

(A) CONOSCENZA TERMINOLOGICA**Dare una breve descrizione dei termini introdotti:**

- Sequenza
- Selezione semplice
- Selezione doppia
- Selezione multipla
- Iterazione predefinita
- Iterazione precondizionata
- Iterazione postcondizionata
- Contatore
- Accumulatore
- Operatore ternario

(B) CONOSCENZA E COMPETENZA**Rispondere alle seguenti domande producendo anche qualche esempio***Conoscenza*

1. A cosa serve la *struttura di controllo sequenziale*?
2. A cosa serve la *struttura di controllo selettiva*?
3. A cosa serve la *struttura di controllo iterativa*?
4. Quali sono le strutture di controllo iterative e qual è la loro sintassi?
5. A cosa serve un *contatore*?
6. A cosa serve un *accumulatore*?

Competenza

1. Qual è la sintassi della *struttura di controllo sequenziale*?
2. Qual è la sintassi della *struttura di controllo selettiva semplice*?
3. Qual è la sintassi della *struttura di controllo selettiva doppia*?
4. Qual è la sintassi della *struttura di controllo selettiva multipla*?
5. Qual è la sintassi della *struttura di controllo iterativa predefinita*?
6. Qual è la sintassi della *struttura di controllo iterativa precondizionata*?
7. Qual è la sintassi della *struttura di controllo iterativa postcondizionata*?

(C) ESERCIZI DI COMPrensIONE

1. La s.d.c. iterativa si usa quando si conosce il numero di cicli che il programma deve eseguire. Il numero di volte che il ciclo viene ripetuto è segnalato dal
2. La s.d.c. iterativa si dice così perché prevede che la condizione di ripetizione del ciclo sia scritta del ciclo. Questa s.d.c. ripete il ciclo per la condizione ed esce quando questa diventa Se la condizione è già dall'inizio, il ciclo potrebbe essere eseguito volte.
3. La s.d.c. iterativa si dice così perché prevede che la condizione di ripetizione del ciclo sia scritta il ciclo. Questa s.d.c. ripete il ciclo per la condizione ed esce quando questa diventa Anche se la condizione è già dall'inizio, il ciclo viene sempre eseguito almenovolta.
4. Quando si esegue un ciclo predefinito, è necessaria la presenza di una variabile, che viene detta, la quale serve a tenere traccia del numero di volte che il ciclo è stato eseguito, in modo da poterne stabilire la fine.

5. Per ciascuna delle s.d.c. riportate in tabella, indicarne la sintassi in codifica.

s.d.c.	Sintassi
Sequenziale	
Selezione semplice	
Selezione doppia	
Selezione multipla	
Iterazione predefinita	
Iterazione precondizionata	
Iterazione postcondizionata	

6. Dato l' algoritmo seguente:

Algoritmo PariDispari

Inizio

Intero numero;

Leggi (numero);

Se (numero %2==0)

Stampa "Il numero e' pari";

Altrimenti

Stampa "Il numero e' dispari";

Fine.

- scrivere nel riquadro a fianco la codifica;
- scrivere il testo del problema nel riquadro sottostante:

```
/* PariDispari.cpp */
```

- c. se l'istruzione

Se (numero %2==0)

venisse sostituita con

Se (numero %2)

cosa dovrebbe cambiare nell'algoritmo?

7. Descrivere il funzionamento dei seguenti frammenti di programmi e determinare l'output che si ottiene.

- a. **int** a, b, c=0;

.....

a=3;

b=6;

if (b>a) c=b-a;

else c=a-b;

cout<<c<<endl;

- b. **int** a, b, c;

a=3;

b=5;

(a<b)? c=a: c=b;

cout<< c<<endl;

```

c. int cont;
   -----
   for (cont=0; cont<=5; cont++)
   if (!(cont%2))
       cout<<cont<<endl;
   -----

d. ...
   int main()
   {
       int i=1;
       char r;
       .....
       do
       { i++;
         cin>>r;
         .....
       } while ((i<=10) && (r!='f'));
       cout<<((i>10) || (r=='f'));
       .....
   }

e. ...
   int v;
   while (1)
   { cout<<"Valore: ";
     cin>>v;
     if ((v==0) || (v==1))
         break;
   }
   cout<<v<<endl;

f. ...
   int n, f=1;
   ....
   for (; n>2; n--)
       f*=n-1;
   cout<<"f = "<<f<<endl;
   ...

g. ...
   int i=0, m, n;
   cin>>n;
   if (n<0)
       cout<<"Valore non ammesso";
   else
   { cin>>n;
     for (i=1; i<=n; i++)
         m+=n;
     cout<<m;
   }
   ....

h. ...
   int a, b, v;
   ....
   a=3, b=5;
   v=(a==b)? a+1 : b-1;
   cout<<v<<endl;

i. ....
   ...
   int a, b, v, c;
   ....
   a=2, b=2;
   v=3*(a==b? a+2 : b=-2) +c;
   cout<<v<<endl;
   ....

j. ...
   char ch;
   for (;)
   { ch=getchar();
     if ((ch=='A') || (ch=='B'))
         break;
     ....
   }
   cout<<ch<<endl;
   ...

k. ...
   int x, y;
   for (x=0; x<100; x++)
   {
       y=1;
       for (;)
       {
           cout<<y<<endl;
           if (y==10) break;
       }
   }

l. ...
   int x;
   for (x=0; x<100; x++)
   {
       if (x%2) continue;
       cout<<x<<endl;
       if (x==10) break;
   }....

m. ...
   int i;
   for (i=1; i<=100; i++)
   cout<<i<<" ";
   cout<<(float)i/3<<endl;
   ....

n. ...
   int i;
   for (i=0; i<100; i++)
   cout<<((i%10==0)? i : ' ')<<endl;
   ....

o. ...
   int a=5;
   do
   { a*=3;
     cout<<a<<endl;
   } while (a<=50); ....

```

8. Riscrivere utilizzando la **while** i frammenti di programmi dell'esercizio 1 che fanno uso della **for**.
9. Il seguente programma contiene errori sintattici. Dopo averli corretti, descrivere l'effetto del programma e i valori in output.

```
#include <iostream>
int main()
{ for (i=0; I<20; i++)
  { if i%2 continue;
    cout <<i*i<<endl;
  }
}
```

(D) ESERCIZI DI APPLICAZIONE

Esercizi risolti

1. Scrivere un programma che, dopo aver letto tre valori numerici in input, stampi il minimo fra essi.

Fase 1 - Analisi del problema

Analisi del testo

Per risolvere il problema servono tre variabili a , b e c per l'input ed una quarta variabile min , che servirà a contenere il minimo tra a , b e c . Le variabili vengono assunte tutte intere..

Analisi delle specifiche di Ingresso/Uscita

	Nome	Tipo	Significato
Input	a	Intera	Primo valore
	b	Intera	Secondo valore
	c	Intera	Terzo valore
Output	min	Intera	Minimo tra a , b e c
	MSG	Stringa	"Il minimo e' "

Tab. 1 Tabella delle variabili di I/O del problema di Esempio 1

Fase 2 - Algoritmo

Algoritmo Minimo2

Inizio

Leggi (a);

Leggi (b);

Leggi (c);

Se (a < b) // Si calcola il minimo tra a e b

min =a;

Altrimenti

min=b;

Se (c < min) // Si confronta c con la variabile min

min=c;

Stampa (min);

Fine.

Fase 3 – Codifica

/* MinTra3.cpp */

```
#include <iostream>
```

```
#include <stdlib>
```

```
#define MINIMO "Il minimo e': " // definizione di una costante stringa
```

```
int main()
```

```

{ int a, b, c, min;
  cout<<"Primo valore: ";   cin>>a;
  cout<<"Secondo valore: "; cin>>b;
  cout<<"Terzo valore: ";  cin>>c;
  if (a < b)
    min=a;
  else
    min=b;
  if (c < min)
    min=c;
  cout<<MINIMO<<min<<"\n";
  system("PAUSE");
  return 0;
}

```

2. Scrivere un programma che simuli una semplice calcolatrice aritmetica intera.

Fase 1 - Analisi del problema

Analisi del testo

La nostra calcolatrice svolge le quattro operazioni fondamentali; in ingresso abbiamo due variabili intere *primo* e *secondo* che indicano gli operandi ed in uscita una variabile *risultato* che contiene il risultato dell'operazione. In input occorre un'altra variabile *op* che indica l'operazione scelta dall'utente e che dovendo essere uno tra i quattro simboli ('+', '-', '*', o '/') sarà di tipo carattere. Nel caso della divisione occorre controllare che il secondo operando sia diverso da zero, altrimenti occorre gestire l'eccezione.

Analisi delle specifiche di Ingresso/Uscita

	Nome	Tipo	Significato
Input	primo	Intera	Primo operando
	secondo	Intera	Secondo operando
	op	Carattere	Simbolo di operazione scelta
Output	risultato	Intera	Risultato
	riga	Stringa	Mssaggio all'utente

Tab. 2 Tabella delle variabili di I/O del problema di Esempio 2

Procedimento risolutivo

Il programma acquisisce il valore del primo operando nella variabile *primo*, poi acquisisce il simbolo di operazione scelta nella variabile *op* e infine il valore del secondo operando nella variabile *secondo*. Successivamente, a seconda dei vari valori di *op*, pone nella variabile *risultato* il risultato dell'operazione *primo op secondo*. Nell'analisi dei vari casi, se viene scelta l'operazione di divisione, occorre considerare il caso che *secondo* sia nullo: se così, viene posta a **true** la variabile *errore*, precedentemente inizializzata a **false**. La stampa di risultato avviene quando *errore* vale **false**.

Fase 2 - Algoritmo

Algoritmo CalcolatriceSemplice

Inizio

```

riga = "Risultato =";
errore=false;           // variabile booleana per gestire l'eccezione
Leggi (primo);
Leggi (op);
Leggi (secondo);
Nel caso che (op) sia

```


*Fase 2 – Algoritmo***Algoritmo** SommaN**Inizio**

```

Leggi (N);
somma = 0;
Per cont = 1 a N fai
  Inizio
    Leggi (valore);
    somma = somma + valore;
  Fine
Stampa ("La somma e' ", somma);

```

Fine.*Fase 3 - Codifica*

```

/* Somman.cpp */
#include <iostream>
#include <cstdlib>
#define RISPOSTA "La somma e': "
#define NUMERO "Immetti valore: "
using namespace std;
int main()
{ int somma=0,
  N,
  cont,
  valore;
  cout<<"Numero di valori da sommare: ";
  cin>>N;
  for(cont=0;cont<N;cont++)
  { cout<<NUMERO;
    cin>>valore;
    somma+=valore;
  }
  cout<<RISPOSTA<<somma<<"\n";
  system("Pause");
  return 0;
}

```

4. Si leggono i dati di n persone e si vuole stampare una tabella contenente la quantità di maschi e di femmine.

*Fase 1 - Analisi del problema**Analisi del testo*

Abbiamo in input la quantità N delle persona e in output le variabili $contm$ e $contf$ che indicano il numero rispettivo di maschi e di femmine e che devono essere azzerati all'inizio dell'esecuzione del programma. Il calcolo viene fatto mediante un ciclo predefinito e un test che stabilisce, per ciascuna persona, il sesso. Al termine, si stampa il contenuto delle variabili $contm$ e $contf$.

Analisi delle specifiche di Ingresso/Uscita

	Nome	Tipo	Significato
Input	n	Intero	Quantità delle persone
	sex	Carattere	Sesso 'm' oppure 'f'
Output	contm	Intero	Quantità dei maschi
	contf	Intero	Quantità delle femmine

Tab. 4 Tabella delle variabili di I/O del problema di Esempio 4

Procedimento risolutivo

Il programma legge il numero n di persone, azzera i due contatori $contm$ e $contf$ e tramite un ciclo **for** chiede, per ciascuna di esse il sesso, leggendolo in una variabile $Sesso$ di tipo carattere; si esegue un test per verificare se la variabile $Sesso$ contenga 'm' oppure 'f': nel primo caso si incrementa il contatore $contm$, altrimenti si incrementa $contf$.

*Fase 2 – Algoritmo***Algoritmo** ContSex**Inizio**

Intero n , $contm=0$, $contf=0$, $i=0$;

Carattere $Sesso$;

Leggi (n);

Fintantoché ($i < n$) **fai**

Inizio

Leggi ($Sesso$);

Se ($Sesso == 'm'$)

$contm++$;

Altrimenti

$contf++$;

$i++$;

Fine;

Stampa ($contm$);

Stampa ($contf$);

Fine.*Fase 3 - Codifica*

La codifica è riportata di seguito

/ ContSex.cpp */*

```

#include <iostream>
#include <cstdlib>
#include <conio.h>
using namespace std;
int main(int argc, char *argv[])
{
    int n,
        i=0,
        contm= 0, // contatore maschi
        contf= 0; // contatore femmine
    char Sesso ;

    cout<<"\nImmetti il numero delle persone: ";
    cin>>n;
    while (i<n)
    {
        cout<<"\n immetti il sesso: ";
        Sesso = toupper (getche());
        if (Sesso=='M')
        {
            contm = contm + 1;
            i++;
        }
        else
            if (Sesso=='F')
            {

```

```

        contf = contf + 1;
        i++;
    }
    else
        cout<<"Immissione errata!";
}
cout<<"\n il totale dei maschi e': "<<contm<<endl;
cout<<"il totale delle femmine e' "<<contf<<endl;
system ("PAUSE");
return 0;
}

```

OSSERVAZIONI:

- Si noti la selezione sul *sex* utilizzata anche per verificare il caso in cui si abbia una immissione errata (nel qual caso ne viene data comunicazione all'utente);
- si noti la funzione `sex = toupper (getche());` che, trasformando tutti i caratteri in maiuscolo, evita errori in caso di immissione con minuscolo.

5. Scrivere un programma che letti i coefficienti di un'equazione di 1° grado in una incognita, ne determina la soluzione, prevedendo anche i casi di indeterminatezza e impossibilità.

*Fase 1 - Analisi del problema**Analisi del testo*

Supponiamo di trattare il caso generale in cui il coefficiente a e il termine noto b siano numeri reali. Nello studiare l'equazione, nel caso a sia diverso da zero, l'equazione è determinata e ammette l'unica soluzione $x=-b/a$, mentre se a vale zero, allora nel caso b sia diverso da zero l'equazione è impossibile, mentre se b vale zero l'equazione risulta indeterminata.

Analisi delle specifiche di Ingresso/Uscita

Input	Nome	Tipo	Significato
	a	Reale	Coefficiente della x
	b	Reale	Termine noto
Output	msg	Stringa	Messaggio all'utente
	x	Reale	Soluzione dell'equazione

Tab. 5 Tabella delle variabili di I/O del problema di Esempio 5

Procedimento risolutivo

Il programma effettua la lettura del coefficiente a e del termine noto b e poi, a seconda del valore di a stabilisce se l'equazione è determinata ($a \neq 0$) oppure, essendo $a=0$, indeterminata ($b \neq 0$) o impossibile ($b=0$). In ciascun caso carica in *msg* un messaggio opportuno. Nel caso $a \neq 0$, viene calcolata e stampata la soluzione $x=-b/a$. Al termine si stampa il messaggio *msg*.

*Fase 2 – Algoritmo***Algoritmo** EquazionePrimoGrado**Inizio**

```

    Reale a, b, x;
    Stringa msg;
    Leggi (a);
    Leggi (b);
    Se (a==0)
        Se (b==0)

```

```

        msg="Indeterminata";
    Altrimenti
        msg="Impossibile";
    Altrimenti
    Inizio
        msg="Determinata";
        x = -b/a;
    Stampa ("La soluzione è ", x);
    Fine
    Stampa ("L'equazione e' ", msg);
Fine.

```

Fase 3 - Codifica

```

/* Eq1.cpp */
#include <cstdlib>
#include <iostream>
using namespace std;
int main()
{
    string s;
    float a,b,x;
    cout<<"\nImmetti il primo coefficiente:";
    cin>>a;
    cout<<"\nImmetti il secondo coefficiente:";
    cin>>b;
    if (a==0)
        if (b==0) // a=0 e b=0
            s="equazione indeterminata";
        else // a=0 e b!=0
            s="equazione impossibile";
    else // a!=0
    {
        s="equazione determinata";
        x=-b/a;
        cout<<"\nLa soluzione dell'equazione e' "<<x<<endl;
    }
    cout<<s<<endl;
    system("PAUSE");
    return 0;
}

```

6. Si vuole realizzare un semplice programma che risolva un'equazione di 2° grado in un'incognita.

Fase 1 - Analisi del problema

Analisi del testo

Supponiamo di considerare il caso generale in cui che i 3 coefficienti a , b e c siano numeri reali.

Nel caso sia a diverso da zero l'equazione ammette 2 soluzioni che possono essere reali e distinte, reali e coincidenti o complesse coniugate a seconda del valore del discriminante (>0 , $=0$ o <0). Nel caso a sia nullo, occorre esaminare il valore di b : se b è diverso da zero l'equazione si riduce ad una equazione di primo grado, che ha come soluzione $x = -c / b$; se, invece, b vale zero, allora se anche c vale zero, l'equazione è indeterminata, mentre se c è diverso da 0 l'equazione è impossibile

Analisi delle specifiche di Ingresso/Uscita

Input	Nome	Tipo	Significato
	a	Reale	Coefficiente del termine di 2° grado
	b	Reale	Coefficiente del termine di 1° grado
	c	Reale	Termine noto
Output	msg	Stringa	Messaggio all'utente

Tab. 6 Tabella delle variabili di I/O del problema di Esempio 6

Procedimento risolutivo

Il programma effettua la lettura dei 3 coefficienti a , b e c . Poi, a seconda del valore di a stabilisce se l'equazione è di primo grado ($a=0$) oppure di secondo grado ($a \neq 0$).

Se $a=0$, l'equazione è di primo grado per cui: se $b=0$ allora nel caso $c=0$, risulta indeterminata, mentre se $c \neq 0$ è impossibile. Se, invece, $b \neq 0$ l'equazione di primo grado ha la soluzione

$$x = -c/b.$$

Se $a \neq 0$, si calcola il $\Delta = b^2 - 4*a*c$ e si esaminano i 3 casi:

- se $\Delta > 0$ si applica la formula risolutiva delle equazioni di 2° grado che dà i due valori x_1 ed x_2 distinti;
 - se $\Delta = 0$, le due soluzioni sono coincidenti e valgono $x_1 = x_2 = -b/(2*a)$;
 - se $\Delta < 0$, si applica la formula risolutiva delle equazioni ponendo $\Delta = -\Delta$ ed ottenendo:
 - $parte reale = -b/(2*a)$
 - $parte immaginaria = \sqrt{\Delta}/(2*a)$
- e si stampano le soluzioni in forma complessa

Fase 2 – Algoritmo

Algoritmo EquazioneSecondoGrado

Inizio

```

Leggi (a);
Leggi (b);
Leggi (c);
Se (a==0)
  Se (b==0)
    Se (c==0)
      msg="Indeterminata";
    Altrimenti
      msg="Impossibile";
  Altrimenti
    Inizio
      Stampa ("Equazione di 1° grado");
      msg="Determinata";
      x = -c/b;
      Stampa ("La soluzione è ", x);
    Fine
  Altrimenti
    Inizio
      Stampa ("Equazione di 2° grado");
       $\Delta = \sqrt{b*b-4*a*c}$ ;
      Se ( $\Delta > 0$ )
        Inizio

```

```

x1= (-b+sqrt(delta)/(2*a);
x2= (-b-sqrt(delta)/(2*a);
Stampa (x1, x2);
Fine
Altrimenti
    Se (delta<0)
        Inizio
            delta=-delta;
            partereale=-b/(2*a);
            parteimmaginaria=sqrt(delta)/(2*a);
            Stampa (partereale, "+i*", parteimmaginaria);
            Stampa (partereale, "-i*", parteimmaginaria);
        Fine
    Altrimenti
        Inizio
            x1=-b/(2*a);
            Stampa (x1, x1);
        Fine
Fine

```

Fine.

Fase 3 - Codifica

```

/* Eq2.cpp */
#include <iostream>
#include <cstdlib>
#include <math.h>
#define INDETERMINATA "Equazione indeterminata"
#define IMPOSSIBILE "Equazione impossibile"
using namespace std;
int main (int argc, char *args[])
{
    float a,b,c;
    float x1, x2, delta, partereale, parteimmaginaria;
    cout<<"Immetti coefficiente a: ";
    cin>>a;
    cout<<"Immetti coefficiente b: ";
    cin>>b;
    cout<<"Immetti coefficiente c: ";
    cin>>c;
    if (a==0) // se a == 0...
        if (b==0) //... e anche b==0 allora...
            if (c==0) //...se c ==0 l'equazione...
                cout<<INDETERMINATA<<endl; //...è indeterminata, altrimenti...
            else
                cout<<IMPOSSIBILE<<endl; //... è impossibile.
        else // a==0 ma b!=0 l'equazione è...
        {
            cout<<"\nEquazione di 1^ grado:"<<endl; //...di grado 1 e la soluzione è...
            x1=-c/b;
            cout<<"Soluzione: x= "<<x1<<endl; //... -c / b
        }
    else // a!=0, quindi l'equazione è...
    {
        cout<<"\nEquazione di 2^ grado:"<<endl; //...di secondo grado
    }
}

```

```

delta=pow(b,2)-4*a*c;
if (delta>0)                                // se il delta è positivo...
{
    x1=(-b+sqrt(delta))/2*a;                //... si hanno le due soluzioni distinta...
    x2=(-b-sqrt(delta))/2*a;
    cout<<"Soluzioni: x1= "<<x1<<endl;
    cout<<"Soluzioni: x2= "<<x2<<endl;
}
else
    if (delta<0)                            //...se è negativo sono immaginarie...
    {    delta=-delta;                       //... e la soluzione può essere stampata...
        partereale=-b/2*a;                 //...stampando separatamente la parte...
        parteimmaginaria=sqrt(delta)/2*a; //...reale da quella immaginaria
        cout<<"x1="<<partereale<<"+i*"<<parteimmaginaria<<endl;
        cout<<"x2="<<partereale<<"-i*"<<parteimmaginaria<<endl;
    }
    else                                    // ...se nullo sono coincidenti
    {
        x1=x2=-b/2*a;
        cout<<"Soluzioni: x1= "<<x1<<endl;
        cout<<"Soluzioni: x2= "<<x2<<endl;
    }
}
system("Pause");
return 0;
}

```

Esercizi proposti

7. Scrivere un programma che consenta di calcolare il M.C.D. tra due interi secondo la tecnica euclidea delle divisioni successive.
(Sugg. Il programma legge i due numeri da input e ripete il calcolo del resto fino a quando questo diventa zero. Ad ogni passo, il divisore diventa il resto precedente e il dividendo diventa il divisore precedente.)
8. Scrivere un programma che realizzi una calcolatrice aritmetica a ciclo continuo, funzionante nel seguente modo. Si immette il primo valore, si immette ripetutamente l'operatore (scelto fra cinque operatori aritmetici, quattro per le operazioni, uno per il simbolo di "=") e, se l'operatore immesso è diverso dall' "=", il secondo valore. Il calcolo termina quando l'utente immette il simbolo "=". Durante l'elaborazione, deve essere stampato il valore provvisorio del risultato del calcolo; al termine del programma, deve essere stampato il valore definitivo.
9. Scrivere un programma che stampi i primi n numeri della successione di Fibonacci, una successione numerica così definita:

$$f_0=0$$

$$f_1 = 1$$

$$f_2 = 1$$

$$f_3 = 2$$

$$f_4 = 3$$

...

$$f_n = f_{n-1} + f_{n-2}$$

Ad esempio, per $n= 6$ si ha

$$0 - 1 - 1 - 2 - 3 - 5$$

(E) ESERCITAZIONI PRATICHE

1. Scrivere un programma per verificare la sensazione del tempo da parte dell'utente.

Il programma fa uso dell'orologio interno del computer. Inizialmente, chiede all'utente di premere INVIO per avviare il conteggio del tempo. In questo istante, parte l'orologio. Successivamente, l'utente, quando ritiene siano passati ad esempio 5 secondi, preme di nuovo INVIO per fermare l'orologio. La differenza tra il tempo iniziale e quello in cui l'utente preme il tasto, dà il tempo trascorso e consente la verifica.

```
/* SensazioneTempo.cpp */
#include <iostream>
#include <ctime>
using namespace std;
int main()
{ long int tm;
  cout<<"Proviamo la tua sensazione del tempo\n"<<endl;
  cout<<"Quando sei pronto premi INVIO \n "<<endl;
  cout<<"Conta fino a 5 e poi premi di nuovo INVIO \n "<<endl;
  getchar();
  tm= time(0);
  getchar();
  if(time(0)-tm==5)
    cout<<"Bene!! Hai vinto!"<<endl;
  else
    cout<<"Sei fuori tempo! Riprova!";
  system("Pause");
  return 0;
}
```

OSSERVAZIONI:

- la variabile *tm* è dichiarata di tipo **long int** in quanto destinata a contenere il valore dell'ora iniziale; il tipo **int** non sarebbe sufficiente, come ampiezza a contenere il valore di *tm*;
- la funzione **time(0)** restituisce il valore corrente del timer interno al sistema. Questo valore, depositato in *tm*, consente, nella s.d.c. selettiva di valutare l'intervallo di tempo trascorso. La libreria **time.h** è indispensabile per poter usare la funzione **time()**.