

Esercizio 1

```
/*      Questo programma legge in ingresso 3 interi
 *      e ne calcola la media. Gli interi devono essere
 *      compresi fra 18 e 30.
 *
 *      N.B.: Non viene effettuato nessun controllo sugli
 *      interi inseriti dall'utente
 *
 *      Algoritmo:
 *      Passo 1: Leggo tre interi.
 *      Passo 2: Calcolo la media.
 *      Passo 3: Stampo il risultato.
 */
```

```
#include <stdio.h>

void main()
{int a,b,c;
 float media;

/* Passo 1 */
printf("Inserire i tre voti [18..30]: ");
scanf("%d%d%d", &a, &b, &c);

/* Passo 2 */
media = (a+b+c)/3.0;

/* Passo 3 */
printf("Valore della media: %f", media);
}
```

OSSERVAZIONI:

1)

```
/*    Passo 2    */  
media = (a+b+c)/3.0;
```

Perché ho scritto 3.0 e non più semplicemente 3 ?

Perché il risultato deve essere reale e se usassi il 3 la divisione verrebbe interpretata come divisione tra interi.

2)

Cosa succede se l'utente digita dei valori diversi dall'intervallo specificato ?

Il programma funziona lo stesso.

Con l'istruzione if si potrebbero controllare i dati inseriti.

Esercizio 1 – Variante 1 (unico if)

```
#include <stdio.h>  
  
void main()  
{int a,b,c;  
  float media;  
  
  printf("Inserire i tre voti [18..30]: ");  
  scanf("%d%d%d", &a, &b, &c);  
  
  if ((a<18)|| (a>30)|| (b<18)|| (b>30)|| (c<18)|| (c>30))  
    printf("I numeri inseriti non sono corretti.\n");  
  else  
    {media = (a+b+c)/3.0;  
    printf("Valore della media: %f", media);  
    }  
}
```

Esercizio 1 – Variante 1 (if innestati)

```
#include <stdio.h>

void main()
{int a,b,c;
 float media;
 printf("Inserire i tre voti [18..30]: ");
 scanf("%d%d%d", &a, &b, &c);

 if ((a<18)||(a>30)) /*if #1*/
   printf("I numeri inseriti non sono corretti.\n");
 else /*if #1*/
   if ((b<18)||(b>30)) /*if #2*/
     printf("I numeri inseriti non sono corretti.\n");
   else /*if #2*/
     if ((c<18)||(c>30)) /*if #3*/
       printf("I numeri inseriti non sono corretti.\n");
     else /*if #3*/
       {media = (a+b+c)/3.0;
        printf("Valore della media: %f", media);
       }
}
```

Esercizio 1 – Variante 2 (unico while)

```
#include <stdio.h>

void main()
{int a,b,c;
 float media;

 printf("Inserire i tre voti [18..30]: ");
 scanf("%d%d%d", &a, &b, &c);

 while ((a<18)||(a>30)||(b<18)||(b>30)||(c<18)||(c>30))
 {printf("I numeri inseriti non sono esatti.\n");
  printf("Inserire i tre voti [18..30]: ");
  scanf("%d%d%d", &a, &b, &c);
 }
 media = (a+b+c)/3.0;
 printf("Valore della media: %f", media);
}
```

Esercizio 1 – Variante 2 (while distinti)

```
#include <stdio.h>

void main()
{int a,b,c;
 float media;
 printf("Inserire i tre voti [18..30]: ");
 scanf("%d%d%d", &a, &b, &c);

 while ((a<18)|| (a>30))
 {printf("Il primo numero inserito non e' corretto.\n");
  printf("Inserire il primo voto [18..30]: ");
  scanf("%d", &a);
 }
 while ((b<18)|| (b>30))
 {printf("Il secondo numero inserito non e' corretto.\n");
  printf("Inserire il secondo voto [18..30]: ");
  scanf("%d", &b);
 }
 while ((c<18)|| (c>30))
 {printf("Il terzo numero inserito non e' corretto.\n");
  printf("Inserire il terzo voto [18..30]: ");
  scanf("%d", &c);
 }
 media = (a+b+c)/3.0;
 printf("Valore della media: %f", media);
}
```

Esercizio 2

```
/*      Questo programma legge in ingresso 3 interi e ne
 *      calcola la media prima in 30-esimi e poi in 100-esimi.
 *      Gli interi devono essere compresi fra 18 e 30, ma non
 *      viene effettuato nessun controllo sulla loro correttezza.
 *
 *      Algoritmo:
 *      Passo 1:      Leggo tre interi
 *      Passo 2:      Calcolo la media in 30-esimi
 *      Passo 3:      Calcolo la media in 100-esimi
 *      Passo 4:      Stampo le medie
 */
```

```

#include <stdio.h>

void main()
{
    int a, b, c;
    float media30, media 100;

    /* Passo 1 */
    printf("Inserire i tre voti [18..30]: ");
    scanf("%d%d%d", &a, &b, &c);

    /* Passo 2 */
    media30 = (a+b+c)/3.0;

    /* Passo 3 */
    media100 = (a+b+c)/3.0*(100/30.0));

    /* Passo 4 */
    printf("Media in 30-esimi: %f", media30);
    printf("\nMedia in 100-esimi: %f", media100);

}

```

OSSERVAZIONI:

```

/*    Passo 3    */
media100 = (a+b+c)/3.0*(100/30.0));

```

Perché ho scritto 30.0 e non più semplicemente 30 ?

Perché se scrivessi 30 il risultato di $100/30$ sarebbe 3 (divisione tra due interi) e non 3.333..., avrei quindi moltiplicato la media in trentesimi per 3 e la media in centesimi risulterebbe scorretta.

Se avessi scritto $100/30$ senza parentesi il risultato sarebbe stato corretto. Gli operatori * e / hanno la stessa precedenza e associatività da sinistra; viene prima eseguita la divisione per 3.0 (risultato float), poi la moltiplicazione per 100 e infine la divisione per 30. Il risultato finale è quindi un float.

Esercizio 3

```
/*      Questo programma legge un intero da input e ne
 *      restituisce il valore assoluto.
 *
 *      Algoritmo:
 *      Passo 1: Leggo un numero da input
 *      Passo 2: Calcolo il valore assoluto di questo numero
 *      Passo 3: Stampo il valore assoluto
 */
```

```
#include <stdio.h>
void main()
{
    int x;
    int abs_x;

    /* Passo 1 */
    printf("Inserire un numero intero: ");
    scanf("%d", &x);

    /* Passo 2 */
    abs_x = ( x>0 ? x : (-x) );

    /* Passo 3 */
    printf("Il valore assoluto di %d e': %d", x, abs_x);

}
```

Esercizio 4

```
/*      Questo programma legge in ingresso un intero che
 *      rappresenta una quantità di denaro in lire, e lo converte
 *      in euro e in dollari in base ad un carattere.
 *
 *      Algoritmo:
 *      Passo 1: Leggo un intero che rappresenta il valore in lire
 *      Passo 2: Effettuo la conversione in euro e in dollari
 *      Passo 3: Stampo i nuovi valori
 */
```

```
#include <stdio.h>

void main()
{
    int lire;
    float conv_euro, conv_dollari;
    char c;

    /* Passo 1 */
    printf("Inserire l'importo in lire: ");
    scanf("%d", &lire);
    scanf("%*c");
    scanf("%c", &c);

    /* Passo 2 */
    conv_euro = lire/1936.27;
    conv_dollari = lire/1525.0;

    /* Passo 3 */
    (c=='e') ? printf("Valore in euro: %f", conv_euro) :
              printf("Valore in dollari: %f", conv_dollari);
}
```

OSSERVAZIONI

Cosa succede se l'utente digita una lettera qualsiasi diversa sia da 'e' che da 'd'. Come posso modificare questo comportamento utilizzando solo l'espressione condizionale (?:) ?

L'inserimento di `scanf("%*c")` e' necessario per evitare un malfunzionamento dovuto alla combinazione di due fattori: input/output bufferizzato e codifica dei caratteri.

Tutto ciò che e' inserito da tastiera viene memorizzato in un'area di memoria (detta BUFFER), alla quale accede la scanf per prelevare i dati. L'inserimento dei dati termina quando si genera un carattere di fine inserimento, tipicamente si preme ENTER (INVIO). E' a quel punto che la scanf preleva i dati dal buffer e, ad ogni chiamata, tenta di prelevare dati dal buffer, se non ne trova si mette in attesa (e noi inseriamo da tastiera).

Ai tasti della tastiera sono associati i codici corrispondenti ai caratteri; nei sistemi DOS/Windows il tasto "ENTER" ("INVIO") e' associato a due caratteri ed entrambi sono memorizzati nel buffer. Quando si inserisce un intero (per esempio) il primo carattere (byte) associato al tasto ENTER viene considerato come terminazione dell'inserimento dati, ma il secondo rimane nel buffer. Quindi una successiva `scanf("%c",&car)` trova subito un byte disponibile nel buffer e lo assegna alla variabile `car`. E' per questo motivo che e' necessario effettuare una lettura "a vuoto" (l'asterisco nella stringa di formato indica alla scanf che non deve memorizzare cio' che legge).

Attenzione: questo fenomeno si verifica solo nel caso in cui si debba leggere un carattere e dopo avere fatto qualche altra operazione di lettura (cioè quando nel buffer e' rimasto qualche dato). Nel caso di numeri, questo non si verifica, perché il carattere residuo viene scartato (non potendo essere interpretato come numero).