

PASCAL: LE PROCEDURE 2

TRATTO DA CAMAGNI-NIKOLASSY, CORSO DI INFORMATICA, VOL. 2, HOEPLI

Informatica



Le procedure con i parametri passati per valore

Procedure e parametri

- All'atto della chiamata della procedura è possibile “passare” alla procedura i valori su cui la procedura stessa dovrà operare
- Tale meccanismo è detto **passaggio dei parametri**
- Consiste nell'elencazione delle variabili di seguito al nome della procedura

```
procedure swap(num1:integer, num2:integer);
```

- L'elenco di queste variabili prende il nome di **lista dei parametri formali**

La lista dei parametri formali è costituita da tre elementi caratteristici:

- **numero dei parametri**: numero di variabili scambiate;
- **tipo dei parametri**: ogni variabile deve essere definita con la sua tipologia (integer, real, char ecc.);
- **ordine dei parametri**: la corrispondenza tra le variabili della procedura e quelle del programma chiamante avviene rispettando l'ordine con cui sono elencate nella lista.

Esempio: swap

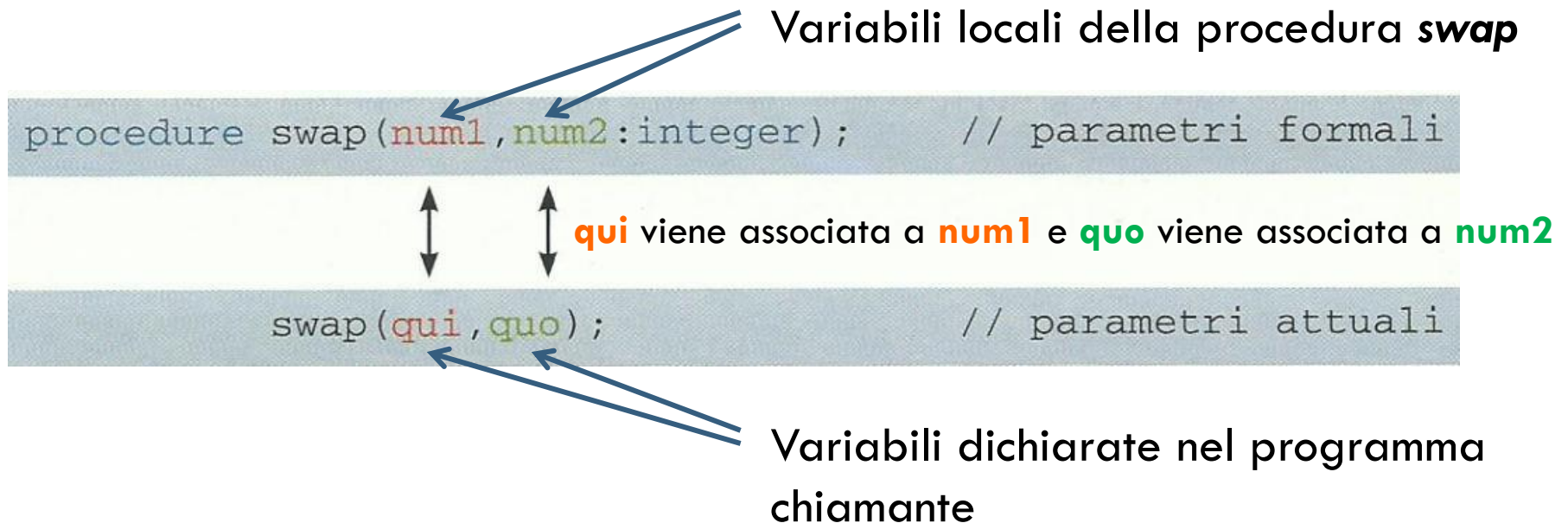
```
0 procedure swap(num1, num2:integer);
1 var
2     tempo:integer;           // variabile locale della procedura
3 begin                       // inizio procedura swap
4     tempo:=num1;
5     num1:=num2;
6     num2:=tempo;
7     writeln(num1, num2);
8 end;                         // fine procedura swap
```

- Nella procedura *swap* ci sono tre variabili:
 - **tempo**: variabile locale definita nella procedura
 - **num1**, **num2**: variabili locali della procedura i cui valori sono assegnati dal programma chiamante
- All'atto della chiamata della procedura bisognerà elencare le variabili contenente i dati sui quali operare

```
swap(qui, quo);
```

Esempio: swap

All'atto della chiamata della procedura viene fatto un'associazione delle variabili in base all'ordine con cui sono scritte



L'elenco dei parametri che vengono scritti all'atto della chiamata prende il nome di **lista dei parametri attuali** in quanto indica il nome delle variabili dalle quali attualmente, cioè all'istante della chiamata, la procedura deve prendere i dati da elaborare.

Osservazioni

- La procedura swap può essere chiamata più volte usando diversi valori all'atto della chiamata
- Con il passaggio dei parametri per valore non vengono modificati i contenuti delle variabili del programma chiamate
 - Questo perché la procedura opera su una copia locale della variabile indicata all'atto della chiamata
 - Al termine dell'esecuzione della procedura tutte le copie (cioè le variabili locali) vengono “rilasciate”, cioè cancellate dalla memoria

Procedure con parametri di tipo diverso

- È possibile passare alle procedure diversi tipi di parametri

```
0 program ordine;
1 (* procedure con parametri per valore di tipo diverso *)
2 var
3   qui, quo, qua: integer;
4   crescente: boolean;
5   procedure mostra(num1, num2: integer; minmax: boolean);
6   begin
7     if minmax=TRUE           // dal più piccolo al più grande
8     then
9       if num1<num2
10      then writeln(num1, ' ', num2)
11      else writeln(num2, ' ', num1)
12     else                     // dal più grande al più piccolo
13       if num1>num2
14       then writeln(num1, ' ', num2)
15       else writeln(num2, ' ', num1);
16   end;                       // fine procedura mostra
17 begin                         // inizio programma principale
18   write('introduci il valore di qui quo qua ');
19   readln(qui, quo, qua);
20   crescente:=TRUE;
21   mostra(qui, quo, crescente); // prima chiamata procedura
22   mostra(quo, qua, crescente); // seconda chiamata procedura
23   mostra(10, 20, FALSE);      // terza chiamata con valori
24   readln;
25 end.                          // fine programma principale
```


Esercizi 1-2, pag. 18

Qual è il risultato dei seguenti codici?

```
1 program prova1;
  var
    x,y:integer;
  procedure ali(num1,x:integer);
  begin
    x:=num1+2;
    y:=-2*x;
    writeln(x,y);
  end;
begin
  x:=2;y:=3;
  ali(x,y);
  writeln(x,y);
  ali(y,x);
  writeln(x,y);
end.
```

```
2 program prova2;
  var
    x,y:integer;
  procedure ali(num1:integer);
  begin
    x:=num1*2;
    y:=num1-2*x;
    writeln(x,y);
  end;
begin
  x:=2;y:=3;
  ali(x);
  writeln(x,y);
  ali(y);
  writeln(x,y);
end.
```

Esercizi 3-4, pag. 18

```
3 program prova3;
var
  x,y:integer;
procedure ali(num1,num2:integer);
begin
  if(num1<0)
  then
    x:=num1+2
  else
    y:=num2-2*x;
  writeln(x,y);
end;
begin
  x:=2;y:=3;
  ali(x,y);
  writeln(x,y);
  ali(y,x);
  writeln(x,y);
end.
```

```
4 program prova4;
var
  x,y:integer;
procedure ali(num1,num2:integer);
begin
  if(num1 mod 2=0)
  then
    x:=num1+1
  else
    y:=num2-1;
  writeln(x,y);
end;
begin
  x:=3;y:=4;
  ali(x,y);
  writeln(x,y);
  ali(y,x);
  writeln(x,y);
end.
```

Esercizi n. 5 pag. 18, n. 6 pag. 19

5 Qual è l'output con i seguenti input?

a) 3, 4 b) 1, -1 c) 6, 3

```
program prova5;
var
  x,y:integer;
procedure ali(num1:integer);
begin
  if (num1<0)
  then
    x:=num1+1
  else
    y:=num1-1;
  writeln(x,y);
end;
procedure baba(num2:integer);
begin
  if(num2>0)
  then
    x:=num2*2
  else
    y:=num2-2;
  writeln(x,y);
end;
begin
  read(x,y);
  ali(x);
  writeln(x,y);
  baba(y);
  writeln(x,y);
end.
```

6 Qual è l'output con i seguenti input?

a) 3, 4 b) 1, 1 c) 6, 3

```
program prova6;
begin
var
  x,y:integer;
procedure ali(num1:integer);
begin
  if(num1<x)
  then
    x:=num1+1
  else
    y:=num1-1;
  writeln(x,y);
end;
procedure baba(num2:integer);
begin
  if(num2>y)
  then
    x:=num2*2
  else
    y:=num2-2;
  writeln(x,y);
end;
begin
  read(x,y);
  ali(x);
  writeln(x,y);
  baba(y);
  writeln(x,y);
end.
```


Problemi 1-9, pag. 19

- 1 Scrivi un programma che legge il valore dell'area di un quadrato e calcola l'area di una circonferenza a esso circoscritta: queste operazioni devono essere ripetute tre volte.
- 2 Scrivi un programma che leggendo il valore dell'area di un triangolo rettangolo calcola il lato di un quadrato a esso equivalente: queste operazioni devono essere ripetute tre volte.
- 3 Scrivi un programma che legge il valore dei lati di un rettangolo, calcola il perimetro e l'area e ripete queste operazioni finché non vengono inseriti valori che li rendono uguali.
- 4 Scrivi una procedura che ricevendo due numeri in ingresso visualizza tutti i multipli di 3 che vi sono compresi.
- 5 Scrivi un programma che utilizza una procedura che verifica se tre numeri sono una terna pitagorica (si ha una terna pitagorica se $x^2 + y^2 = z^2$).
- 6 Scrivi una procedura che converte un numero che indica i secondi in ore, minuti e secondi.
- 7 Scrivi una procedura che ricevendo in ingresso la data di nascita calcola il tempo di vita vissuta espresso in ore.
- 8 Scrivi un programma che legge un insieme di numeri da tastiera e termina l'elaborazione quando viene inserito un numero primo.
- 9 Scrivi un programma che esegue la scomposizione di un numero in fattori primi utilizzando la procedura definita nell'esercizio precedente.



Le procedure con i parametri passati per indirizzo

Parametri per indirizzo

- Quando si effettua il passaggio dei parametri per valore, al termine dell'esecuzione della procedura i valori memorizzati nelle variabili locali si cancellano
- Tuttavia ci sono casi in cui è necessario tenere traccia (*salvare in modo permanente*) del risultato di un'elaborazione fatta da una procedura
- Ciò è possibile con il **passaggio per indirizzo** dei parametri

Passaggio per indirizzo

- Con il passaggio per indirizzo non viene scambiato il valore contenuto nelle variabili, ma viene passato l'indirizzo di memoria della variabile contenente il valore
- In questo modo le operazioni hanno effetto direttamente sulla variabile scambiata e dichiarata nel programma principale (chiamante)
- Il passaggio per indirizzo viene indicato così:

```
procedure modulo(var numero:integer);
```



Esempio: modulo di un numero

```
0 program indirizzo;
1 (* procedura con parametri per indirizzo *)
2 var
3   qui, quo: integer;
4   procedure modulo(var numero: integer);
5   begin
6     if numero < 0           // cambia il segno
7       then
8         numero := -1 * numero
9   end;                       // fine procedura modulo
10 begin                          // inizio programma principale
11   write('introduci il valore di qui quo ');
12   readln(qui, quo);
13   modulo(qui);                 // prima chiamata procedura
14   modulo(quo);                 // seconda chiamata procedura
15   write('dopo la procedura i valori sono ');
16   writeln(qui, ' ', quo);
17   readln;
18 end.
```

```
introduci il valore di qui quo -2 -3
dopo la procedura i valori sono 2 3
```

Parametri per indirizzo e per valore

- È possibile fare uno scambio misto dei parametri
- Per esempio, nella procedura che segue una variabile è scambiata per valore e l'altra per indirizzo:

```
procedure swap(num1:integer; var num2:integer);
```



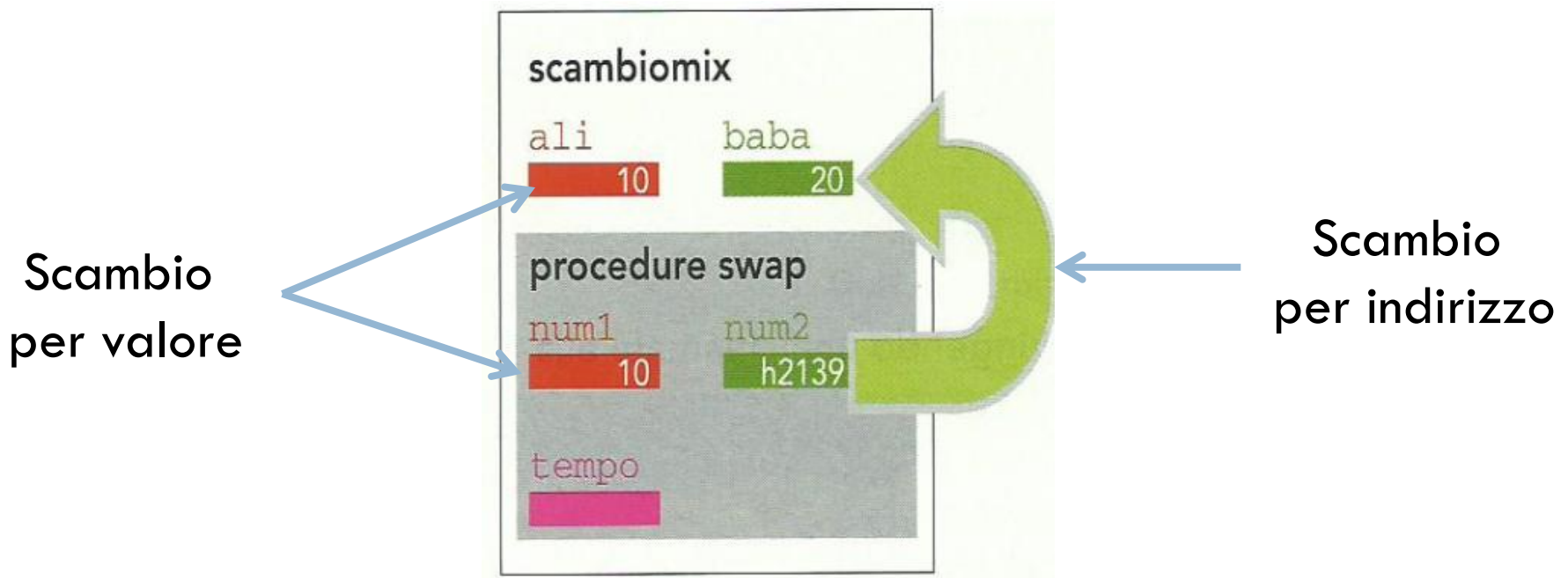
Scambio
per valore



Scambio
per indirizzo

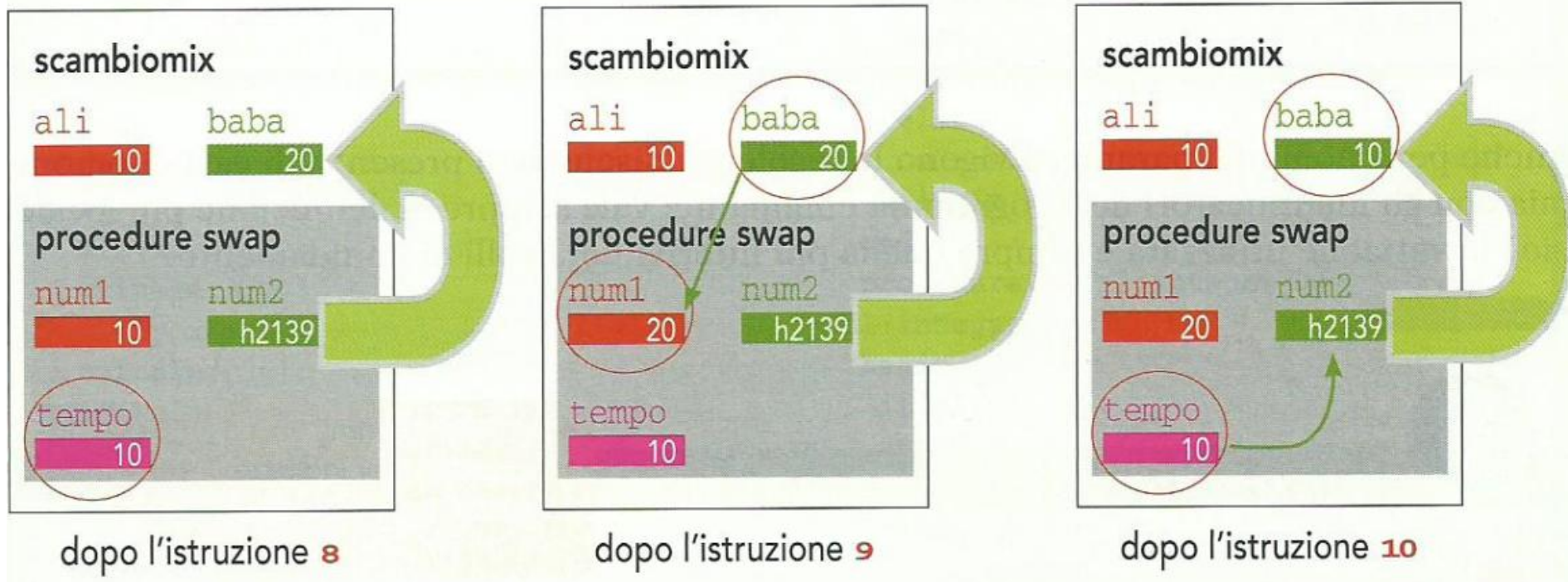
Modello ad ambienti

- Supponendo di aver assegnato alle variabili *ali* e *baba* rispettivamente i valori 10 e 20, all'atto della chiamata della procedura *scambiomix* il modello ad ambienti (rappresentazione grafica della memoria) è il seguente:



Modello ad ambienti (esecuzione procedurale)

- Dopo l'esecuzione della procedura si ha:



L'output
invece sarà:

```
introduci il valore di ali e baba 10 20  
20 10  
10 10
```


Procedure swap corretta

- Pertanto una procedura che effettua lo swap deve necessariamente scambiare i parametri per indirizzo

```
4 procedure swap(var num1:integer;var num2:integer);
```

- In definitiva abbiamo che:

Il risultato che otteniamo al termine della chiamata alla procedura è “notevolmente” diverso se i parametri sono passati per valore o per indirizzo. È quindi necessario prestare molta attenzione in sede di progetto del programma e utilizzare:

- il **meccanismo per valore**: se non interessa mantenere i risultati al termine della procedura;
- il **meccanismo per indirizzo**: se si vuole rendere permanenti le modifiche alle variabili.

Esercizi 1-2, pag. 25

Qual è il risultato dei seguenti codici?

```
1 program prova;
var
  x,y:integer;
procedure ali(var num1,num2:integer)
begin
  if(num1<0)
  then
    x:=num1+2
  else
    y:=num2-2*x;
  writeln(x,y);
end;
begin
  x:=2;y:=3;
  ali(x,y);
  writeln(x,y);
  ali(y,x);
  writeln(x,y);
end.
```

```
2 program prova2;
var
  x,y:integer;
procedure ali(var n1,n2:integer)
begin
  if(n1 mod 2=0)
  then
    x:=n1+1
  else
    y:=n2-1;
  writeln(x,y);
end;
begin
  x:=3;y:=4;
  ali(x,y);
  writeln(x,y);
  ali(y,x);
  writeln(x,y);
end.
```

Problemi 1-6, pag. 26

- 1 Scrivi una procedura che effettua lo scambio di due variabili passate per indirizzo. Quindi utilizzala in un programma che, leggendo quattro numeri, deve individuare il maggiore.
- 2 Leggi un numero intero (compreso tra 1 e 10) e calcola il volume del cubo avente tale numero come lato: successivamente utilizza tale valore come lato di un nuovo cubo e calcola il volume del nuovo cubo così costruito.
- 3 Scrivi una procedura che, ricevendo in ingresso le coordinate di due punti, ne ritorna la coordinata del punto medio.
- 4 Scrivi una procedura che, ricevendo in ingresso due numeri x e n , calcola la potenza ennesima x^n esclusivamente mediante moltiplicazioni.
- 5 Scrivi un programma che genera casualmente N dati e li memorizza in un vettore `vett1` di N elementi. Scrivi due procedure che individuano rispettivamente l'elemento minore e maggiore del vettore ricevendo in ingresso l'indirizzo del vettore da elaborare.
- 6 Scrivi un programma che legge N dati e li memorizza in un vettore `vett1` di N elementi: quindi costruisci partendo da questo altri due vettori `vett2`, `vett3` di dimensioni opportune, contenenti rispettivamente gli elementi di `vett1` non maggiori e maggiori di un intero `NUM` inserito dall'utente.
Scrivi una procedura che ordina tutti i vettori ricevendo in ingresso l'indirizzo del vettore da ordinare.

Problemi 7-9, pag. 26

- 7 Scrivi una procedura che legge un numero e memorizza in un vettore i suoi fattori primi utilizzando una procedura `scomponi()` che riceve come parametri il numero letto e l'indirizzo del vettore nel quale scrivere i risultati.
- 8 Scrivi un programma che implementa il gioco nel quale l'utente deve indovinare un numero segreto entro un numero massimo di tentativi definendo:
 - ▶ la procedura `generaNumero()`:
 - che permette di generare casualmente un numero di 4 cifre;
 - ▶ la procedura `indovinaNumero()`:
 - che permette di inserire un numero;
 - che confronta il numero da individuare con quello inserito e visualizza il messaggio “troppo grande” o “troppo piccolo”;
 - che, se il numero inserito è corretto, fa sì che la funzione ritorni VERO, altrimenti ritorni FALSO.
- 9 Leggi una sequenza di numeri interi positivi diversi da 0. I valori sono letti in input da tastiera e hanno come tappo l'inserimento dello 0.
Presenta un menu per l'esecuzione delle seguenti opzioni, ciascuna realizzata mediante una procedura:
 - ▶ calcolo del numero di coppie di numeri consecutivi uguali;
 - ▶ calcolo del numero di coppie in cui il secondo numero è divisore del primo;
 - ▶ calcolo del numero di coppie in cui il secondo numero è multiplo del primo.