—

STATUS REGISTER E ISTRUZIONI ARITMETICHE

STATUS REGISTER E ISTRUZIONI ARITMETICHE

In questa lezione impareremo tutto ciò che riguarda il funzionamento dello STATUS REGISTER e dei FLAGS, per essere in grado di affrontare le istruzioni aritmetiche della CPU e per conoscere le basi necessarie per un apprendimento più completo di nuove istruzioni.

4.1 STATUS REGISTER

Prima di affrontare la spiegazione delle nuove istruzioni della CPU 6510 è necessario capire il funzionamento di un particolare registro, già nominato nell'illustrare l'architettura interna del microprocessore, lo "STATUS REGISTER" o "REGISTRO DI STATO". Esso è costituito da un insieme di 8 bits, ciascuno con un suo preciso significato. Il valore di tali bits (1 oppure 0) è determinato dal risultato di precedenti operazioni della CPU e tali valori possono più o meno influire sull'elaborazione di un programma in un modo piuttosto che in un altro. Quando viene eseguita una istruzione in l/m, viene memorizzato il tipo di risultato ottenuto dall'elaborazione di quest'ultima, settando (portando a 1) o resettando alcuni bits che hanno un preciso significato nello STATUS REGISTER. Se per esempio volessimo controllare se la differenza tra due numeri è zero, cioè se tali numeri sono uguali, controlleremmo se un certo bit nel suddetto registro sia stato settato o no e ci comporteremmo di conseguenza. Esistono infatti istruzioni che permettono di cambiare indirizzo di elaborazione (equivalente al "GOTO" Basic) a seconda del valore dei FLAGS (i bits dello S.R. sono anche chiamati così). Ovvio è quindi l'importanza dello S.R. in un programma ed il significato dei singoli bits che ci accingiamo ad illustrare di seguito esaminandoli dal bit meno significativo a quello più significativo.

4.1.1 Bit 0, CARRY

Questo bit viene settato da operazioni aritmetiche quando sussiste un risultato non rappresentabile in un numero di 8 bits. Se

per esempio l'Accumulatore contenesse il numero 211 a cui venisse sommato il numero 60, non potendo la somma 271 essere rappresentata da un numero ad 8 bits (il massimo è infatti 255), in A verrebbe memorizzato il numero $271-256 = 15$ ed il Flag di CARRY settato ad 1. Questo bit è quindi un "RIPORTO" (traduzione di Carry). Vale 0 se non vi è alcun riporto nell'ultima operazione eseguita, 1 per il contrario.

4.1.2 Bit 1, ZERO

Questo bit è quasi certamente il più usato in un programma. Esso, dopo un'operazione, viene settato ad 1 se il risultato è uguale a zero, resettato a 0 per il contrario. Se per esempio sottraessimo due numeri, il Flag ZERO sarebbe uguale ad 1 nel caso che i due numeri fossero uguali, resettato a 0 nel caso contrario.

4.1.3 Bit 2, IRQ DISABLE

Come abbiamo accennato nella prima puntata di questo corso, un microprocessore può essere interrotto durante l'elaborazione di un programma da un dispositivo periferico che gli richiede un qualunque lavoro. Il bit IRQ (Interrupt Request Disable) è proprio quello che controlla che la CPU possa rispondere o no ad una richiesta esterna di Interrupt. Questo Flag settato ad 1 fa sì che la CPU ignori qualsiasi eventuale richiesta di interruzione, mentre è ovvio che portato a 0 permetterà ad essa di rispondere a qualsiasi interruzione. L'utilità di questo flag è quindi di non permettere al microprocessore di essere "distolto" dall'elaborazione della parte delicata di un programma o permette di aumentare la velocità globale di esecuzione.

4.1.4 Bit 3, DECIMAL MODE

Come dovrete sapere, tutti i calcoli vengono eseguiti nei registri e nelle memorie in modo binario tra cifre rappresentabili in base esadecimale. Settando questo bit ad 1, la CPU invece farà calcoli aritmetici in base decimale (BCD), per cui il massimo numero rappresentabile dall'Accumulatore sarà 99. Il Flag settato a 0 farà eseguire i calcoli nel solito modo binario.

4.1.5 Bit 4, BRK COMMAND

Esiste un certo tipo di istruzione detta "Interruzione Forzata" (BRK) che simula una richiesta esterna di Interrupt. Dopo l'esecuzione di tale comando il bit BRK COMMAND viene settato ad 1.

4.1.6 Bit 5, 1

Questo bit non è utilizzato e vale sempre 1.

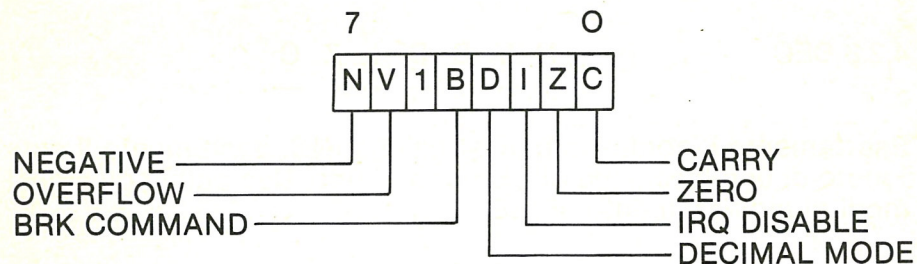
4.1.7 Bit 6, OVERFLOW

Questo Flag viene settato ad 1 se il risultato di un'operazione aritmetica è maggiore di 127 o minore di -127. Se invece il risultato è compreso tra questi due valori il Flag è resettato a 0.

4.1.8 Bit 7, NEGATIVE

Questo bit è lo "specchio" del bit 7 di 1 byte risultante da un'operazione aritmetica. Se tale bit è ad 1 il Flag viene settato ad 1, viceversa se il bit è a 0.

Possiamo quindi concludere questa parte dedicata allo STATUS REGISTER dandovi lo schema di quest'ultimo:



4.2 ISTRUZIONI ARITMETICHE

Introdotta lo STATUS REGISTER possiamo ora affrontare altre istruzioni in I/m, dando di seguito lo stato dei flags ad elaborazione avvenuta. Useremo per tale scopo i seguenti segni:

- per dire stato del Flag non alterato

- 1 per dire stato settato ad 1
- 0 per dire stato settato a 0
- ↑ per dire stato dipendente dal risultato dell'operazione

Per esempio:

```
N V B D I Z C
0 0 - - - ↑ 0
```

vuol dire che l'operazione resetta i Flags N, V, C, lascia inalterati B, D, I e se il risultato è zero setta Z ad 1 altrimenti a 0.

4.2.1 INC

```
N V B F I Z C
↑ - - - - ↑ -
```

Questa istruzione incrementa di uno il contenuto di una memoria. I possibili indirizzamenti sono:

ABSOLUTE
ZERO PAGE
Z. PAGE,X
ABB.,X

4.2.2 INX INY

```
N V B D I Z C
↑ - - - - ↑ -
```

Incrementa di uno il registro Y (INX) o X (INY).

4.2.3 DEC

```
N V B D I Z C
↑ - - - - ↑ -
```

Esattamente il contrario dell'istruzione INC, decrementa il contenuto di una locazione di memoria di uno. Usa quindi gli stessi modi di indirizzamento di INC.

4.2.4 DEX DEY

```
N V B D I Z C
↑ - - - - ↑ -
```

Il contrario di INX ed INY, decrementano i registri.

4.2.5 CMP

```
N V B D I Z C
↑ - - - - ↑ ↑
```

Esegue una comparazione tra memoria ed Accumulatore. Essa setta i Flags come se venisse eseguita una sottrazione tra i due numeri, sottrazione "immaginaria" in quanto i contenuti dell'Accumulatore e della memoria rimangono inalterati. I modi di indirizzamento sono:

IMMEDIATE
 ABSOLUTE
 ZERO PAGE
 (IND),X
 (IND),Y
 Z.PAGE,X
 ABSOLUTE,X
 ABSOLUTE,Y

4.2.6 CPX CPY

N V B D I Z C
 ↑ — — — — ↑ ↑

Uguale a CMP, ma il contenuto della memoria è comparato con i registri X o Y. I modi di indirizzamento sono:

IMMEDIATE
 ABSOLUTE
 Z.PAGE

4.2.7 ADC

N V B D I Z C
 ↑ ↑ — — — ↑ ↑

Somma il contenuto di una memoria al contenuto di A con riporto. Ciò vuol dire che se C è settato ad 1 (CARRY) in A avremo la somma dei due numeri scelti più uno. I modi di indirizzamento sono:

IMMEDIATE
 ABSOLUTE
 Z.PAGE
 (IND),X
 (IND),Y
 Z.PAGE,X
 ABS.,X
 ABS.,Y

4.2.8 SBC

N V B D I Z C
↑ ↑ — — — ↑ *

Sottrae il contenuto di una memoria al contenuto dell'Accumulatore con "prestito". Si ottiene cioè esattamente il contrario di ADC. I modi di indirizzamento sono:

IMMEDIATE
ABSOLUTE
ZERO PAGE
(IND,X)
(IND),Y
Z.PAGE,X
ABS.,X
ABS.,Y

* Il Carry rappresenta l'eventuale prestito a sottrazione avvenuta.

Siamo così giunti alla fine di questa quarta puntata sul l/m avendo introdotto i concetti di Flags e STATUS REGISTER. Abbiamo inoltre esaminato le istruzioni aritmetiche di cui dispone la CPU 6510. Chiudiamo salutandovi e sperando ancora una volta di essere stati chiari e di avervi interessati con nuovi argomenti.

A.V.S.

PRENOTATE IN EDICOLA

COMPUTER SET

N.2

per VIC 20, C.16, Plus/4

**Lo straordinario mensile che offre un elegante
manuale di 100 pagine e 3 cassette**

**(una di utility importanti e supergiochi,
due speciali per salvare il vostro software).**

!!!

**I FERRI DEL MESTIERE
PER IL PROGRAMMATTORE
A SOLE 10.000 LIRE**

- (3) TELEFONO (ric.)
- (4) INSERIMENTO
- (5) VARIAZIONE
- (6) CANCELLAZIONE
- (7) LETTURA
- (8) REGISTRAZIONE

Le opzioni sono facili da usare e non richiedono quindi spiegazioni particolareggiate: ricordiamo solo che i nuovi dati possono essere inseriti dopo aver premuto il tasto numero 4, e che potranno essere corretti o cancellati dopo aver richiesto le apposite opzioni. La ricerca dei dati avviene fornendo indizi quali l'indirizzo, il nome, ecc.

Con le opzioni numero 7 (lettura) e 8 (registrazione) è possibile caricare i dati da nastro o salvarli.

VIC VIC...BANG!

Questo gioco è una versione incruenta della famosa "roulette russa". Il VIC e il suo avversario spareranno a turno un colpo ciascuno, ma solo una pallottola su dieci non è a salve. Il colpo mortale andrà simbolicamente a colpire uno scrigno colmo di dobloni e una moneta d'oro passerà nel forziere avversario.

Dopo dieci tentativi, o quando uno dei concorrenti avrà perso tutti i suoi dobloni, il gioco avrà termine.

Seguendo le istruzioni comparirà sullo schermo la richiesta del VIC di conoscere l'identità del suo avversario: scrivi allora il tuo nome, premi RETURN e preparati psicologicamente alla sfida. Sul video apparirà lo schema del gioco: quando sei pronto per iniziare, premi nuovamente RETURN.

Sia tu che il VIC 20 avete a disposizione 100\$ a testa: sullo schermo in alto apparirà la cifra relativa alle giocate.

Il risultato finale comparirà al termine della sfida: in bocca al lupo!

LE TRE CARTE

Questo è un gioco che ti permetterà di misurarti non soltanto con la tua fortuna, ma anche col tuo colpo d'occhio.

Ti verrà chiesto di dichiarare il tuo capitale; effettuata la scelta della cifra che sei disposto a rischiare, compariranno tre carte sul video mischiate da una veloce mano invisibile;

QUALE SCEGLI? (1, 2, 3).

Dopo aver scelto la carta sulla quale tentare la fortuna dichiarerai l'entità della tua puntata.

Se indovinerai la carta vincente guadagnerai entrambe le puntate effettuate sulle altre due carte, quindi, il doppio della tua posta.

A questo punto concentrati, aguzza la vista e...buona fortuna!

IPPODROMO

Inserendo questo divertentissimo gioco, immaginiamo di stare seguendo attentamente, attraverso il nostro binocolo, il cavallo su cui abbiamo scommesso tutti i nostri risparmi e che potrà farci vincere cifre favolose o ridurci letteralmente sul lastrico!

La prima cosa che dovremo decidere sarà il numero dei giocatori, da 1 a 6: inseriamo allora il nostro nome e quello dei nostri amici appassionati anche loro di corse di cavalli.

Dopo che saranno apparsi i nomi dei cavalli in gara, scegliamo il nostro galoppatore schiacciando il tasto col numero corrispondente e subito dopo il RETURN.

Apparirà la scritta: PUNTATA; decidete la cifra che volete rischiare e attenzione! La corsa comincia, i cavalli galoppo alzando un polverone di terra coi loro zoccoli.

Cercate di mantenere la calma e il sangue freddo: gli avversari possono sempre battervi con uno scatto finale!

Alla fine della corsa, sul tabellone compariranno le quote con cui sarà pagato il cavallo vincente (per esempio 1/9).

Siete pronti? Allora attenzione: la corsa ha inizio!

(LATO B: C. 16 - PLUS/4)

PAROL

Inserito il programma potete scegliere 10 lettere dell'alfabeto tra vocali -V- e consonanti -C-.

Avete a disposizione un minuto per formare una parola, utilizzando naturalmente il maggior numero possibile di lettere; tenete sottomano un vocabolario per non cadere nella facile tentazione di scrivere parole inesistenti.

Potete giocare con un amico o in due gruppi più numerosi. Aguzzate l'ingegno e scrivete alla fine del tempo consentito il numero delle lettere che formano il vocabolo elaborato e poi il vocabolo stesso, (se sbagliate a batterlo premete CLR e ricominciate da capo).

Il vostro computer a questo punto vi chiederà "onestamente" se la parola inserita è lecita: nel caso non abbiate trovato parole esistenti premete R per ricominciare.

In caso elaboraste parole di uguale lunghezza a quelle del vostro avversario, il vincitore risulterà il giocatore che ha distribuito le lettere.

Ricordatevi di definire all'inizio del gioco il numero di manche.

Non preoccupatevi se vi sembrerà che il computer non risponda alla vostra richiesta delle lettere, il programma richiede qualche secondo per l'elaborazione dei dati.

Ricordatevi che di parole ne esistono davvero tantissime quindi, spremetevi le meningi e...buon divertimento!

FILETTO

Certamente non sarà facile battere un computer super allenato come il vostro C16, ma non lasciatevi scoraggiare tanto facilmente perché non è detto che la famosa dea cieca non sorrida proprio a voi!

Inserito il programma, dovrete per prima cosa rispondere alla domanda su chi farà la prima mossa. Se deciderete che debba essere proprio il computer a cominciare, subito apparirà sul video la mossa effettuata.

Per vincere dovrete riuscire ad occupare con le vostre pedine tre caselle in sequenza verticale, orizzontale o diagonale prima del vostro avversario, cercando inoltre di prevenire le sue mosse.

Per occupare una casella scrivete il numero sul video.

Per giocare più partite, premete RETURN e poi S.

Attenti al computer e buona fortuna!

VOCABOLARIO

VOCABOLARIO è un programma nuovo ed estremamente interessante per tutti noi che viviamo in una società come quella odierna in cui imparare le lingue straniere è diventato di fondamentale importanza in tutti i campi della vita, dai viaggi alle relazioni sociali fino a quello lavorativo.

Il programma funziona come un vero e proprio vocabolario, ma è ancora più utile in quanto permette, con la velocità di ricerca consentita, un notevole risparmio di tempo.

Avrete la possibilità di scegliere la lingua straniera che vi interessa:

- 1) Italiano-Inglese
- 2) Italiano-Francese
- 3) Italiano-Tedesco

o, con il numero 4), qualsiasi altra traduzione desideriate.

Dopo aver selezionato il numero tra 1 e 4, potrete inserire i vocaboli con la relativa traduzione.

Es.: Il vostro vocabolario è Italiano-Inglese; battete il n. 1.

Alla richiesta: "Vocabolo", scrivete la parola italiana e premete il RETURN. Il computer vi chiederà se desiderate inserire in memoria il vocabolo: se sì, vi chiederà la traduzione inglese.

Potrete così costituire la lista di parole che vi interessa stampare nel vostro vocabolario.

Se alla domanda "Vocabolo?" battete il tasto 1, otterrete l'inversione della traduzione, nel nostro caso Inglese-Italiano. (Per tornare all'Italiano-Inglese battete nuovamente 1).

Se dopo la domanda "Vocabolo?" battete invece il numero 2, potrete correggere gli eventuali errori commessi nell'inserimento delle parole: basterà allora riscrivere i vocaboli corretti.

Per registrare il vostro VOCABOLARIO su nastro, premete il tasto 3 sempre dopo la domanda "Vocabolo?", e seguite le istruzioni che vi verranno fornite dal computer stesso.

Con il numero 4 potrete ottenere la lista dei vocaboli inseriti usando i tasti di controllo del cursore: CRSR UP e CRSR DOWN.

ISTOGRAMMI 3D

Questo programma serve per la rappresentazione grafica di gruppi di dati o informazioni legati da particolari relazioni e di cui sia utile ottenere le percentuali relative: avrete così l'immagine visiva immediata dei vostri dati, divisi in 6 gruppi, sotto forma di istogrammi tridimensionali.

Caricate il programma e inserite i nomi o le sigle che indicano i vostri dati e successivamente anche le quantità di cui vi interessa ottenere le percentuali relative: sul video appariranno gli istogrammi tridimensionali. Avrete anche la possibilità di conoscere con maggiore precisione le percentuali richieste, rispondendo in maniera affermativa alle domande poste dal computer.

CHITARRA FACILE

Il programma che vi proponiamo mostra un metodo estremamente semplice, ma sicuramente pratico ed efficace per imparare a suonare la chitarra da soli in breve tempo. Chiaramente non vi promettiamo di fare di voi degli esperti musicisti, ma ci proponiamo di insegnarvi le basi per poterlo diventare; sta a voi, dopo, studiare a lungo e allenarvi seriamente per mettere in pratica ciò che avete imparato.

Ma sicuramente con CHITARRA FACILE sarete rapidamente in grado di cimentarvi in un accompagnamento musicale di una voce o di un altro strumento.

Caricato il programma vi verrà mostrato l'elenco degli accordi disponibili fra cui scegliere quello desiderato: alla domanda "accordo?" scrivete la vostra richiesta. Comparirà allora l'immagine della tastiera della chitarra in cui i pallini azzurri indicano la corda e il tasto da premere e i numeri posti di fianco, il dito da utilizzare:

1 = indice, 2 = medio, 3 = anulare, 4 = mignolo.

L'asterisco indica che la corda corrispondente non deve essere suonata.

Numeri già apparsi:

PLAY ON TAPE N. 1

per i possessori di VIC 20

Nel manuale n. 1: I tasti funzionali del VIC 20 - Pseudocodice e programmazione Basic - Il joystick.

Nella cassetta n. 1: Totocalcio - Air attack - Cervellone - Inferno 3D - Bilancio familiare.

PLAY ON TAPE N. 2

per i possessori di VIC 20

Nel manuale n. 2: Il basic più veloce - I caratteri speciali del VIC 20 - Disegnare con la tastiera e il joystick - Pseudocodice: 2ª lezione.

Nella cassetta n. 2: Test per misurare il Quoziente intellettuale - Easyword - Caccia al tesoro - Gestione Magazzino - Formula 1.

PLAY ON TAPE N. 3

per i possessori di VIC 20

Nel manuale n. 3: I cicli annidati del Basic - Come sviluppare un programma (Corsa automobilistica - Piranha) - L'interfaccia sconosciuta - Conosci il tuo VIC 20?

Nella cassetta n. 3: Sette e mezzo - Dieta - Calorie dei cibi - Gangster - Costi chilometrici.

PLAY ON TAPE N. 4

per i possessori di VIC 20

Nel manuale n. 4: Pseudocodice: Istruzioni read e data - Figura richiama - Grafici a barra. Il basic del VIC 20 - I linguaggi di programmazione - Come personalizzare i programmi.

Nella cassetta n. 4: G.O MO-KU - Calcolatrice - Golf - Vic Tab - Cabala e sogni.

PLAY ON TAPE N. 5

per i possessori di VIC 20

Nel manuale n. 5: Pseudocodice: 5ª lezione - I sistemi di numerazione - Le periferiche di ingresso e uscita - BASIC: I cicli.

Nella cassetta n. 5: Atterraggio - Memory Trainer - Stop Music - Program Recorder: 1) Crealista, 2) Stampalista - Istocambio.

PLAY ON TAPE N. 6

per i possessori di VIC 20, C.16, PLUS/4

Nel manuale n. 6: LINGUAGGIO MACCHINA: 1ª lezione - Pseudocodice: I numeri a caso - La smazzata - Basic: 3ª lezione.

Nella cassetta n. 6:

Lato Vic 20: Redknight - Il risparmiatore - Api e ragni - Quindici 3D - Oroscopo. Lato C.16 - Plus/4: Totocalcio - Colour Search - Roulette - Wargame - Bioritmo.

PLAY ON TAPE N. 7

per i possessori di VIC 20, C.16, PLUS/4

Nel manuale n. 7: LINGUAGGIO MACCHINA: 2ª lezione - Pseudocodice: Liste legate - Il bestiario - Il basic e le periferiche.

Nella cassetta n. 7:

Lato Vic 20: Anagrammi - Ponticello - Puzzle - Sonar Cave - Julianus dies. Lato C.16 - Plus/4: Hat in the ring - Dieta - Calorie - Poker - Quindici 3D.

PLAY ON TAPE N. 8

per i possessori di VIC 20, C.16, PLUS/4

Nel manuale n. 8: Pseudocodice: I files - Come tradurre un numero decimale alla base 2 o 16. LINGUAGGIO MACCHINA: Le istruzioni di caricamento - Gli algoritmi fondamentali.

Nella cassetta n. 8:

Lato Vic 20: Meteore - The last battle - Seven eleven - Contatore telefonico - Riflessometro. Lato C.16 - Plus/4: Riflessometro - Quadrati - C.16 Memory - Contatore telefonico - Dama.

I numeri arretrati costano L. 15.000. Indirizzare vaglia o assegno a Editoriale VIDEO via Castelvetro 9 - 20154 Milano specificando il numero richiesto. Ufficio tecnico e arretrati: telefono 02/3184829

PLAY ON TAPE
COMPUTER

9

VIC 20:

- IPPODROMO
- VIC VIC... BANG
- LE TRE CARTE
- POWER
- CONVERSIONI

C. 16 - PLUS/4:

- VOCABOLARIO
- ISTOGRAMMI 3D
- CHITARRA FACILE
- FILETTO
- PAROL

EV EDITORIALE VIDEO