

Programmazione I

Compitino del 28 ottobre 1999

Svolgere l'esercizio numero 1 e, a scelta, alcuni tra gli altri esercizi.

1. Utilizzando *esclusivamente* variabili di *tipo semplice*, scrivere un programma che legga da input una sequenza di numeri interi positivi e negativi e scriva in output:

- il valore del piú grande multiplo di 3 contenuto nella sequenza;
- la media dei numeri pari presenti nella sequenza.

Si supponga che l'inserimento di 0 indichi la fine della sequenza (0 non deve essere considerato come elemento della sequenza). Nel caso le quantità richieste non possano essere calcolate, il programma dovrà fornire un messaggio d'errore.

2. Siano x e y due variabili di tipo `integer`. Si esprimano in linguaggio Pascal, senza ricorrere all'operatore `NOT`, le seguenti condizioni:

- $y < x \leq y + 10$
- la negazione della condizione precedente.

3. Nei seguenti frammenti di programma vengono utilizzati dei cicli `FOR`. Si individuino i frammenti in cui il ciclo è utilizzato in maniera impropria, indicando il motivo. Si riscrivano invece i frammenti corretti sostituendo al posto del ciclo `FOR` un ciclo `WHILE` equivalente (le variabili `somma`, i , x , n sono di tipo `integer`):

- ```
somma := 0;
FOR i := 1 TO n DO
 BEGIN
 readln(x);
 IF x = 0 THEN i := n + 1
 ELSE somma := somma + x
 END;
writeln(somma)
```

- ```
somma := 0;
FOR i := 1 TO n DO
  BEGIN
    readln(x);
    IF x = 0 THEN n := n + i
                ELSE somma := somma + x
  END;
writeln(somma)
```

4. Si consideri il seguente frammento di programma, in cui i e x sono due variabili di tipo `integer`, mentre c è una variabile di tipo `char`:

```
FOR i := x DOWNTO -x DO
  FOR c := 'A' TO 'Z' DO
    writeln(i,c)
```

a. Nell'ipotesi che il valore contenuto inizialmente in x non sia negativo, indicare, in funzione del contenuto di x , quante volte verrà eseguita l'istruzione `writeln(i,c)`.

b. Riscrivere il frammento di codice sostituendo i due cicli `FOR` con cicli `WHILE`.

5. Si considerino i seguenti programmi. Per ciascuno di essi indicare se ci saranno errori in fase di compilazione (in tal caso indicare il motivo), se ci saranno errori in fase d'esecuzione (in tal caso indicare dei valori di ingresso che causino errore), se il programma può entrare in un ciclo infinito (in tal caso indicare dei valori d'ingresso per cui ciò avviene). Non si tenga conto degli errori d'esecuzione che si potrebbero avere durante la lettura dei dati.

- PROGRAM p1 (input, output);
VAR x: real;
BEGIN
 readln(x);
 WHILE x >= 0 DO
 BEGIN
 writeln(x);
 x := x DIV 2
 END
 END.
END.
- PROGRAM p2 (input, output);
VAR x, y: integer;
BEGIN
 REPEAT
 readln(x, y)
 UNTIL x <= y;
 WHILE x <> y DO
 BEGIN
 writeln(x,y);
 x := x + 1;
 y := y - 1
 END
 END.
END.
- PROGRAM p3 (input, output);
VAR a, z: char;
 b: boolean;
BEGIN
 REPEAT
 readln(a);
 b := (a < 'a');
 b := b OR (a > 'z')
 UNTIL NOT b;
 FOR z := 'q' DOWNT0 a DO
 writeln(z)
 END.
END.

6. Si considerino i seguenti frammenti di codice. Per ognuno di essi scrivere delle dichiarazioni di variabile, in modo che le istruzioni risultino corrette dal punto di vista della compatibilità dei tipi. Se ciò non fosse possibile spiegare il motivo.

- x := a = b;
a := b MOD (ord(x) + 100)
- z := chr(x DIV (ord(z) MOD x))
- z := succ(z + pred(y)) / round(y)

7. Riscrivere il seguente frammento di programma utilizzando esclusivamente sequenza, selezione e cicli REPEAT.

```
WHILE (a > b) OR (b <= 36) DO
BEGIN
  a := a - 2;
  b := b + 1
END
```