

Dopo la parentesi del numero scorso, dedicata al software del PC-1500, torniamo questo mese ad interessarci agli elaborati per il PC-1211 che i lettori continuano ad inviarci. Vorremmo, anzi, approfittare di questo spazio per congratularci delle numerose lettere che puntualmente la rubrica riceve; purtroppo il nostro peggior nemico è lo spazio a disposizione, il quale ci costringe a ritardare la pubblicazione di alcuni programmi, a causa della catasta (tanto per citare un termine in tema) di lavori ricevuti. A proposito di cataste, code, liste e alberi: perché qualcuno non tenta di adattare una di queste strutture di dati al mini-basic del PC-1211, o a quello più esteso del PC-1500?

Ma veniamo ai lavori pubblicati questo mese. Il primo non è un vero e proprio programma, ma una serie di "remarks" di Marino Coretti (il maestro di Trieste che i nostri lettori conoscono) sui salti condizionati del PC-1211. Il secondo invece, di Massimo Mangia, è un adattamento al PC-1211 di un gioco di simulazione automobilistica: il Driver.

Note sui salti condizionati

di Marino Coretti (Trieste)

Leggendo gli articoli di Giustozzi sulla programmazione strutturata, mi sono posto il problema di come utilizzare al meglio il dialetto Basic del PC-1211, allo scopo di evitare il più possibile la stesura di programmi spaghetati. Gli esempi che seguiranno sono volutamente banali, ma sono convinto che, dal punto di vista didattico, non esiste nulla di peggio del presentare esemplificazioni difficili da comprendere quando si deve trattare di un argomento poco conosciuto.

I sacri testi dicono che SE (IF) la condizione è VERA, ALLORA (THEN) si SALTA alla linea n di programma, altrimenti si prosegue con l'istruzione successiva. Più delle parole valgono gli esempi. In figura 1 è riportato il diagramma di flusso per risolvere il seguente problemino: - dati due numeri z e w (entrambi diversi da zero), trovare il loro M.C.D.

Chi ha appena iniziato a programmare, in genere traduce il flow-chart alla lettera, ottenendo un programma di difficile lettura perché spaghetato, simile a quello proposto in figura 2. Fortunatamente il basic della 1211 offre invece qualche possibilità di scrivere un programma più strutturato.

Prima di affrontare il problema, sarà però necessario scoprire un po' meglio co-

me si effettuano i salti condizionati con la SHARP. Provate a far girare questo semplice programma:

```
10 INPUT "NUMERO A?"; A
20 INPUT "NUMERO B?"; B
30 IF A > B PRINT "I NUMERI SONO DIVERSI"; PRINT A; "MAGGIORE DI"; B; END
40 IF A < B PRINT "I NUMERI NON SONO UGUALI"; PRINT A; "MINORE DI"; B; END
50 IF A = B PRINT "I NUMERI SONO UGUALI"; PRINT "CIAO"; END
```

Potete notare come le "inflexioni dialettali" della SHARP siano piuttosto interessanti: se la condizione posta dopo IF è vera, oltre all'istruzione PRINT vengono eseguite anche tutte le altre istruzioni poste sulla stessa linea di programma; in caso contrario il programma stesso esegue la linea successiva. La nostra programmabile permette, in definitiva, di realizzare una struttura del tipo IF..THEN..ELSE.

Torniamo ora al problema iniziale. A pensarci bene, il test di controllo più importante è proprio l'ultimo ($Z=W?$). Ho pensato allora di scriverlo per primo e di sfruttare le caratteristiche della SHARP esposte poc'anzi. Ho ricavato così il programma in figura 3.

Tutto sommato, non ho fatto altro che seguire il consiglio di Hasenmajer (vedi MC n° 5 pag. 73): ho infatti "pascalizzato" il mio ragionamento, traducendo delle

strutture iterative tipiche del Pascal, in Basic.

Esistono però altri sistemi per semplificare la procedura dei salti condizionati. Uno dei più usati è quello che impiega gli operatori logici AND e OR.

Come tantissime persone, anch'io ho voluto cimentarmi con un programma che mi permettesse di giocare a Master Mind. Chi ha realizzato questo programma potrà dirvi che la difficoltà maggiore consiste nel trovare il modo più semplice per confrontare le cifre introdotte dal giocatore, con quelle generate dal calcolatore, al fine di stabilire quante cifre (indipendentemente dalla posizione occupata) sono state trovate.

Un sistema potrebbe essere quello di usare un contatore e di sfruttare la possibilità di scrivere LET dopo il test come in figura 4. Ma impiegando l'algebra di Boole, il problema può essere risolto con quattro linee di programma (fig. 5). A, B, C, D sono le cifre memorizzate dal computer; E, F, G, H quelle introdotte invece nel calcolatore.

Un ultimo esempio ancora, per chiarire come funziona la sintassi dell'AND.

Come noto, la Sharp rifiuta di eseguire divisioni del tipo 0/0 oppure n/0. È opportuno perciò introdurre, nei programmi in cui il risultato della divisione sia importante, dei test di controllo, per far capire subito all'operatore ciò che sta accadendo.

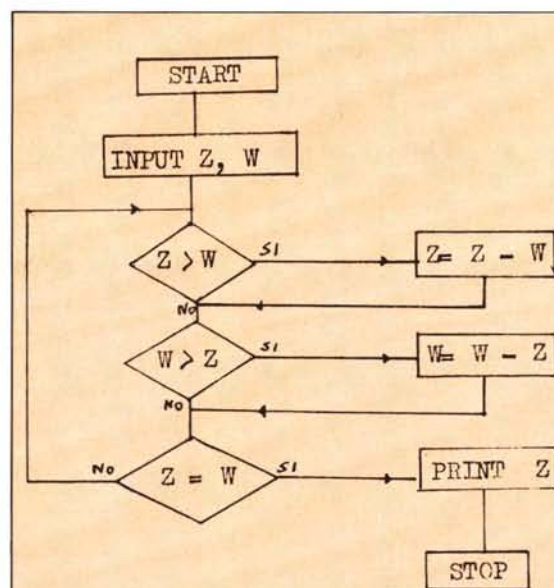


Figura 1

```
10: INPUT Z, W
20: IF Z > W THEN 1
   00
30: IF W > Z THEN 8
   0
40: IF Z = W THEN 6
   0
50: GOTO 20
60: PRINT Z
70: END
80: W = W - Z
90: GOTO 40
100: Z = Z - W
110: GOTO 30
```

Figura 2

```
10: INPUT Z, W
20: IF Z = W PRINT
   Z: END
30: IF Z > W LET Z =
   Z - W
40: IF W > Z LET W =
   W - Z
50: GOTO 20
```

Figura 3

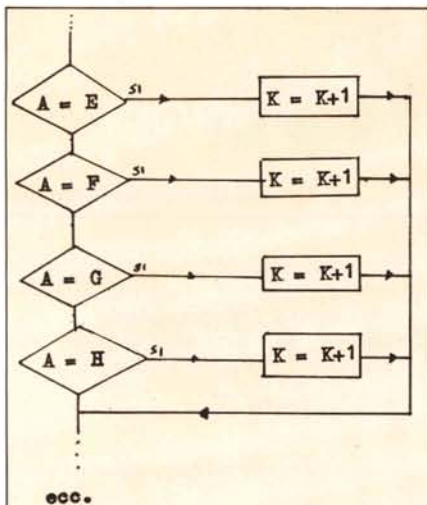


Figura 4

```

10: INPUT A, B, C, D
20: INPUT E, F, G, H
30: K=0
40: IF (A=E)+(A=F)+(A=G)+(A=H) <> 0 LET K=K+1
50: IF (B=E)+(B=F)+(B=G)+(B=H) <> 0 LET K=K+1
60: IF (C=E)+(C=F)+(C=G)+(C=H) <> 0 LET K=K+1
70: IF (D=E)+(D=F)+(D=G)+(D=H) <> 0 LET K=K+1
80: PRINT K: END
100: INPUT A, B
110: IF (A <> 0) * (B <> 0) LET D=A/B
120: IF (A=0) * (B <> 0) LET D=0
130: PRINT "DATI INACCETTABILI": PRINT "DIVISIONE NON ESEGUITA": END
    
```

Figura 5

Figura 6

Il programma è quello in figura 6. Le linee 110-140 si leggono così: se A è diverso da 0 e B è diverso da 0, esegui la divisione e fai apparire il risultato; altrimenti se A è uguale a 0 e B è diverso da 0, stampa 0; altrimenti fai comparire il messaggio.

F.1 - Driver

di Massimo Murgia
(Castellammare di Stabia)

Il programma simula una corsa automobilistica, ma a differenza di molti altri programmi simili, presenta alcune difficoltà fondamentali, quali: programmabilità del circuito, accurata simulazione del rendimento della macchina, ed una visualizzazione dinamica dello svolgimento del gioco.

Dopo aver dato il RUN al programma, sul display apparirà la richiesta "SEZ. PISTA?": bisognerà quindi inserire il numero di sezioni che comporranno il circuito (il massimo numero ammesso è 48). Dopodiché si immetteranno, una per una, tali sezioni adoperando i cinque simboli seguenti:

- I rettilineo
- (curva larga a destra
-) curva larga a sinistra
- > curva stretta a sinistra

< curva stretta a destra

I successivi input richiesti saranno il numero di giri di pista, ed i dati relativi alla guida, cioè: marcia (da 1 a 5), acc/freno (da 1 a 9 per accelerare, da -1 a -9 per frenare, 0 non comporta mutamenti di velocità), e lo sterzo (da 1 a 9 per girare a destra, da -1 a -9 a sinistra).

Dopo aver inserito l'ultimo dato, la "vettura" partirà ed il display mostrerà a sinistra la pista entro la quale l'auto sarà rappresentata dal simbolo Z, e sulla destra i 4 successivi tratti di pista. Tutto questo sarà visualizzato tante volte quante sezioni di pista si riuscirà a percorrere in base alla velocità raggiunta.

Infine apparirà il numero del giro, il tempo impiegato, il numero di giri del motore e la velocità in km/h. Premendo ENTER verrà visualizzato l'ultimo tratto di pista percorso, e premendo di nuovo il ta-

sto, saranno richiesti i dati relativi alla guida. In caso di uscita fuori pista saranno emessi due bip, e sarà sufficiente premere SHIFT B per rientrare.

SHIFT C invece permetterà di rivedere il numero di giri del motore e la velocità.

Il massimo regime di giri è 12000 g/m oltre il quale apparirà la scritta FUORI GIRI. È prevista inoltre una variabile per determinare eventuali rotture del motore, ed una che tiene conto del surriscaldamento freni, avvisando in caso di fading.

Cambi o scalate di marcia fanno rispettivamente scendere o salire il numero di giri del motore, come effettivamente avviene in realtà, e la sterzata è inversamente proporzionale alla velocità, quindi attenti a tenere conto. Con SHIFT A si ricomincia il gioco, senza dover impostare le sezioni di pista, usando quindi lo stesso circuito.

```

20: A$="  " : B$="  " : C$="  " : D$="  " : E$="  " : F$="  " : G$="  " : H$="  " : I$="  " : J$="  " : K$="  " : L$="  " : M$="  " : N$="  " : O$="  " : P$="  " : Q$="  " : R$="  " : S$="  " : T$="  " : U$="  " : V$="  " : W$="  " : X$="  " : Y$="  " : Z$="  " :
30: INPUT "SEZ. PISTA? " I
40: FOR W=27 TO N
50: INPUT "SEZ. " A$(W): NEXT W
60: FOR W=N+27 TO N+30
62: A$(W)="*": NEXT W
64: "A" INPUT " GIRI? " K
66: T=0: P=0: Q=22: L=27: Y=1
68: J=0: R=0: S=3: O=1
70: INPUT "MARCIA " M
72: INPUT "ACC/FRENO " F
75: INPUT "STERZO " S
80: IF G <> 0 LET Q=Q+G: IF Q < 5 BEEP 2: PAUSE " FADING": IF Q < 0 LET G=-Q: G=0
90: J=J+G+(Q-M)*5: IF J < 0 LET J=1
100: IF J > 20 BEEP 2: PAUSE " FUORI GIRI": P=P+J-20: J=20
110: IF P > 3 BEEP 3: PRINT " MOTORE ROTTO. " : END
120: G=LN J/1.4*M: O=M
130: FOR W=1 TO INT G
140: L=L+.5: M=INT L: U=M+(L-M)*2: V=U+1: X=U+2: Z=U+3
150: IF M < N+26 LET L=27: M=L: Y=Y+1
160: IF A$(M)="(" LET R=-1: GOT0 200
170: IF A$(M)=")" LET R=1: GOT0 200
180: IF A$(M)="(" LET R=-2: GOT0 200
190: IF A$(M)=")" LET R=2
200: R=R+H/(3*G+.1): S=S+R: I=INT S
210: IF I < 1 BEEP 2: PRINT " " : A$(M);
220: IF I > 6 BEEP 2: PRINT " " : A$(M);
225: PAUSE " " : A$(M); A$(I); A$(M);
230: NEXT W
240: T=T+1: PAUSE " TEMPO " : T
245: IF Y=K+1 BEEP 3: M=INT (N*K/4+E2): PRINT " TEMPO TOT. " : T: MEDIA: M: END
250: "O" H=J*600: W=INT (G*28)
260: PRINT "G/M " : H: VEL. " : W: " KM/H "
265: PRINT " " : A$(M); A$(I); A$(M); A$(U); A$(V); A$(X); A$(Z)
270: Q=Q+3: IF Q > 2 LET Q=22
280: GOT0 70
290: "B" T=T+5: GOT0 68
    
```

Figura 7 - Listing programma F1 - Driver

Programma F.1 - Driver

20:	Inizializzazione stringhe	150:	posizione e indicatori
da 30 a 50:	Caricamento in memoria del circuito	160 a 190:	Controllo fine pista
da 60 a 62:	Caricamento in memoria ultime 4 sezioni	200:	Calcolo coefficiente di pista
64:	Input giri	210 a 220:	Calcolo variabili posizione laterale
da 66 a 68:	Inizializzazione variabili	225:	Controllo fuori pista
da 70 a 75:	Input dati guida	240:	Visualizzazione pista
80:	Controllo surriscaldamento freni	245:	Visualizzazione giri e tempo
90:	Calcolo giri motore	250:	Controllo giri terminati
da 100 a 110:	Controllo fuori giri e rottura motore	260:	Calcolo giri motore e velocità
120:	Calcolo velocità (sez. da percorrere)	270:	Visualizzazione giri motore e velocità
da 130 a 230:	Visualizzazione pista	290:	Incremento variabile surriscaldamento freni
140:	Incremento variabili		Rientro in pista

Più del flow chart valgono questi brevi commenti