

Questa rubrica occupa, questo mese, più spazio del consueto. La presenza dei due diagrammi di flusso molto chiari e completi, tuttavia, facilita la possibilità di trasferire i programmi su altre macchine, anche grazie al fatto che in essi si fa uso di istruzioni Basic abbastanza standard. Invitiamo quindi anche chi possiede macchine diverse dalla PC-1211 ad adattare i programmi al proprio computer: divertimento a parte, è un esercizio utilissimo per migliorare le proprie capacità.

Torniamo questo mese ad occuparci della PC-1211, dopo la parentesi dedicata alla sorella maggiore PC-1500, per pubblicare due programmi che, sebbene si occupino di soggetti completamente diversi, sono accomunati dallo stesso tema di base: la didattica e l'apprendimento.

Il programma "Scacco 3" infatti, insegna all'operatore un finale di Torre e Cavallo nel gioco degli scacchi. Il computer dispone di un solo Re, l'operatore del Re, Torre e Cavallo. Ci si può pertanto esercitare nel concludere la partita con il minor numero di mosse possibili.

Il secondo lavoro pubblicato, invece, permette al computer di imparare un certo elenco di animali in base alle indicazioni fornite dall'operatore. "Indovino l'animale" risponde, fra l'altro, alla nostra richiesta effettuata in uno dei numeri scorsi, riguardante l'impiego di particolari strutture di dati con le tascabili Sharp. Il programma di Valerio Oss impiega infatti, per la gestione dell'elenco dei dati, una struttura ad albero binario simmetrico con due puntatori.

## SCACCO 3

di Lorenzo Spoladore (Rovigo)

SCACCO 3 riproduce un finale di Torre e Cavallo.

Il programma manovra uno dei due Re, mentre il suo avversario muove l'altro Re con la Torre e il Cavallo. Appare evidente, quindi, che il computer può solo pareggiare (nella migliore delle ipotesi) o perdere, anche se quest'ultima eventualità è meno facile da conseguire di quanto ci si possa aspettare.

Tutte le regole del gioco sono rispettate dal programma e devono essere seguite anche dall'altro giocatore. Ad esempio, quando il Re del computer è sottoposto a scacco è necessario dichiararlo, con una procedura che chiarirò in seguito.

L'algoritmo utilizzato in SCACCO 3 è del tipo esaustivo, cioè valuta tutte le otto mosse possibili, assegnando a ciascuna un

"voto", in base alle seguenti regole:

1) Il Re deve mantenersi quanto più possibile lontano dai lati della scacchiera (cioè perché le insidie maggiori per il Re del computer si nascondono presso i bordi della scacchiera).

2) Il Re deve attaccare (nell'ordine) Torre e Cavallo, oppure la Torre, oppure il Cavallo a patto, però, di non avvicinarsi troppo ai limiti della scacchiera.

La regola 2) implica che il programma può derogare, entro certi limiti, alla stretta osservanza della 1), nel caso in cui la mossa gli permetta di attaccare (e, verosimilmente,

di mettere in difficoltà) l'avversario. Questa flessibilità nel seguire le due regole, conferendo al programma un certo grado di imprevedibilità, riproduce una situazione più simile ad una normale partita a scacchi.

Utilizzare SCACCO 3 è facilissimo. All'inizio si richiama il programma con il comando SHIFT D (è necessario, quindi, operare nel modo DEF). Si devono stabilire le posizioni iniziali di tutti e quattro i pezzi. Il simbolo R+R indica la riga (si veda la scacchiera di fig. 3) del pezzo R+ (Re del computer), mentre R+C indica la

```

40: "D" X=0: INPUT "R+R", A, "R+C", F
20: "A" A(36)=0: INPUT "T*R", Q, "T*C", R
30: INPUT "C*R", U, "C*C", V
40: INPUT "R*R", S, "R*C", T
50: B=F+1: C=A: D=F-1: E=A+1: G=A-1: H=F: I=A+1: J=F+1
60: K=A-1: L=F-1: H=A+1: U=F-1: O=A-1: P=F+1
70: A(35)=1: Y=0: FOR Z=1 TO 15 STEP 2: Y=Y+1: W=0: A(26+Y)=0
80: IF (A(Z)>8)+(A(Z)<1)+(A(Z+1)>8)+(A(Z+1)<1) THEN 190
90: IF ABS(S-A(Z))<2 IF ABS(T-A(Z+1))<2 THEN 190
100: IF ABS((A(Z)-U)*(A(Z+1)-V))=2 THEN 190
110: IF A(Z)=Q IF A(Z+1)=R LET W=400: GOTO 170
120: IF Q=A(Z) THEN 300
130: IF R=A(Z+1) THEN 350
140: IF ABS(Q-A(Z))<2 IF ABS(R-A(Z+1))<2 LET W=9.5
150: IF A(Z)=U IF A(Z+1)=V LET W=W+200: GOTO 170
160: IF ABS(U-A(Z))<2 IF ABS(V-A(Z+1))<2 LET W=W+6
170: A(26+Y)=(A(Z)-1)*(8-A(Z))+(A(Z+1)-1)*(8-A(Z+1))+W+1
180: A(35)=0
190: NEXT Z
200: IF A(35) IF A(36) BEEP 3: PRINT "MATTO! _ _ N. MOSSE=_ _"; X: END
210: IF A(35) BEEP 2: PRINT "*PATTA*_ _ _ N. MOSSE=_ _"; X: END
220: X=X+1
230: Y=27
240: FOR Z=28 TO 34
250: IF A(Z)>A(Y) LET Y=Z
260: NEXT Z
270: A=A(2*Y-53): F=A(2*Y-52)
280: BEEP 1: PRINT "N. MOSSE=", X: PRINT "RIGA=", A; " _ COLUMNA="; F
290: GOTO 20
300: IF S=Q IF T<R IF T>A(Z+1) THEN 130
310: IF S=Q IF T>R IF T<A(Z+1) THEN 130
320: IF U=Q IF V<R IF V>A(Z+1) THEN 130
330: IF U=Q IF V>R IF V<A(Z+1) THEN 130
340: GOTO 190
350: IF T=R IF S<Q IF S>A(Z) THEN 140
360: IF T=R IF S>Q IF S<A(Z) THEN 140
370: IF V=R IF U<Q IF U>A(Z) THEN 140
380: IF V=R IF U>Q IF U<A(Z) THEN 140
390: GOTO 190
400: "S" A(36)=1: BEEP 1: PAUSE " _ _ _ _ _ SCACCO_AL_RE!": GOTO 50

```

Figura 1 - Listing programma "Scacco 3"

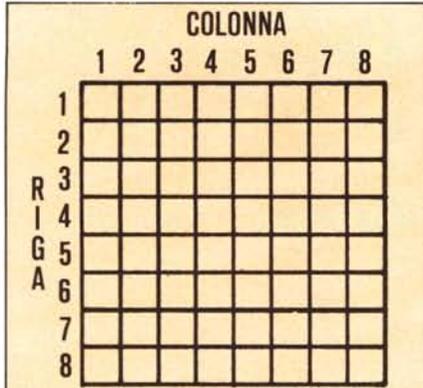


Figura 3

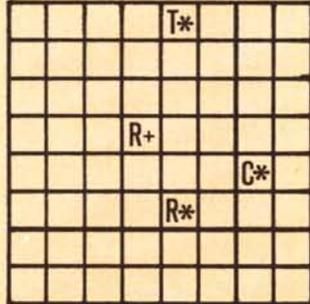


Figura 4

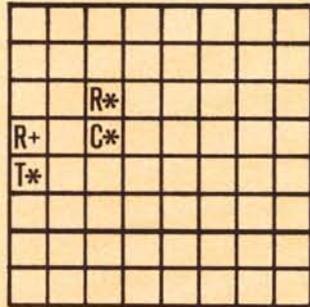


Figura 5

colonna di R+; T\*R indica la riga del pezzo T\* (Torre) etc.

Nella posizione iniziale R+ non può essere sotto scacco (in quanto non è possibile dichiararlo).

A questo punto la prima mossa spetta al computer, che, dopo alcuni secondi (max ca.35 sec.), mostra il numero di mosse da lui effettuate fino a quel momento (compresa quella attuale) e, quindi, mostra la sua decisione. Dopo aver premuto ENTER compare T\*R. Si deve, allora, muovere un pezzo. Se il pezzo mostrato non è quello che si vuole spostare, si preme ENTER e comparirà un altro pezzo. Nel caso in cui, durante questa operazione si commetta un errore, si preme SHIFT A e comparirà nuovamente T\*. Se, poi, il pezzo mosso pone sotto scacco R+, allora si preme SHIFT S e il computer "prenderà atto" dell'avvenuto scacco. Ma, se la mossa non provoca nessuna azione offensiva nei confronti di R+, premere ENTER finché il display è vuoto. Il numero di mosse mostrato quando è verificata la situazione di matto o di patta si riferisce alle mosse effettive di R+.

Se R+ mangia uno dei pezzi dell'avversario (!), condizione che si verifica quando R+ viene mosso dal programma in una casella occupata da uno dei pezzi, allora, al nuovo input bisogna rispondere, oltre che con la nuova mossa prescelta tra i pezzi rimasti, con -20 alla richiesta della riga e della colonna del pezzo mangiato. Successivamente quando ricompare la richiesta di input per questo pezzo, premere ENTER.

**Contenuto delle memorie**

A = RIGA R+

B ÷ P (esc. F) = CASELLE POSSIBILI PER R+

F = COLONNA R+

Q = RIGA T\*

R = COLONNA T\*

S = RIGA R\*

T = COLONNA R\*

U = RIGA C\*

V = COLONNA C\*

W = variabile ausiliaria

X = NUMERO MOSSE

Y = var. aus.

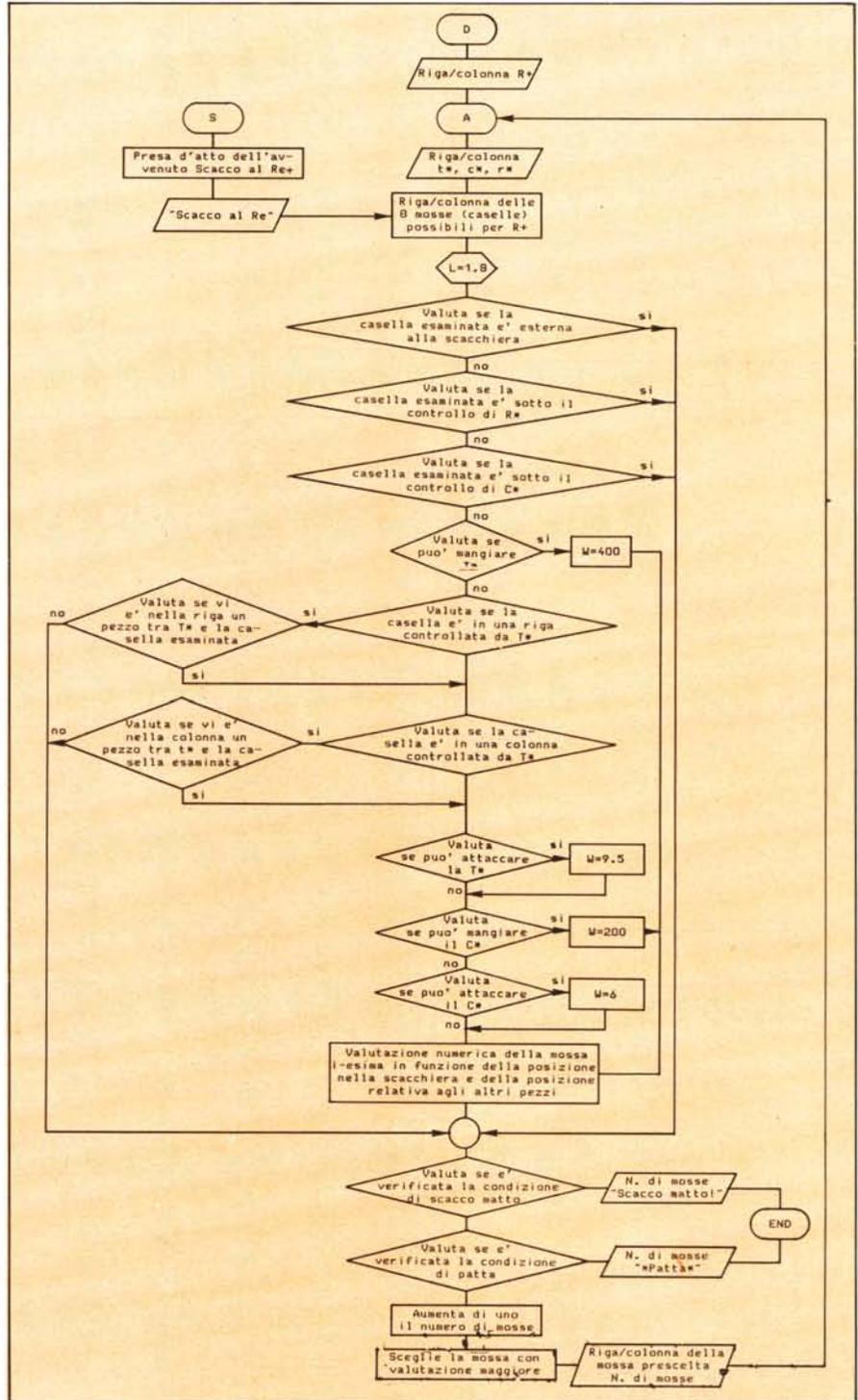


Figura 2 - Flow-chart programma "Scacco 3"

Z = var. aus.  
 A(27) ÷ A(34) = VOTI  
 A(35) = var. aus.  
 A(36) = var. aus.

### Nota

Vorrei, ora, attirare l'attenzione sulla variabile W. Come risulta dallo schema di flusso, essa assolve ad una importante funzione, cioè quella di assumere determinati valori, in rapporto alla posizione relativa di R+ rispetto agli altri pezzi. I valori sono più o meno rilevanti in base ai vantaggi che offre la posizione. Ad es., il valore è alto se R+ è messo nella condizione di poter mangiare T\*.

Trascrivo, ora, una partita che ho giocato con SCACCO 3. Riporto, per le prime mosse, anche i comandi per il corretto funzionamento del programma, ad esemplificazione di quanto ho precedentemente scritto.

La posizione iniziale della partita è quella illustrata in figura 4; la posizione finale è, invece, quella di figura 5 (vedi pagina 79).

DISPLAY	COMANDI
	SHIFT D
R+R	4 ENTER
R+C	4 ENTER
T*R	1 ENTER
T*C	5 ENTER
C*R	5 ENTER
C*C	7 ENTER
R*R	6 ENTER
R*C	5 ENTER
N. MOSSE = 1.	ENTER
RIGA = 4. COLONNA = 3.	ENTER
T*R	1 ENTER
T*C	4 ENTER
C*R	ENTER
R*R	ENTER
N. MOSSE = 2.	ENTER
RIGA = 5. COLONNA = 3.	ENTER
T*R	ENTER
C*R	4 ENTER
C*C	5 ENTER
R*R	SHIFT S
SCACCO AL RE!	
N. MOSSE = 3.	ENTER
RIGA = 4. COLONNA = 3.	ENTER
...	...
...	...

NUMERO MOSSE	+	- *
1°	R43	T14
2°	R53	C45+ (scacco)
3°	R43	T54
4°	R42	T53
5°	R32	R54
6°	R42	C24
7°	R41	R43
8°	R31	R33
9°	R41	T55 se C43 o C 12 patta
10°	R31	C43+
11°	R41	T51+

matto in 11 mosse

Come si è visto, ho dato scacco matto al computer (o, meglio, al programma SCACCO 3) in 11 mosse. Invito i lettori ad ottenere un risultato migliore (in particolare, meno di 10 mosse) partendo dalla stessa posizione iniziale ed utilizzando, naturalmente, SCACCO 3.

## INDOVINO L'ANIMALE

di Valerio Oss (Trento)

In questo programma il protagonista è certamente il calcolatore: esso infatti cerca di indovinare un animale da voi pensato dipendentemente dalle risposte che voi date alle domande che vi porrà. In pratica, il calcolatore segue un albero decisionale binario, passando al livello inferiore a seconda delle risposte date (SI/NO) fino a giungere ad una risposta.

Bisogna precisare che, siccome le variabili a stringa della PC-1211 contengono solo sette lettere, ogni domanda o risposta del calcolatore sarà necessariamente abbreviata e del tipo: "MAMMIF.?", "UCCELLO.?", "CARNIV.?".

Qualora la risposta data non fosse l'animale da voi pensato, ecco che la PC-1211 vi porrà una serie di domande per imparare e quindi memorizzare un nuovo animale e una nuova domanda.

Vediamo un esempio: immaginiamo che voi abbiate pensato il TOPO: ecco, nel listato pubblicato qui a fianco, come si svol-

```
PENSA UN ANIMALE E IO CERCO DI
INDOVINARLO
```

```
MAMMIF. ?
S
CARNIV. ?
N
ACQUATI. ?
N
CORNA. ?
N
CAVALLO. ?
N
QUALE ANIMALE
AVEVI PENSATO?
TOPO
DAMMI UNA DOMANDA
CHE DISTINGUA
UN CAVALLO
DA UN TOPO
ZOCCOLI?
E PER UN/UNA
TOPO LA RI-
SPONSA SAREBBE
(S/N)?
N
```

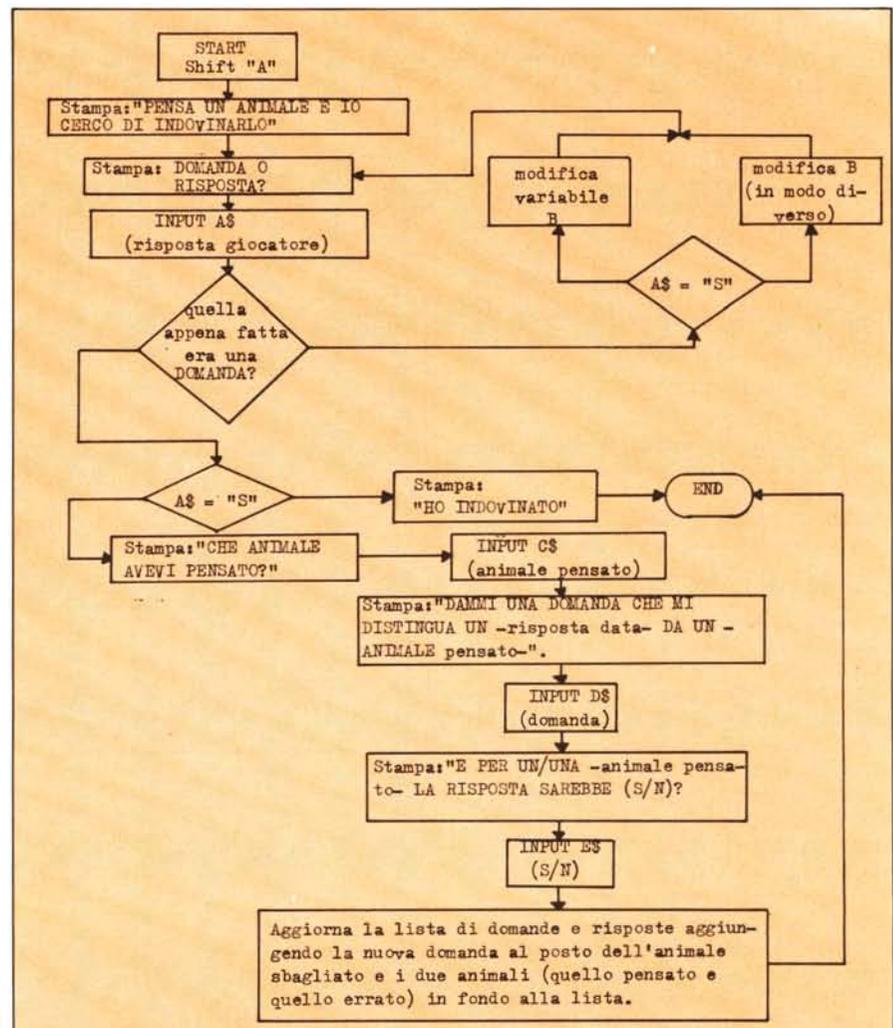


Figura 6 - Flow-chart del programma "Indovino l'animale"

gono le domande della PC-1211 dopo aver premuto shift "A" per partire.

La PC-1211 memorizza il nuovo animale (TOPO) e la risposta che lo distingue (ZOCCOLI?) dal cavallo. Chiede poi se per un topo la risposta alla domanda "ZOCCOLI?" sarebbe "S" o "N": ovviamente essa è NO. Se ora riprovate, pensando ancora al topo, ecco cosa accadrà:

```
PENSA UN ANIMALE E IO CERCO DI INDOVINARLO
```

```
MAMMIF.?
S
CARNIV.?
N
ACQUATI.?
N
CORNA.?
N
ZOCCOLI.?
N
TOPO.?
S
HO INDOVINATO!
```

Il calcolatore ha infatti corretto l'errore commesso precedentemente e non lo ripete più.

Tutte le domande e le risposte risiedono nelle memorie dalla 10a in poi. Introducendo inizialmente nella PC-1211 la lista di animali e domande proposte da Valerio Oss (fig. 8) si aggiungeranno via via animali ogni volta che il calcolatore sbaglia. C'è un limite ovviamente, data la capacità di memoria limitata della PC-1211, comunque si arriva lo stesso ad un buon numero di animali e domande (circa 66).

Ecco un brevissimo programma per l'introduzione degli animali e delle domande in memoria:

```
999: FOR A = 10 TO 140 STEP 2: B = A + 1: INPUT A$(A), A(B): NEXT A
```

Dopo aver premuto RUN 999 comparirà il "?" e si dovranno introdurre in questo ordine gli elementi della figura 8: inserire l'animale o la domanda; premere ENTER, inserire il numero che segue la domanda (0 nel caso dell'animale) e premere ancora ENTER. Ricordarsi di cancellare la linea 999 prima di lanciare il programma principale.

Si consiglia di registrare su nastro sia la memoria programma che quella dati, tramite le istruzioni CSAVE e PRINT#. Importante: prima di far girare il programma per la prima volta, occorrerà porre nella variabile G il numero 108: essa verrà aggiornata ogni volta che introdurrete nuovi

elementi.

Nel caso in cui qualche domanda non risultasse chiara a causa dell'abbreviazione, ecco la "traduzione" per le più incomprensibili:

AC. DOLC = vive in acqua dolce?

MANGIAB = è mangiabile?

CARNIV = è carnivoro?

VI. ASIA = vive in Asia?

ARACNID = è un aracnide?

BEI COL = ha bei colori?

GIOCHER = è giocherellone?

```
10: "A"PRINT "PENSA UN ANIMALE E IO CERCO DI INDOVINARLO":
PRINT " ": B = 10: GOTO 40
20: IF A$ = "S" LET B = INT (A(B+1)): GOTO 40
25: B = (A(B+1) - INT (A(B+1))) * 1000
40: PRINT A$(B): ".?"
45: INPUT A$: PRINT A$: IF A(B+1) <> 0 THEN 20
110: IF A$ = "S" PRINT "HO INDOVINATO!": END
130: PRINT "QUALE ANIMALE A VEDI PENSATO?": INPUT C$: PRINT C$
140: PRINT "DAMMI UNA DOMANDA CHE DISTINGUA UN ": A$(C): PRINT "DA UN ": C$: INPUT D$: PRINT D$: "?"
145: PRINT "E PER UN/UNA": PRINT C$: " L A RI-": PRINT "SPOSTA SARE BBE (S/N)?": INPUT E$: PRINT E$
150: A$(G) = A$(B): A$(B) = D$: A(G+1) = 0: G = G + 2: A$(G) = C$: A(G+1) = 0
160: IF E$ = "S" LET A(B+1) = G + (G - 2) / 1000: GOTO 170
165: A(B+1) = G / 1000 + G - 2
170: G = G + 2: END
```

Figura 7 - Listing del programma "Indovino l'animale"

```
MAMMIF 14.012
UCCELLO 16.018
CARNIV 34.032
MIGRA 36.038
PESCE 22.02
INSETTO 24.026
AC. DOLC 28.03
VOLA 42.04
RETTILE 44.046
TROTA 0.
MANGIAB 62.06
ACQUATI 66.05
VI. ASIA 54.052
RONDINE 0.
PICCOLO 58.056
FORMICA 0.
FA MIEL 72.074
ZAMPE 94.092
ARACNID 78.076
MUCCA 0.
CORNA 48.064
DOMEST 102.1
TIGRE 0.
BEI COL 82.08
COLIBRI 0.
SQUALO 0.
TONNO 0.
CAVALLO 0.
GIOCHER 70.068
BALENA 0.
DELFINO 0.
APE 0.
PUNGE 86.084
ANFIBIO 88.09
RAGNO 0.
RAPACE 98.096
PAYONE 0.
MOSCA 0.
ZANZARA 0.
RANA 0.
VIRUS 0.
VIPERA 0.
LUCERT 0.
GALLINA 0.
AQUILA 0.
LEONE 0.
FELINO 106.104
CANE 0.
GATTO 0.
0.
```

Figura 8 - Elenco domande e risposte da inserire nelle memorie per il programma "Indovino l'animale"