

Algoritmi in Pascal

per il corso di

Informatica Generale — Scienze Ambientali

PROF. GIANCARLO MAURI
Appunti scritti dal Dott. Alberto Leporati

Dipartimento di Informatica, Sistemistica e Comunicazione (DISCo)
Università degli Studi di Milano - Bicocca
Via Bicocca degli Arcimboldi 8, 20126 Milano

Test di primalità

Problema: Scrivere un programma che, letto da tastiera un numero intero positivo, scriva sullo schermo se il numero inserito è primo oppure no¹.

```
Program TestPrimalita (input, output);
Var n, i: integer;
    primo: boolean;
Begin
  write('Inserisci un numero intero positivo: ');
  readln(n);

  primo := true;
  i := 2;
  while (i < n) AND primo do
    if (n MOD i) = 0
      then primo := false
      else i := i + 1;

  if primo
    then writeln('Il numero inserito e'' primo')
    else writeln('Il numero inserito non e'' primo')
End.
```

¹Si ricorda che un numero intero (positivo) n è primo se e solo se è divisibile solo per 1 e per se stesso.

Somma dei primi n multipli di 3

Problema: Scrivere un programma che, letto da tastiera un numero intero positivo n , scriva sullo schermo la somma dei primi n multipli di 3.

```
Program SommaMultipli3 (input, output);
Var i, n, somma: integer;
Begin
  write('Inserisci un numero intero positivo: ');
  readln(n);

  somma := 0;
  i := 3;
  while i <= 3*n do
    begin
      somma := somma + i;
      i := i + 3
    end;

  writeln ('La somma dei primi ', n, ' multipli di 3 e''': ', somma)
End.
```

Si poteva anche utilizzare un ciclo for, come segue:

```
Program SommaMultipli3 (input, output);
Var i, n, somma: integer;
Begin
  write('Inserisci un numero intero positivo: ');
  readln(n);

  somma := 0;
  for i := 3 to n do
    somma := somma + 3*i;

  writeln ('La somma dei primi ', n, ' multipli di 3 e''': ', somma)
End.
```

Somme di elementi di posto pari e dispari

Problema: Scrivere un programma che, letto da tastiera un numero intero positivo n , legga successivamente da tastiera $2n$ numeri reali e scriva sullo schermo la somma degli elementi di posto pari e la somma degli elementi di posto dispari.

```
Program SommePariDispari (input, output);
```

```

Var n, i: integer;
    sommapari, sommadispari, x: real;
Begin
    write('Inserisci un numero intero positivo: ');
    readln(n);

    sommapari := 0;
    sommadispari := 0;

    writeln('Ora inserisci ', 2*n, ' numeri reali');
    for i := 1 to n do
        begin
            readln(x);
            sommadispari := sommadispari + x;
            readln(x);
            sommapari := sommapari + x
        end;

        writeln('La somma degli elementi di posto pari e'' : ', sommapari);
        writeln('La somma degli elementi di posto dispari e'' : ', sommadispari)
    End.

```

Somma dei cubi delle cifre di un numero intero

Problema: Scrivere un programma che trovi i numeri coincidenti con la somma dei cubi delle cifre che li compongono, compresi tra 1 e 500.

```

Program SommaDeiCubi (input, output);
Var n, c, d, u: integer;
Begin
    for i := 1 to 500 do
        begin
            u := n MOD 10;
            d := ((n MOD 100) - u) DIV 10;
            c := (n - d*10 - u) DIV 100;
            if n = c*c*c + d*d*d + u*u*u
                then writeln(n)
        end
    End.

```

Fattoriale Iterativo

Problema: Scrivere un programma che, letto da tastiera un numero intero positivo n ,

scriva sullo schermo il valore del *fattoriale* di n^2 .

L'algoritmo che mostriamo qui di seguito è denominato **FattorialeIterativo** poiché fa uso di un ciclo per il calcolo del risultato; in seguito, vedremo come un altro algoritmo — basato sulla ricorsione — consente di calcolare il fattoriale di un numero in maniera molto più intuitiva (anche se meno efficiente dal punto di vista di occupazione della memoria).

```
Program FattorialeIterativo (input, output);
Var n, i, fatt: integer;
Begin
  write('Inserisci un numero intero positivo: ');
  readln(n);

  fatt := 1;
  for i := 2 to n do
    fatt := fatt * i;

  writeln('Il fattoriale di ', n, 'e''': ', fatt)
End.
```

Fibonacci Iterativo

Problema: Scrivere un programma che, letto da tastiera un numero intero positivo n , scriva sullo schermo il valore dell' n -esimo numero nella successione di Fibonacci³.

Come per il fattoriale, l'algoritmo che mostriamo qui di seguito è denominato **FibonacciIterativo** poiché fa uso di un ciclo per il calcolo del risultato; in seguito, vedremo come un altro algoritmo — basato sulla ricorsione — consente di calcolare l' n -esimo numero della successione di Fibonacci in maniera molto più intuitiva (sebbene in questo caso, a differenza di quanto avviene con il calcolo del fattoriale, l'occupazione in memoria dell'algoritmo ricorsivo risulterà essere veramente eccessiva).

```
Program FibonacciIterativo (input, output);
Var n, i, fib1, fib2, fib3: integer;
Begin
```

²Si ricorda che il fattoriale di un numero intero positivo n , indicato con $n!$, è definito come il prodotto di tutti i numeri interi compresi tra 1 ed n (incluso). Ovvero:

$$n! = n \cdot (n - 1) \cdot (n - 2) \cdot \dots \cdot 3 \cdot 2 \cdot 1$$

³Si ricorda che la successione di Fibonacci è la successione:

$$1, 1, 2, 3, 5, 8, 13, 21, 34, \dots$$

nella quale ciascun numero (tranne i primi due, che sono uguali a 1 per definizione) è uguale alla somma dei due numeri precedenti.

```

write('Inserisci un numero intero positivo: ');
readln(n);

fib1 := 1;
fib2 := 1;
fib3 := 2;
if n <= 1
  then writeln(1)
  else begin
    for i := 2 to n do
      begin
        fib1 := fib2;
        fib2 := fib3;
        fib3 := fib1 + fib2
      end;
    writeln(fib2)
  end
End.

```

Dato che il risultato del calcolo è contenuto nella variabile **fib2** (e non nella variabile **fib3**, come sarebbe stato logico aspettarsi), che viene inizializzata a 1, il controllo svolto dall'**if** è inutile. Quindi, il programma di cui sopra è esattamente equivalente a quello dato qui di seguito:

```

Program FibonacciIterativo (input, output);
Var n, i, fib1, fib2, fib3: integer;
Begin
  write('Inserisci un numero intero positivo: ');
  readln(n);

  fib1 := 1;
  fib2 := 1;
  fib3 := 2;
  for i := 2 to n do
    begin
      fib1 := fib2;
      fib2 := fib3;
      fib3 := fib1 + fib2
    end;
  writeln(fib2)
End.

```

Se $n \leq 1$, semplicemente non viene eseguito il ciclo **for**, e viene stampato il valore 1 contenuto nella variabile **fib2**.

Elemento massimo di un vettore

Problema: Scrivere un programma che, letti $N = 100$ numeri reali dalla tastiera, li inserisca in un vettore e successivamente scriva sullo schermo il valore massimo tra quelli inseriti.

```
Program ValoreMassimo (input, output);
Const N = 100;
Var vettore: array[1..N] of real;
    i: integer;
    massimo: real;
Begin
  writeln('Inserisci ', N, ' numeri reali: ');
  for i := 1 to N do
    readln(vettore[i]);

    massimo := vettore[1];
    for i := 2 to N do
      if vettore[i] > massimo
        then massimo := vettore[i];

    writeln('L'elemento massimo inserito e'' : ', massimo)
End.
```

Somme parziali di una successione

Problema: Scrivere un programma che, letti N numeri reali dalla tastiera — i quali rappresentano i primi N elementi di una successione numerica $\{a_n\}_{n \in \mathbf{N}}$ — scriva sullo schermo le corrispondenti N somme parziali⁴.

```
Program SommeParziali (input, output);
Const N = 100;
Var A, SommeParziali: array[1..N] of real;
    i: integer;
```

⁴Si ricorda che, data una successione numerica $\{a_n\}_{n \in \mathbf{N}}$, la successione $\{s_n\}_{n \in \mathbf{N}}$ delle *somme parziali* di $\{a_n\}_{n \in \mathbf{N}}$ è definita nel seguente modo (supponendo che la successione $\{a_n\}_{n \in \mathbf{N}}$ sia definita per $n \geq 1$):

$$\begin{aligned} s_1 &= a_1 \\ s_2 &= a_1 + a_2 = s_1 + a_2 \\ s_3 &= a_1 + a_2 + a_3 = s_2 + a_3 \\ &\vdots \\ s_k &= a_1 + a_2 + \dots + a_k = s_{k-1} + a_k \end{aligned}$$

```

Begin
  writeln('Inserisci ', N, ' numeri reali: ');
  for i := 1 to N do
    readln(A[i]);

    SommeParziali[1] := A[1];
  for i := 2 to N do
    SommeParziali[i] := SommeParziali[i-1] + A[i];

  writeln('Ecco le somme parziali: ');
  for i := 1 to N do
    writeln(SommeParziali[i]);
End.

```

Trasferimento di una matrice in un vettore

Problema: Scrivere un programma che, letti $N \cdot M$ numeri reali dalla tastiera, li inserisca in una matrice di N righe e di M colonne. Successivamente, si trasferiscano i numeri letti dalla suddetta matrice a un vettore di $N \cdot M$ componenti, e si stampi il vettore risultante sullo schermo.

```

Program TrasferisciMatrice (input, output);
Const N = 10, M = 20;
Var A: array[1..N, 1..M] of real;
    B: array[1..N*M] of real;
    i, j, k: integer;
Begin
  writeln('Inserisci ', N*M, ' numeri reali: ');
  for i := 1 to N do
    for j := 1 to M do
      readln(A[i, j]);

  k := 1;
  for i := 1 to N do
    for j := 1 to M do
      begin
        B[k] := A[i, j];
        k := k + 1
      end;

  writeln('Vettore risultante: ');
  for k := 1 to N*M do
    writeln(B[k]);
End.

```