

## Esercizio 1

```
#include <stdio.h>

void main()
{
    int bit;          /* l'elemento della sequenza appena letto */
    int cont = 0;      /* lunghezza attuale della sequenza di zeri;
                        inizialmente non abbiamo ancora letto nulla */

    int maxlung = 0;   /* valore temporaneo della massima lunghezza;
                        inizialmente e` 0 */

    printf("Inserisci una sequenza di 0, 1 terminata da 2\n");
    do
    {scanf("%d", &bit);
     if (bit == 0)      /* e` stato letto un altro 0 */
     {cont++;          /* aggiorna la lunghezza della sequenza corrente*/
      if (cont > maxlung) /* se necessario, aggiorna il massimo temporaneo*/
        maxlung = cont;
     }
     else              /* e` stato letto un 1 (o il 2 finale) */
     {cont = 0;        /* azzera la lunghezza della sequenza corrente */
     } while (bit != 2); /* quando si legge 2 il ciclo termina */

    printf("La piu` lunga sottosequenza di soli zeri e` lunga %d\n", maxlung);

    return 0;
}
```

## Esercizio 2

```
#include <stdio.h>
void main()
{int N,voto,max,min,i;
 float media=0;
 printf("Inserire il numero dei voti: ");
 scanf("%d",&N);
 for (i=0; i<N; i++)
 {do
 {printf("Voto [18..30]: ");
  scanf("%d",&voto);
 }
 while ((voto<18)|| (voto>30));
 if (i==0)
 {max = min = voto;
  media = voto;
 }
 else
 {if (voto>max) max = voto;
  if (voto<min) min = voto;
  media += voto;
 }
 }
 printf("\nVoto max:%d\tVoto min:%d\tMedia:%f\n",max,min,media/N);
}
```

### Osservazione:

- L'inizializzazione di max e min è stata fatta utilizzando il primo valore inserito.

### Variante (con do...while):

```
#include <stdio.h>
void main()
{int N,voto,max=18,min=30,i=0;
 float media=0;
 printf("Inserire i voti [18..30] e terminare con 0:\n");
 do
 {printf("Voto: ");
  scanf("%d",&voto);
  if ((voto >= 18)&&(voto <= 30))
  {if (voto>max) max = voto;
   if (voto<min) min = voto;
   media+=voto;
   i++;
  }
 } while (voto != 0);
 if (i==0) printf("Non e' stato inserito nessun voto valido");
 else printf("\nVoto max:%d\tVoto min:%d\tMedia:%f\n",max,min,media/i);
}
```

### Osservazione:

- Poiché i voti sono compresi tra 18 e 30, i valori max e min sono stati inizializzati rispettivamente al minor voto massimo (18) e al maggior voto minimo (30)

### Variante (con while):

```
#include <stdio.h>
main()
{int N,voto,max,min,i;
 printf("Inserire i voti e terminare con 0:\n");
 printf("Voto: ");
 scanf("%d",&voto);
 max = min = voto;
 while (voto != 0)
 {if (voto>max) max = voto;
  else if (voto<min) min = voto;
  printf("Voto: ");
  scanf("%d",&voto);
 }
 printf("\nVoto max:%d\tVoto min:%d\n",max,min);
}
```

Osservazione: è stato introdotto un ELSE per evitare di valutare la condizione del secondo IF; infatti, se voto>max allora sarà sicuramente voto>min e il secondo confronto può essere evitato. Si osservi che, con questa modifica, non sarebbe corretto (dal punto di vista logico) inizializzare MAX e MIN a 18 e 30 (perché?).

### Esercizio 3

```
#include <stdio.h>

void main()
{int num1, num2, resto, m, n, mcm, MCD;

    printf("Inserire due numeri interi: ") ;
    scanf("%d%d", &num1, &num2);

    /*Calcolo il Massimo Comune Divisore*/

    m = (num2 > num1) ? num2 : num1;
    n = (num2 > num1) ? num1 : num2;
    while (n > 0)
        {resto = m % n;
          m = n;
          n = resto;
        }
    /* a questo punto la variabile m contiene il MCD*/

    MCD = m;

    /*Calcolo il minimo comune multiplo*/
    mcm = (num1 * num2) / MCD;

    printf("\nIl minimo comune multiplo tra %d e %d e` %d\n", num1, num2, mcm);
    printf("\nIl massimo comune divisore tra %d e %d e` %d\n", num1, num2, MCD);
}
```

### Esercizio n.4

```
#include <stdio.h>

int main(void)
/* Legge una sequenza di numeri interi di lunghezza letta da tastiera.
   Stampa la somma dei numeri positivi e la somma dei numeri negativi. */
{
    int i;                                /* contatore di ciclo */
    int lung;                             /* lunghezza della sequenza */
    int numero;                           /* numero letto di volta in volta */
    int somma_pos, somma_neg;             /* somme parziali degli interi positivi e
                                          negativi letti fino ad un certo punto */

    printf("Immettere la lunghezza della sequenza di numeri interi!\n");
    scanf("%d", &lung);
    printf("Immettere %d numeri interi, uno per riga!\n", lung);

    somma_pos = 0;                        /* prima di leggere il primo numero
                                          la somma parziale dei numeri positivi e` 0 */
    somma_neg = 0;                        /* prima di leggere il primo numero
                                          la somma parziale dei numeri negativi e` 0 */

    i = 1;
    while (i <= lung) {
        scanf("%d", &numero);
        if (numero >= 0)
            somma_pos += numero;
        else
            somma_neg += numero;
    }
}
```

```
        i++;
    }

    printf("Somma dei positivi: %d\n", somma_pos);
    printf("Somma dei negativi: %d\n", somma_neg);

    return 0;
} /* main */
```

### Esercizio n.5

```
#include <stdio.h>

main()
{int mese,giorni;
 do
    {printf("Inserire il numero corrispondente al mese: ");
      scanf("%d",&mese);
    }while (mese<1 || mese>12);
 switch (mese)
 {case 2 : giorni = 28; break;
  case 4 :
  case 6 :
  case 7 :
  case 9 : giorni = 30; break;
  default : giorni = 31;
 }
 printf("Il mese %d ha %d giorni",mese,giorni);
}
```

### Esercizio n.6

```
#include<stdio.h>

void main()
{int n1 ,n2 ,d1 ,d2, resto;
 printf("Inserire le due frazioni n1/d1 n2/d2: ");
 scanf("%d/%d %d/%d",&n1,&d1,&n2,&d2);
 //   a/b + c/d = (ad + cb)/(bd)
 //   Riutilizzo le variabili
 n1 = n1*d2 + n2*d1;
 d1 = d1*d2;

 n2 = (n1 > d1) ? n1 : d1;
 d2 = (n1 > d1) ? d1 : n1;
 while (d2 > 0)
 {resto = n2 % d2;
  n2 = d2;
  d2 = resto;
 }
 /* a questo punto la variabile n2 contiene il MCD tra n1 e d1*/

 printf("La frazione e' %d/%d",n1/n2,d1/n2);
}
```

**Osservazione:** Le divisioni nell'ultima printf sono tra interi, ma dato che n2 è il MCD di n1 e d1, le divisioni danno un risultato intero e non si rischia di perdere precisione.