

8 - LA STRUTTURA ITERATIVA (= di "iterazione", cioè "ripetizione") FOR ... DO ...

La struttura

FOR i:= n1 TO n2 DO istruzione

oppure:

```
FOR i:= n1 TO n2 DO
    BEGIN
        istruzione 1;
        istruzione 2;
        istruzione 3;
        ...
    END;
```

ordina al computer di:

- ❑ **assegnare il valore n1 alla variabile i** (NOTA);
- ❑ **eseguire l'istruzione che segue il DO** (oppure, il blocco di istruzioni compreso fra il BEGIN che segue il DO e l' END successivo);
- ❑ **incrementare di una unità il valore di i;**
- ❑ **eseguire nuovamente l'istruzione** (o il blocco);
- ❑ **incrementare di un'altra unità i;**
- ❑ **eseguire nuovamente l'istruzione** (o il blocco);
- ❑ ...

Quando, a forza di incrementi di un'unità, la variabile i assume il valore n2, viene eseguita per l'ultima volta l'istruzione (o il blocco), poi si esce dal ciclo.

NOTA: ho usato la lettera i per fissare le idee.

Al posto di i si può mettere un qualunque identificatore di variabile *intera*.

La variabile del ciclo FOR viene detta "**variabile di controllo**", e dev'essere sempre di tipo *intero*.

ESEMPI di programmi con l'uso della FOR ... DO ...

program somma_dei_numeri_naturali_da_1_fino_a_n; uses crt;

var k, n, somma: longint;

(* in questo programma la variabile k funge da "CONTATORE"
e la variabile somma da "ACCUMULATORE" *)

begin

```
  clrscr;
  writeln ('Dammi un numero intero positivo n;');
  writeln ('ti scriverò la somma dei numeri interi da 1 fino ad n. ');
  readln (n);
  somma:= 0;
  for k:= 1 to n do somma:=somma+k;
  write ('La somma che ti avevo promesso vale ', somma);
  readln;
```

end.

program massimo; uses crt;

var a, n, k, max: longint;

(* questo programma riceve in input dei numeri e stabilisce quale è stato il più grande fra i numeri introdotti.

La variabile max svolge il ruolo di "MASSIMO PROVVISORIO";

il suo valore finale corrisponderà al massimo cercato *)

begin

```
  clrscr; writeln ('Quanti numeri vuoi introdurre?'); readln (n);
  writeln ('Dammi pure questi ', n, ' numeri: ti dirò qual è il massimo');
  max:=0;
  for k:= 1 to n do
    begin
      readln (a);
      if a>max then max:=a;
    end;
  writeln ('Massimo = ', max);
  readln;
```

end.

ESERCIZI**Esercizio 4)**

Scrivi un programma che, letto da tastiera un numero intero positivo n , fornisca in output, su tre colonne:

- i numeri interi da 1 fino a n ;**
- i rispettivi quadrati;**
- le rispettive radici quadrate (in notazione non esponenziale).**

Supponendo, per fissare le idee, $n = 4$, l'output desiderato è:

NUMERO	QUADRATO	RADICE QUADRATA
1	1	1.00000
2	4	1.41421
3	9	1.73205
4	16	2.00000

Esercizio 5)

Letto in input un intero positivo n , si vuole in output il valore della somma $1+1/2+1/3+1/4+ \dots +1/n$.

Esercizio 6)

Letti in ingresso un numero qualunque a ed un numero intero positivo n , si vuole in output l' n -esima potenza di a , ossia il numero a^n .

E' indispensabile, in questo programma, una variabile accumulatore, il cui valore FINALE sarà la potenza desiderata.

Esercizio 7)

Letti in ingresso n numeri (n è fornito in input dall'utente) in parte positivi e in parte negativi, che rappresentano gli esiti (in euro) di altrettante partite di poker,

- **si conta quanti sono quelli negativi e se ne fa la media aritmetica;**
- **si conta quanti sono quelli positivi e se ne fa la media aritmetica.**

L'output dovrà essere:

Hai perso ... volte, perdendo in media ... euro ogni volta;
e hai vinto ... volte, vincendo in media ... euro ogni volta.
In totale, hai (perso oppure vinto) ... euro.

*Non è evidentemente necessario utilizzare tante variabili diverse per ciascuno dei numeri in ingresso;
basterà*

- *una sola variabile per la lettura;*
- *una variabile accumulatore per la somma dei negativi e una variabile accumulatore per la somma dei positivi;*
- *una variabile contatore per i negativi e una variabile contatore per i positivi;*
- *oltre, naturalmente, alla variabile di controllo della struttura FOR ... DO ...*

Esercizio 8)

Scrivi un programma che mandi **in output tutti gli interi da 0 a 30, ciascuno colorato del colore di cui l'intero considerato rappresenta il codice** (vedi paragrafo 5f, "Testo colorato in output").