Programmi Informatica

**/* Il programma seguente stampa il proprio nome e cognome normalmente e su una cornice.
Creato da Guido Cioni*/**

```
#include <stdio.h>
#include <stdlib.h>

main()
{
	printf("Guido\n");
	printf("Cioni \n");
	printf("\n");
	printf("*******\n*Guido*\n*******\n\n");
	system("PAUSE");
}
```

/*Scrivere il programma ordine_caratteri che chiede all'utente due caratteri, verifica se il primo carattere precede il secondo in ordine alfabetico e, in questo caso stampa il carattere '<', in caso contrario stampa la stringa '>='.

N.d.A. Ho modificato , non sostanzialmente, il programma! */

```
#include <stdlib.h>
#include <stdio.h>

main()
{
	char primo,secondo;
	printf("Immetti i caratteri in ordine\n\n ");
	scanf("%c %c",&primo ,&secondo);
	if(primo>secondo)
	printf("Il primo carattere '%c' segue il secondo '%c'" , primo , secondo);
	else printf("Il secondo carattere '%c' segue il primo '%c'\n\n" , secondo
, primo);

	system ("PAUSE");

}
```

**/*Scrivere un programma che dichiara due variabili reali (float) che rappresentano i lati di un rettangolo,
assegna a tali variabili due valori e stampa il perimetro e l'area del rettangolo risultante.
Il programma deve anche calcolare e stampare l'area di un quadrato avente lo stesso perimetro ed
il perimetro di un quadrato avente la stessa area.
Per calcolare la radice quadrata utilizzare la funzione sqrt(x) della libreria math.h */**

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>

int main()
{
    float
lato1=0,lato2=0,area=0,perimetro=0,latoquadrato=0,areaquadrato=0,perimetroquadra
to=0;
```

```
    printf("Immettere i lati del rettangolo \n\n Lato 1=");
    scanf("%f" , &lato1);
    printf("\n\n Lato 2=") ;
    scanf("%f" , &lato2);
    if(lato1>0 && lato2>0)
    {

     perimetro=lato1+lato2 ;
     area=lato1*lato2;
     printf("\n\nIl perimetro del rettangolo e' %f" , perimetro);
     printf("\n\nL'area del rettangolo e' %f\n\n",area);
     areaquadrato=(perimetro/4)*(perimetro/4);
     printf("L'area del quadrato equivalente e' %f\n\n" , areaquadrato);
     perimetroquadrato=4*(sqrt(area));
     printf("Il perimetro del quadrato equivalente e' %f\n\n" ,
perimetroquadrato);

     }

     else printf ("Mi dispiace , i dati inseriti non sono validi\n\n");

    system("PAUSE");

      }
```

/*Scrivere un programma che stampa la memoria allocate per I tipi studiati a lezione */

```
#include <stdlib.h>
#include <stdio.h>
main()
{
      printf("La memoria allocata al tipo short e' %d bit\n" , sizeof(short));
      printf("La memoria allocata al tipo int e' %d bit \n" , sizeof(int));
      printf("La memoria allocata al tipo long e' %d bit \n\n" , sizeof(long));
      printf("Possiamo facilmente verificare che vale \n\n
sizeof(short)<=sizeof(int)<=sizeof(long) \n\ninfatti %d<=%d<=%d \n\n " ,
sizeof(short) ,sizeof(int),sizeof(long));
      system("PAUSE");

}
```

/*Scrivere un programma che dichiara due variabili reali (float) che rappresentano i lati di un rettangolo,
assegna a tali variabili due valori e stampa il perimetro e l'area del rettangolo risultante.
Il programma deve anche calcolare e stampare l'area di un quadrato avente lo stesso perimetro ed
il perimetro di un quadrato avente la stessa area.
Per calcolare la radice quadrata utilizzare la funzione sqrt(x) della libreria math.h
A differenza del precedente questo programma verifica se i lati sono validi*/

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>

int main()
{
    float
lato1=0,lato2=0,area=0,perimetro=0,latoquadrato=0,areaquadrato=0,perimetroquadra
to=0;
    printf("Immettere i lati del rettangolo \n\n Lato 1=");
```

```
    scanf("%f" , &lato1);
    printf("\n\n Lato 2=") ;
    scanf("%f" , &lato2);
    if(lato1>0 && lato2>0)
    {

     perimetro=lato1+lato2 ;
     area=lato1*lato2;
     printf("\n\nIl perimetro del rettangolo e' %f" , perimetro);
     printf("\n\ nL'area del rettangolo e' %f\n\n",area);
     areaquadrato=(perimetro/4)*(perimetro/4);
     printf("L'area del quadrato equivalente e' %f\n\n" , areaquadrato);
     perimetroquadrato=4*(sqrt(area));
     printf("Il perimetro del quadrato equivalente e' %f\n\n" ,
perimetroquadrato);

     }

     else printf ("Mi dispiace , i dati inseriti non sono validi\n\n");

    system("PAUSE");

      }
```

/*Scrivere il programma cambia_formato_carattere che chiede all'utente un carattere e se questo è un carattere alfabetico minuscolo lo trasforma in maiuscolo, se invece è un carattere alfabetico maiuscolo lo trasforma in minuscolo. */

```
#include <stdlib.h>
#include <stdio.h>

main()
{
      char car,CAR;
      printf("Inserisci il carattere : ");
      scanf("%c",&car);
      if(car>='a'&&car<='z')
      {
        CAR=car-('a'-'A');

        /*La differenza 'a'-'A' è lo scarto fra la rappresentazione ASCI delle
        lettere maiuscole e minuscole dell'alfabeto*/

        printf("La lettera maiuscola per %c e' %c\n\n", car, CAR);

      }
      else if(car>='A'&&car<='Z')
      {
            CAR=car+('a'-'A');
            printf("La lettera minuscola per %c e' %c\n\n" , car , CAR);
      }

 system("PAUSE");
}
```

/*Scrivere il programma migliorare_la_media che chiede ad uno studente la somma dei suoi voti di profitto e il numero di esami sostenuti. Stampa la sua media aritmetica esatta ed il voto minimo che dovrebbe ottenere al prossimo esame perchè la media migliori raggiungendo la votazione (intera) immediatamente superiore alla media attuale. Se tale voto fosse maggiore di 30, il programma deve stampare il messaggio "Mi dispiace, non si puo'".

N.B.: si tenga conto che sia il numero degli esami che la votazione ottenuta nel singolo esame sono interi, sono votazioni valide per il superamento di un esame solo quelle comprese tra 18 e 30 (estremi inclusi) */

```
#include <stdio.h>
#include <stdlib.h>

main()
{
      int somma=0,esami=0,mediaint,voto;
      float media;
      printf("Inserisci la somma dei tuoi voti e il numero di esami sostenuti,
separati da uno spazio : ");
      scanf("%d %d" , &somma , &esami);
      media=(float)somma/esami;
      printf("\n\nLa tua media e' %f\n\n",media);
      mediaint=media;
      voto=(int)((2*mediaint)+2-media);
      if(voto<30&&mediaint>=18)
      printf("Per raggiunere la votazione di %d devi prendere : %d\n\n" ,
mediaint+1 , voto);
      else
      printf("Mi dispiace ma non e' possibile\n   Se sei ignorante non e' colpa
mia!\n");

 system ("PAUSE");
}
```

/* Scrivere il programma valore_monete che calcola e stampa il valore complessivo in vecchie lire della seguente serie di monete:
5 monete da due Euro;
8 monete da un Euro;
7 monete da mezzo euro (50 centesimi di Euro);
7 monete da 5 cent;
Per rendere leggibile il codice definire opportune costanti DUE_EURO, EURO, MEZZO_EURO e CINQUE_CENT.
Si ricordi che 1 Euro = 1936.27 lire italiane.

Il programma è stato modificato per permettere l'inserimento di una qualsiasi serie di monete*/

```
#include <stdlib.h>
#include <stdio.h>

int main()
{
    const float EURO=1936.27;
    float dueeuro,uneuro,cinquantacent,cinquecent,somma=0;
    printf("Digitare il numero di monete nell'ordine indicato dal programma
\n\n");
    printf("Monete da 2 Euro=");
    scanf("%f",&dueeuro);
    printf("\nMonete da 1 Euro=");
    scanf("%f",&uneuro);
    printf("\nMonete da 50 Cent=");
    scanf("%f",&cinquantacent);
    printf("\nMonete da 5 Cent=");
    scanf("%f",&cinquecent);
    somma=EURO*((dueeuro)*2+(uneuro)+cinquantacent*0.5+cinquecent*0.05);
    printf("La somma delle monete e' %f Lire\n\n",somma);

    system("PAUSE");
```

```
}
```

**/*Scrivere il programma tri_isoscele che chiede all'utente due interi che rappresentano
la base e l'altezza di un triangolo isoscele, li legge e stampa il perimetro e l'area
del triangolo rappresentato. Per calcolare la radice quadrata utilizzare la funzione sqrt(x)
della libreria math.h */**

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

main()
{
	int base=0,altezza=0;
	float area=0,perimetro=0,lato=0;
	printf("Immettere la base del triangolo :");
	scanf("%d",&base);
	printf("\n\nImmettere l'altezza del triangolo :");
	scanf("%d",&altezza);
	lato=sqrt(altezza^2+(base^2)/4);
	perimetro=(lato*2+base);
	area=base*altezza/2;
	printf("\n\nIl perimetro del triangolo e' : %f \nmentre l'area e' :%f ",
perimetro,area);

	system ("PAUSE");

}
```

**/*Scrivere il programma multiplo che legga un intero e
determini se è multiplo di 7, di 5 o di entrambi.*/**

```
#include <stdio.h>
#include <stdlib.h>

main()
{
	int in;
	int div1,div2;
	printf("Immetti l'intero :");
	scanf("%d",&in);
	div1=in%7;
	div2=in%5;
	if(div1==0&&div2==0) printf ("\n\nL'intero e' multiplo di 5 e 7\n");
	else if (div1==0) printf ("\n\nL'intero e'multiplo di 7\n");
	else if(div2==0) printf("\n\nL'intero e' multiplo di 5\n");
	else printf ("\n\nL'intero non e' multiplo di nessuno dei numeri\n");

	system ("PAUSE");

}
```

**/*Un anno bisestile è identificato da un intero maggiore di 1584
che sia divisibile per 4 ma non per 100 oppure che sia divisibile
per 400. Scrivere il programma bisestileche letto un anno determini
se tale anno è bisestile.*/**

```
#include <stdio.h>
#include <stdlib.h>
```

```
main()
{
	int anno;
	printf ("Immetti l'anno :");
	scanf("%d",&anno);
	if(anno>1584)

	{
	  if(anno%400==0||(anno%4==0&&anno%100!=0))
	  printf("L'anno e' bisestile\n\n");
	  else printf("L'anno non e' bisestile\n\n");

	}

	else printf("Mi dispiace, hai immesso un anno non valido\n\n");

	system("PAUSE");

}
```

**/*Scrivere il programma ordina_tre che legge tre interi
e li stampa in ordine decrescente facendo il minimo
numero di confronti possibile.**

**Il programma suppone che gli interi siano diversi
altrimenti si dovrebbe fare un altro controllo*/**

```
#include <stdio.h>
#include <stdlib.h>
main()
{
	int a,b,c;
	printf("Immetti 3 interi diversi tra loro (separati da uno spazio): ");
	scanf("%d%d%d",&a,&b,&c);
	if(a>b)

	{
		if(b>c) printf("%d%d%d",a,b,c);
		else

		{
			if(c>a) printf("%d%d%d",c,a,b);
			else printf("%d%d%d",a,c,b);

		}

	}

	else

	{
	     if(b<c) printf("%d%d%d",c,b,a);

	     else

	     {
	         if(c>a) printf("%d%d%d",b,c,a);
	         else printf("%d%d%d",b,a,c);

	     }

	}
```

```
	printf("\n");
	system("PAUSE");

}
```

**/*Scrivere il programma voto_2_giudizio che legge un valore intero v
che rappresenta un voto e a seconda del valore letto stampa il giudizio
corrispondente come specificato dalla tabella riportata sotto:
30 < v	"Valore illecito! "
24 < v <= 30	"Ottimo"
21 < v <= 24	"Buono"
18 <= v <= 21	"Sufficiente"
0 <= v < 18 "Insufficiente"
v < 0 "Valore illecito!" */**

```
#include <stdio.h>
#include <stdlib.h>

main()
{
	int voto;
	printf("Immetti il voto : ");
	scanf("%d",&voto);
	if(voto>30) printf("Valore illecito\n");
	else if(voto>=24) printf("Ottimo\n");
	else if(voto>=21) printf("Buono\n");
	else if(voto>=18) printf("Sufficiente\n");
	else if(voto>=0) printf("Insufficiente\n");
	else printf("Valore illecito\n");

	system("PAUSE");

}
```

**/*Scrivere il programma calcolatrice che legge un valore di tipo double,
uno di tipo char e poi ancora uno di tipo double e infine fa in modo che:
se il carattere letto è '+' allora stampa la somma dei due valori numerici;
se il carattere è '*' allora stampa il prodotto dei due valori numerici;
se è un altro carattere allora stampa il primo valore.*/**

```
#include <stdio.h>
#include <stdlib.h>

main()
{
	float primo,secondo;
	int rappr;
	char segno;
	printf("Immetti l'operazione :");
	scanf("%f%c%f",&primo,&segno,&secondo);
	if(segno=='+')
	printf("La somma dei due numeri e' %f \n", primo+secondo);
	else if(segno=='*')
	printf("Il prodotto dei due numeri e' %f \n",primo*secondo);
	else
	printf("%f",primo);

	system("PAUSE");

}
```

/*Scrivere il programma tipo_triangolo che legge tre valori interi

che rappresentano le lunghezze dei lati di un triangolo, stabilisce se sono dati attendibili (se la somma di due lati fosse minore dell'altro lato non potrebbero formare un triangolo) e, in caso affermativo, se si tratta di un triangolo scaleno, isoscele o equilatero, stampando un opportuno messaggio. */

```
#include <stdio.h>
#include <stdlib.h>

main()
{
      float primo,secondo,terzo;
      printf("Immetti le lunghezze dei lati (separate da uno spazio) :");
      scanf("%f%f%f",&primo,&secondo,&terzo);

      if(primo+secondo>terzo)

      {

        if (primo==secondo)

        {
          if(secondo==terzo) printf("Equilatero\n");
          else printf("Isoscele\n");

        }

        else

        {
            if(secondo==terzo) printf("Isoscele\n");
            else if(primo==terzo) printf("Isoscele\n");
            else printf("Scaleno\n");

      }

      }

      else printf("Mi dispiace ma non e' un triangolo");

      system("PAUSE");

}
```

```
/***    Il file contiene  una possibile
**    implementazione del gioco della morra
**    ed illustra l'uso dei comandi 'switch'
**    e if annidati
**
**    Variabili esportate:
**
**    Funzioni esportate:
**
**    Storia:
**
*/

#include<stdlib.h>
#include<stdio.h>
#include<time.h>

#define PC_SASSO  1
#define PC_FORBICI      2
```

```c
#define     PC_CARTA    3

#define     UM_SASSO    's'
#define UM_FORBICI      'f'
#define UM_CARTA  'c'

int main()
{
  int   pc_simbolo=0;
  char      um_simbolo='x';

  /* Per sapere cosa fanno le funzioni srand() e rand()
  ** usare il comando linux 'man srand'
  */
  srand(time(NULL));
  pc_simbolo=(rand()%3)+1;

  /* Da qui in poi il codice che usa il valore pseudo-casuale calcolato */


  printf("Immetti il tuo simbolo\n");
  scanf("%c", &um_simbolo);

  printf("Il tuo simbolo (%c) ",um_simbolo);

  switch (um_simbolo)
    {
    case UM_SASSO :
      if (pc_simbolo==PC_FORBICI) printf("batte il mio simbolo ");
      else if (pc_simbolo==PC_SASSO) printf("pareggia con il  mio simbolo ");
      else printf("perde con  il mio simbolo ");
      break;

    case UM_FORBICI:
      if (pc_simbolo==PC_CARTA) printf("batte il mio simbolo ");
      else if (pc_simbolo==PC_FORBICI) printf("pareggia con il  mio simbolo ");
      else printf("perde con  il mio simbolo ");
      break;

    case UM_CARTA:
      if (pc_simbolo==PC_SASSO) printf("batte il mio simbolo ");
      else if (pc_simbolo==PC_CARTA) printf("pareggia con il  mio simbolo ");
      else printf("perde con  il mio simbolo ");
      break;
    default:
      printf("non e` valido, il mio e` ");
    }

  switch (pc_simbolo)
    {
    case PC_SASSO:
      printf("s.\n");
      break;
    case PC_FORBICI:
      printf("f.\n");
      break;
    case PC_CARTA:
      printf("c.\n");
      break;
    }

  return(0);
}
```

```
/*Scrivere il programma tipo_bevanda che legge un valore in virgola mobile g
che rappresenta la gradazione alcolica di una bevanda e a seconda del valore
letto stampa il messaggio corrispondente come specificato dalla tabella
riportata sotto:
50 < g      "Superciuk!"
33.3 < g <= 50    "Superalcolico"
20 < g <= 33.3    "Alcolico"
15 < g <= 20      "Vino liquoroso"
12 < g <= 15      "Vino forte"
10.5 < g <= 12    "Vino normale"
0 < g <= 10.5     "Vino leggero"
g = 0 "Bevanda analcolica"  */

#include <stdio.h>
#include <stdlib.h>

main()
{
      int grado;
      printf("Immetti la gradazione alcolica : ");
      scanf("%d",&grado);
      if(grado>50) printf("Superciuk\n");
      else if(grado>33.3) printf("SuperAlcolico\n");
      else if(grado>20) printf("Alcolico\n");
      else if(grado>15) printf("Vino Liquoroso\n");
      else if(grado>12) printf("Vino forte\n");
      else if (grado>10.5) printf("Vino normale\n");
      else if (grado>0) printf("Vino leggero\n");
      else if (grado=0) printf("Bevanda analcolica\n");

      system("PAUSE");

}

/*Scrivere un programma che legge una sequenza di interi non negativi terminata
 da un numero negativo e stampa il corrispondente istogramma.
Esempi di esecuzione (input utente in blu):

 Inserire gli  interi non negativi:
 8 2 0 5 3 9 4 -1

8 |********
2 |**
0 |
5 |*****
3 |***
9 |*********
4 |****

*/

#include <stdio.h>
#include <stdlib.h>

main()
{
      int intero,boolean=1,i;
      printf("Inserire la sequenza terminata da un numero negativo\n");
      while(boolean)

      {
```

```
            scanf("%d*c",&intero);

            /* Il carattere *c serve per evitare che legga lo spazio*/

            if (intero<0) boolean=!boolean;

            /* termina il ciclo nel caso che il numero sia negativo*/

            else

            {
                printf("%d |",intero);

                for (i=0 ; i<=intero-1 ; i++)
                printf("*");

                printf("\n");    /*il ritorno a capo deve essere fuori dal ciclo*/

            }

        }

        system("PAUSE");

}
```

/*Scrivere un programma che, legge una sequenza di 0 e di 1 di dimensione prefissata K e stampa il numero intero la cui rappresentazione in complemento a 2 su K cifre è la sequenza letta.*/

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

main()
{
        int dimensione,variabile;
        float numero=0;
        int i,countdim=0;
        printf("Quanto e' lunga la sequenza? ");
        scanf ("%d",&dimensione);
        countdim=dimensione-1;
        printf("Inserire la sequenza");

        for(i=0 ; i<dimensione ; i++)

        {
            scanf("%d*c",&variabile);
            numero=numero+variabile*(pow(2,countdim));
            countdim=countdim-1;

        }

        printf("Il numero in decimale e':%\n",numero);

        system("PAUSE");

}
```

/*Scrivere un programma che chiede all'utente una sequenza di caratteri alfabetici minuscoli verificando che ogni carattere letto sia maggiore o uguale ai precedenti (secondo l'ordine alfabetico). Il primo carattere inserito può essere un qualsiasi carattere minuscolo.

La sequenza termina quando l'utente immette un carattere non alfabetico o maiuscolo oppure se immette un carattere minore di quello letto precedentemente. Terminata la lettura dei caratteri il programma deve stampare il numero di caratteri minuscoli diversi appartenenti alla sequenza (il carattere che causa la terminazione non è considerato parte della sequenza).
Se la sequenza è vuota, cioè non viene immesso alcun carattere minuscolo, allora il programma stampa solo un avvertimento.*/

```
#include <stdio.h>
#include <stdlib.h>

main()
{
  char carattere;
  char carattereprec;
  int count=0,boolean=1,control=1;

  while(boolean)

  {
     printf("Dammi un carattere : ");
     scanf("%c%*c",&carattere);

     if(control==1&&carattere<='z'&&carattere>='a')

     {
       count++;
       carattereprec=carattere;
       control=0;
     }

     else if(control==1&&carattere<='Z'&&carattere>='A')

     {
       printf("La sequenza di lettere minuscole e' vuota\n");
       control=0;
     }

     else if(carattere<='z'&&carattere>='a'&&carattere>carattereprec)
     {
       count++;
       carattereprec=carattere;

     }
     else if(carattere<='z'&&carattere>='a'&&carattere==carattereprec)
     {
       carattereprec=carattere;
     }
     else boolean=!boolean;

  }

  printf("Totale lettere minuscole ordinate e diverse: %d", count);

  system("PAUSE");

}
```

/*Scrivere un programma che chieda all'utente un intero n, e stampi un triangolo isoscele di asterischi, di altezza lunga n e base lunga 2n-1. (Se il valore letto è negativo si consideri il suo valore assoluto). */

```
#include <stdio.h>
```

```c
#include <stdlib.h>

main()
{
  int intero,count=1,indice,altezza,spazi,indicespazi;
  printf("Dammi il numero intero :");
  scanf("%d",&intero);


  for(altezza=1 ; altezza<=intero ; altezza++)

  {
    spazi=intero-altezza;
    for(indicespazi=1 ; indicespazi<=spazi ; indicespazi++)
    {
      printf(" ");
    }
    for(indice=1 ; indice<=count ; indice++)
    {
    printf ("*");
    }
    for(indicespazi=1 ; indicespazi<=spazi ; indicespazi++)
    {
      printf(" ");
    }
    printf("\n");
    count=count+2;

  }

  system ("PAUSE");

}
```

**/*Un anno bisestile è identificato da un intero maggiore di 1584
che sia divisibile per 4 ma non per 100 oppure che sia divisibile
per 400. Scrivere il programma bisestileche letto un anno determini
se tale anno è bisestile.**

**Con riferimento all'esercizio [7] del laboratorio 3, modificare il programma
bisestile in modo che continui a leggere valori finchè l'utente immette anni non
bisestili e che si fermi quando l'utente immette un anno bisestile.*/**

```c
#include <stdio.h>
#include <stdlib.h>

main()
{
      int anno,boolean=1;

      while(boolean)

      {
        printf ("Immetti l'anno :");
        scanf("%d",&anno);

       if(anno>1584)

       {
         if(anno%400==0||(anno%4==0&&anno%100!=0))
         {
         printf("Finalmente un anno bisestile!\n\n");
         boolean=0;
```

```
          }

         else printf("L'anno non e' bisestile\n\n");

         }


      else printf("Mi dispiace, hai immesso un anno non valido\n\n");

      }

      system("PAUSE");

}
```

/*Modificare il programma dell'esercizio [5] per stampare un triangolo isoscele, di altezza lunga n (con n compreso tra 0 e 9) e base lunga 2n-1 fatto come sotto. Se il valore letto è non è compreso tra 0 e 9 si chieda all'utente un nuovo valore finché l'intero immesso non appartenga all'intervallo richiesto.
Esempio di interazione con il programma (in blu l'input dell'utente):

Dammi un numero intero positivo compreso tra 0 e 9: -2
Dammi un numero intero positivo compreso tra 0 e 9: 5

```
     1
    212
   32123
  4321234
 543212345   */

#include <stdio.h>
#include <stdlib.h>

main()
{
  int intero,count=1,indice,altezza=1,spazi,indicespazi,k;
  k=altezza;
  printf("Dammi il numero intero :");
  scanf("%d",&intero);
  if(intero>9||intero<0)
  printf("Hai inserito un numero non valido");
  else
      {
       for(altezza=1 ; altezza<=intero ; altezza++)

        {
          spazi=intero-altezza;
          for(indicespazi=1 ; indicespazi<=spazi ; indicespazi++)
          {
            printf(" ");
          }
             while(k>1)
             {
               printf ("%d",k);
               k--;
             }
             while(k<=altezza)
             {
               printf("%d",k);
               k++;
             }
```

```
  for(indicespazi=1 ; indicespazi<=spazi ; indicespazi++)
    {
      printf(" ");
    }
    printf("\n");
    count=count+2;

  }
    }

  system ("PAUSE");

}
```

/*Scrivere un programma pos_in_seq.c che legge da tastiera N valori interi e stampa le posizioni di quelli maggiori di un ulteriore valore acquisito in input */

```
#include <stdio.h>
#include <stdlib.h>

main()
{
  int controllo,variabile,lung,i;
  printf("Inserire la lunghezza della lista : ");
  scanf("%d",&lung);
  printf("\nInserire l'intero da controllare : ");
  scanf("%d",&controllo);
  printf("\nInserire la lista di interi da controllare :");
  for(i=1; i<=lung ;i++)
  {
    scanf("%d",&variabile);
    if(variabile>controllo)
    printf("%d ",variabile);
  }

  system("PAUSE");
}
```

/*Scrivere un programma che crei un array di 19 caratteri forniti **dall'utente, chieda all'utente un intero n, e stampi su un'unica riga i caratteri memorizzati nelle posizioni divisibili per n. */**

```
#include <stdio.h>
#include <stdlib.h>
#define LUNG 19

main()
{
  char vet[LUNG] ;
  int intero,i;
  printf("Inserire l'intero : ");
  scanf("%d",&intero);

  for(i=0; i<LUNG ; i++)
  {
   printf("Inserisci l'elemento n %d :",i+1);
   scanf("%c%*",&vet[i]);
  }

  for(i=0; i<LUNG ; i++)
  {
   if((i+1)%intero==0) printf("%c ",vet[i]);
```

```
  }

  system("PAUSE");

}
```

/* Scrivere un programma che modifichi, usando i puntatori, un vettore di 7 interi azzerando tutti gli elementi in posizione pari. */

```
#include <stdio.h>
#include <stdlib.h>
#define LUNG 7

main()
{
  int vet[LUNG],i;
  for(i=0; i<LUNG ; i++)
  {
   printf("Inserisci l'elemento n %d :",i+1);
   scanf("%d",&vet[i]);
  }
  for(i=0; i<LUNG ; i++)
  {
    if((i+1)%2==0) vet[i]=0;
  }
  for(i=0; i<LUNG ; i++)
   printf("%d ",vet[i]);

   system("PAUSE");

}
```

/*Scrivere un programma che crei un array di 7 interi forniti dall'utente e stampi, usando i puntatori, la somma degli elementi in posizione dispari. */

```
#include <stdio.h>
#include <stdlib.h>
#define LUNG 7

main()
{
  int vet[LUNG] ;
  int *pi=vet;
  int somma=0,i;

  for(i=0; i<LUNG ; i++)
  {
   printf("Inserisci l'elemento n %d :",i+1);
   scanf("%d",&vet[i]);
  }

  for(i=0; i<LUNG ; i++)
  {
   if((i+1)%2!=0)
   somma=somma+*(pi+i);
  }

  printf("\nLa somma degli elementi in posizione dispari e' : %d\n",somma);

  system("PAUSE");

}
```

/*Scrivere un programma che dati due interi, ne scambi il valore usando i puntatori.*/

```
#include <stdio.h>
#include <stdlib.h>

main()
{
	int var1=0,var2=0,temp=0;
	int *pvar1 , *pvar2;
	printf("Immetti i valori\n");
	scanf("%d %d",&var1,&var2);
	pvar1=&var1;
	pvar2=&var2;

	printf("\n\nI valori prima dello scambio sono : var1=%d , var2=%d
\n\n",*pvar1,*pvar2);

	temp=*pvar1;
	*pvar1=*pvar2;
	*pvar2=temp;

	printf("\n\nI valori dopo lo scambio sono : var1=%d , var2=%d
\n\n",*pvar1,*pvar2);

	system("PAUSE");
}
```

/*Scrivere un programma che crei un array di 7 interi forniti dall'utente e controlli se l'array è ordinato.*/

```
#include <stdio.h>
#include <stdlib.h>
#define LUNG 7

main()
{
	int vet[LUNG],BOOLEAN=0,i,ordine,primo;

	for(i=0 ; i<LUNG ; i++)
	{
	  printf("Inserisci l'elemento numero %d:",i+1);
	  scanf("%d",&vet[i]);
	}

	printf("Che ordine vuoi controllare ?\n Premi 1 per crescente , 2 per
decrescente");
	scanf("%d",&ordine);

	if(ordine==1)

	{
	  for(i=1 ; i<LUNG ; i++)
	  {
	    if(vet[i]<vet[i-1]) BOOLEAN=1;
	  }

	}

	else if(ordine==2)

	{
	  for(i=1; i<LUNG ; i++)
```

```
        {
         if(vet[i]>vet[i-1]) BOOLEAN=1;
        }
      }

      if(BOOLEAN==1) printf("L'array non e'ordinato\n\n");
      else if(BOOLEAN==0) printf("L'array e' ordinato\n\n");

system("PAUSE");
}
```

/* Scrivere un programma che crei un array di 7 interi e determini (se c'e') la posizione del primo elemento che e' minore della somma degli elementi che lo precedono.*/

```
#include <stdio.h>
#include <stdlib.h>
#define LUNG 7

main()
{
      int vet[LUNG],somma,posizione,i,boolean=1;

      for(i=0 ; i<LUNG ; i++)
      {
        printf("Inserisci l'elemento numero %d:",i+1);
        scanf("%d",&vet[i]);
      }
      somma=vet[0];

        for(i=0; i<LUNG ; i++)
        {
          if(vet[i]>somma)
           somma=somma+vet[i];

          else if(vet[i]<somma)
          {
           printf("L'elemento cercato e' in posizione %d\n" , i);
           i=LUNG+1;
          }
        }
system("PAUSE");
}
```

**/* Creare una matrice di interi 5 (N) per 7 (M), assegnando a ciascun elemento il valore dell'espressione 10*n+m dove n ed m sono rispettivamente il valore del primo e del secondo indice della matrice relativi a quell'elemento.
Definire un puntatore ad interi che punti inizialmente all'elemento (0,0).
Scrivere un programma che faccia spostare il puntatore alla posizione (i,j) (dove i e j sono interi letti dall'input) e stampi il contenuto della cella corrispondente.
Eseguire il programma con valori di (i,j) uguali a (3,4), (5,7), (10,12).*/**

```
#include <stdio.h>
#include <stdlib.h>
#define RIGHE 5
#define COLONNE 7

main()
{
      int mat[RIGHE][COLONNE],i,j,*pmat;
      for(i=0; i<RIGHE; i++)
```

```
        for(j=0; j<COLONNE; j++)
          mat[i][j]=10*(i+j);
          printf("MATRICE CREATA\n");

        pmat=&mat[0][0];

        printf("inserire l'indice dell'elemento (i,j) da stampare ");
        scanf("%d %d", &i ,&j);
        pmat=&mat[i][j];
        printf("\n\nL'elemento (%d,%d) e' : %d \n\n", i,j,*pmat);

        system("PAUSE");

}
```

/* Scrivere un programma che crea un array di 7 interi e determina (se c'e') la posizione del primo elemento che e' minore della somma degli elementi che lo seguono.*/

```
#include <stdio.h>
#include <stdlib.h>
#define LUNG 7

main()
{
      int vet[LUNG],somma=0,i,j,boolean=1;

      for(i=0 ; i<LUNG ; i++)
      {
        printf("Inserisci l'elemento numero %d:",i+1);
        scanf("%d",&vet[i]);
      }
      for(i=0;i<LUNG;i++)
      {
      {
       for(j=i+1;j<LUNG; j++)
        somma=somma+vet[j];
      }
      if(vet[i]<somma)
      {
       printf("L'elemento voluto e' in posizione %d\n" , i+1);
       i=LUNG+1;
      }
      }

      system("PAUSE");

}
```

/*Scrivere un programma che crei un array di 5 caratteri forniti dall'utente, chieda all'utente un carattere c, e verifichi se il carattere c appare nell'array.*/

```
#include <stdio.h>
#include <stdlib.h>
#define LUNG 5

main()
{
  char vet[LUNG],carattere;
  int trovato=0,i;

      for(i=0 ; i<LUNG ; i++)
```

```
      {
        printf("Inserisci l'elemento numero %d:",i+1);
        scanf("%c%*c",&vet[i]);
      }

      printf("CARATTERI ACQUISITI \n Ora inserisci il carattere di controllo : "
);
      scanf("%c%*c",&carattere);

      for(i=0;i<LUNG;i++)
      {
        if(vet[i]==carattere)
         trovato=1;
      }

      if(trovato==1) printf("Il carattere e' presente nell'array\n");
      else printf("Il carattere non e' presente nell'array\n");

      system("PAUSE");
}
```

/*Scrivere un programma che crei due array di caratteri forniti dall'utente a, di 5 elementi, e b, di 3 elementi, e verifichi se gli array non sono vuoti ed esiste almeno un carattere dell'array a che segue (secondo l'ordinamento alfabetico) tutti i caratteri presenti nell'array b. */

```
#include <stdio.h>
#include <stdlib.h>
#define LUNGa 5
#define LUNGb 3

main()
{
      char veta[LUNGa],vetb[LUNGb];
      int i,BOOLEAN=1,j,trovato;

      printf("Acquisizione primo array di %d elementi\n" , LUNGa);
      for(i=0 ; i<LUNGa ; i++)
      {
        printf("Inserisci l'elemento numero %d:",i+1);
        scanf("%c%*c",&veta[i]);
      }
      printf("Acquisizione secondo array di %d elementi\n", LUNGb);
      for(i=0 ; i<LUNGb ; i++)
      {
        printf("Inserisci l'elemento numero %d:",i+1);
        scanf("%c%*c",&vetb[i]);
      }

      /*CONTROLLO ARRAY*/

      for(i=0;i<LUNGa;i++)
      {
        if(veta[i]<'z'&&veta[i]>'A') BOOLEAN=1;
        else
        {
            printf("\nMi dispiace ma hai immesso un carattere non valido\n");
            BOOLEAN=0;
            i=LUNGa+1;
        }
      }
      while(BOOLEAN)
      {
```

```
        for(i=0;i<LUNGa;i++)
         for(j=0;j<LUNGb;j++)
          if(veta[i]>vetb[j])
          {
           printf("\n\nHo trovato il carattere che cercavi\n");
           i=LUNGa+1;
           j=LUNGb+1;
           BOOLEAN=0;
          }
          else
          {
           printf("\nIl carattere non e' presente\n");
           BOOLEAN=0;
           i=LUNGa+1;
           j=LUNGb+1;
          }
      }
      system("PAUSE");

}
```

FUNZIONI

/*Scrivere una procedura che preso un array di caratteri modifichi il vettore in modo che ogni vocale venga sostituita dal simbolo '$' .*/

```
void VOCALI(char vet[] , int dim)
{
  int i;
  for(i=0;i<dim;i++)
  {
   if(vet[i]=='a'||vet[i]=='e'||vet[i]=='i'||vet[i]=='o'||vet[i]=='u')
    vet[i]='$';
  }
}
```

/*Scrivere una funzione che presi due interi m e n restituisca true se m e' multiplo di n o viceversa e false. */

```
int  DIVISORI(int m , int n)
{
     int BOOLEAN;
     if(m%n==0||n%m==0) BOOLEAN=1;
     else BOOLEAN=0;
     return BOOLEAN;
}
```

/*Scrivere una funzione che accetti in ingresso tre valori x,a e b e verifichi che x è compreso nell'intervallo [a,b]*/

```
int INTERVALLO(float x , float a , float b)
{
  int BOOLEAN;
  if(x<=b&&x>=a) BOOLEAN=1;
  else BOOLEAN=0;
  return BOOLEAN;
}
```

/*Scrivere una procedura che preso un array di caratteri lo inverta.*/

```
void INVERTICARATTERI(char vet[] , int dim)
{
  int i,max;
  char temp;
  if(dim%2!=0)
  {
   max=((dim-1)/2);
   for(i=0;i<max;i++)
   {
    temp=vet[i];
    vet[i]=vet[dim-i-1];
    vet[dim-i-1]=temp;
   }
  }
   else
   {
     max=(dim/2);
     for(i=0;i<max;i++)
    {
    temp=vet[i];
    vet[i]=vet[dim-i-1];
    vet[dim-i-1]=temp;
    }
  }
}
```

/*Scrivere una funzione che presi due interi m e n calcoli il loro massimo comun divisore.*/

```
int MCD(int primo , int secondo)
{
    int output;
    while(primo=!secondo)
     if(primo>secondo)
      primo=primo-secondo;
     else
      secondo=secondo-primo;
      output=secondo;
      return output;
}
```

/*Scrivere una funzione che preso un array di caratteri, verifichi se l'array è palindromo.*/

```
int PALINDROMO(char vet[] , int dim)
{
 /*ATTENZIONE
 DICHIARARE DIMENSIONE SOLO DA DEFINE */

  int i,trovato=0,max,output;
  if(dim%2!=0)
  {
   max=((dim-1)/2);
   for(i=0;i<max;i++)
   {
     if(vet[i]==vet[dim-i-1]) trovato=1;
     else trovato=0;
   }
  }
   else
   {
     max=(dim/2);
     for(i=0;i<max;i++)
```

```
    {
      if(vet[i]==vet[dim-i-1]) trovato=1;
      else trovato=0;
    }
   }
  return trovato;
}
```

/*Scrivere una funzione che preso un array di interi, restituisca la posizione dell'ultima occorrenza di un elemento pari nell' array.*/

```
int PARIVETTORE(int *vet,int dim)
{
    int i,posizione;
    for(i=0;i<dim;i++)
    {
      if(vet[i]%2==0) posizione=i+1;
    }
    return posizione;
}
```

/* Scrivere delle funzioni per verificare le seguenti proprietà sui vettori:

- **tutti gli elementi sono maggiori di un dato elemento el,**
- **il vettore è ordinato in senso descrescente,**
- **ogni elemento pari è seguito da almeno un elemento dispari */**

```
int PROPVETT1(int vet[] , int dim, int elemento )
{
      int i,somma=0;
      for(i=0;i<dim;i++)
       if(vet[i]>elemento) somma++;
      if(somma==dim) return 1;
      else return 0;
}

int PROPVETT2(int vet[] , int dim)
{
      int i,BOOLEAN=1;
      for(i=0 ; i<dim ; i++)
        {
          if(vet[i]<vet[i-1]) BOOLEAN=1;
              else
              {
               BOOLEAN=0;
               i=dim+1;
            }
        }
       return(BOOLEAN);
}

int PROPVETT3(int vet[] , int dim)
{
      int i,j,BOOLEAN=1,somma,controllo;
      for(i=0 ; i<dim ; i++)
      {
        if(vet[i]%2==0)
      {
        somma=0;
        for(j=i+1;j<dim;j++)
        {
```

```
            if(vet[j]%2!=0) somma++;
        }

        if(somma=!0) controllo++;
      }
    }

    if (controllo==dim) return 1;
    else return 0;
}
```

Procedure Input-Output Vettori

```
void LEGGIVETTORECARATTERI(char vet[] ,int dim)
{
     int i;
     for(i=0;i<dim;i++)
     {
       printf("Immettere l'elemento di indice %d: " , i+1);
       scanf("%c%*c",&vet[i]);
     }
}

void LEGGIVETTOREINTERI(int vet[] ,int dim)
{
     int i;
     for(i=0;i<dim;i++)
     {
       printf("Immettere l'elemento di indice %d: " , i+1);
       scanf("%d",&vet[i]);
     }
}

void STAMPAVETTORECARATTERI(char vet[],int dim)
{
       int i;
       for(i=0;i<dim;i++)
        printf("%c ",vet[i]);
     printf("\n");
}

void STAMPAVETTOREINTERI(int vet[],int dim)
{
       int i;
       for(i=0;i<dim;i++)
        printf("%d ",vet[i]);
     printf("\n");
}
```

Funzioni Ricorsive

/* Scrivere una funzione serie che abbia come argomento un intero non negativo e restituisca un valore calcolato come segue:

serie(n) = 1 se n = 0

serie(n) = (n - (serie(n -1) * n)) se n > 0

Scrivere un programma tabulaserie.c che chiede all'utilizzatore un range di interi non negativi e stampa i valori della funzione per tutti gli argomenti contenuti nel range.
(Aiuto: la funzione chiede all'utilizzatore il minimo e il massimo del range)*/

```
/*PROTOTIPO FUNZIONI*/

float SERIE(int intero)
{
    float ris;
    if(intero==0) ris=1;
    else /* si suppone che l'intero sia positivo ( il controllo viene fatto nel
main)*/
      ris=(intero-(SERIE(intero-1)*intero));
      return ris;
}

/*PROTOTIPO FUNZIONI*/

#include<stdio.h>
#include<stdlib.h>

main()
{
  int min,max,i;
  float ris;
  printf("Immetti il range di interi non negativi:");
  scanf("%d %d",&min,&max);printf("\n\n");
  if(min<0||max<0) printf("Ti avevo detto non negativi, IDIOTA!");
  else
  {
   for(i=min;i<=max;i++)
   {
     ris=SERIE(i);
     printf("Serie(%d)=%f\n",i,SERIE(i));
   }
  }

  system("PAUSE");
}
```

/*Scrivere una funzione ricorsiva palindroma che legge una sequenza di caratteri di dimensione ignota, con un punto centrale, e verifica se tale sequenza (esclusi gli spazi bianchi) è palindroma.*/

```
#include<stdio.h>
#include<stdlib.h>
#include <string.h>

main()
{
  int ris;
  int dim;
  char vet[dim];

  int PALINDROMO(char *PC,char *UC);

  printf("Inserisci la stringa da controllare\n");
  scanf("%s",vet);
  dim=strlen(vet);
```

```
  ris=PALINDROMO(&vet[0],&vet[dim-1]);
  if(ris==1)
   printf("La stringa e' palindroma\n");
  else
   printf("La stringa non e' palindroma\n");

  system("PAUSE");
}

/*PROTOTIPO FUNZIONI*/

int PALINDROMO(char *PC, char *UC)

{
    /* Se la stringa e' vuota , e' costituita da un solo carattere o
    abbiamo sorpassato la metà allora è sicuramente palindroma*/

      if(PC>=UC) return 1;

    /*Se il primo e l'ultimo carattere sono diversi allora non e' palindroma*/

      else if(*PC!=*UC) return 0;

   /*Se invece i caratteri sono uguali dobbiamo richiamare ricorsivamente la
   funzione sui caratteri successivi a quelli appena analizzati*/

      else
      return PALINDROMO(PC+1,UC-1);
}

/*PROTOTIPO FUNZIONI*/
```

TIPI USER DEFINED

/*Definire un nuovo tipo di dato capace di rappresentare i dipendenti di una ditta. Di tali dipendenti interessa il cognome, il numero degli anni di anzianità maturati e lo stipendio. Si supponga che la dimensione massima del cognome sia di 40 caratteri. Scrivere poi delle opportune funzioni/procedure:
-aggiorna_stipendio che, dato un dipendente, aumenti del 1% il suo stipendio per ogni anno di anzianità accumulato (attenzione l'interesse da calcolare e' un interesse composto).
-precede che dati due impiegati a e b verifichino che il cognome del dipendente a preceda il cognome del dipendente b in ordine alfabetico.
Scrivere un programma che richiede all'utente di inserire i dati di due impiegati, utilizza le due procedure sopra e ne stampa l'esito.*/

```
 /* DICHIARAZIONE TIPI */

struct dipendente
        {
        char cognome[10];
        int anni;
        float stipendio;
        };

 /*DICHIARAZIONE TIPI*/

 /*PROGRAMMA PRINCIPALE*/
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>


typedef struct dipendente dip;

 main()
 {
  dip dipendente1,dipendentea,dipendenteb,*dip1,*dipa,*dipb;
  dip1=&dipendente1;
  dipa=&dipendentea;
  dipb=&dipendenteb;
  int ris;

  void AGGIORNASTIPENDIO(dip *pdip);
  int PRECEDE(dip *pdipa, dip *pdipb);


  printf("Inserisci cognome dipendente : ");
  scanf("%s",&(dip1->cognome));
  printf("\nInserisci anni di anzianità : ");
  scanf("%d",&(dip1->anni));
  printf("\nInserisci stipendio annuale : ");
  scanf("%f",&(dip1->stipendio));
  AGGIORNASTIPENDIO(dip1);
  printf("\nLo stipendio maturato e' : %f" , dip1->stipendio);
  printf("\nInserisci cognome dipendente a : ");
  scanf("%s",&(dipa->cognome));
  printf("\nInserisci cognome dipendente b : ");
  scanf("%s",&(dipb->cognome));
  ris=PRECEDE(dipa,dipb);
  if (ris==1) printf("Il cognome del dipendente a :%s precede alfabeticamente
quello del dipendente b %s\n",dipa->cognome,dipb->cognome);
  else printf("Mi dispiace, la proprietà non e' verificata\n");


  system("PAUSE");

}


 /*PROGRAMMA PRINCIPALE*/

 /*FUNZIONI E PROCEDURE*/

 void AGGIORNASTIPENDIO(dip *pdip)
 {
  float incremento;
  int i;
  for(i=0;i<=(pdip->anni);i++)
  {
   incremento=(pdip->stipendio)*(0.01/100);
   pdip->stipendio=pdip->stipendio+incremento;
  }
}

int PRECEDE(dip *pdipa , dip *pdipb)
{
  int ris;
  ris=strcmp(pdipa->cognome,pdipb->cognome);
  if (ris<0) return 1;
  else return 0;
```

```
}
```

/*FUNZIONI E PROCEDURE*/

/* Definire un nuovo tipo di dato capace di rappresentare una data.
Scrivere poi delle opportune funzioni/procedure che
-data una data la aggiorni al giorno successivo (ignorando gli anni bisestili),
-dati due date verifichino che la prima preceda la seconda.*/

```
 /* DICHIARAZIONE TIPI */

struct data {
       int giorno;
       int mese;
       int anno;
       } ;


 /*DICHIARAZIONE TIPI*/

 /*PROGRAMMA PRINCIPALE*/

#include <stdio.h>
#include <stdlib.h>

typedef struct data DATA;


 main()
 {
  DATA input,*p; /*ATTENZIONE, BISOGNA INIZIALIZZARE IL PUNTATORE!!!*/
  DATA input1,input2,*p1,*p2;
  p1=&input1;
  p2=&input2;
  p=&input;
  int ris;

  void aggiornadata(DATA *input);

  printf("Immetti il giorno (gg):");
  scanf("%d",&(p->giorno));
  printf("\nImmetti il mese (mm):");
  scanf("%d",&(p->mese));
  printf("\nImmetti l'anno (aaaa): ");
  scanf("%d",&(p->anno));
  aggiornadata(p);
  printf("La data aggiornata e' %d/%d/%d\n\n\n",(p->giorno),(p->mese),(p-
>anno));
  printf("Inserisci ora due date da confronare\n");
  printf("Immetti il giorno (gg):");
  scanf("%d",&(p1->giorno));
  printf("\nImmetti il mese (mm):");
  scanf("%d",&(p1->mese));
  printf("\nImmetti l'anno (aaaa): ");
  scanf("%d",&(p1->anno));
  printf("Prima data acquisita\n");
  printf("Immetti il giorno (gg):");
  scanf("%d",&(p2->giorno));
  printf("\nImmetti il mese (mm):");
  scanf("%d",&(p2->mese));
  printf("\nImmetti l'anno (aaaa): ");
```

```
  scanf("%d",&(p2->anno));
  printf("Seconda data acquisita\n");
  ris=confrontadata(p1,p2);
  if (ris==1) printf("La prima data %d/%d/%d precede la seconda %d/%d/%d
\n",(p1->giorno),(p1->mese),(p1->anno),(p2->giorno),(p2->mese),(p2->anno));
  else printf("Mi dispiace ma le due date non sono ordinate\n");

 system("PAUSE");

 }


 /*PROGRAMMA PRINCIPALE*/

 /*FUNZIONI E PROCEDURE*/

void aggiornadata(DATA *input)
 {
  if((input->giorno)<=29&&(input->giorno)>=1)
  {
    if((input->giorno)==28&&(input->mese)==02)
    {
     (input->mese)++;
     (input->giorno)=1;
    }
  else (input->giorno)++;
 }

 else if((input->giorno)==30)
 {
   if((input->mese)==11||(input->mese)==4||(input->mese)==6||(input->mese)==9)
   {
     (input->giorno)=1;
     (input->mese)++;
   }
   else (input->giorno)++;
 }

 else if((input->giorno)==31)
 {
   if((input->mese)!=12)
   {
    (input->giorno)=1;
    (input->mese)++;
   }
   else
   {
     (input->giorno)=1;
     (input->mese)=01;
     (input->anno)++;
   }
 }
}

int confrontadata(DATA *input1, DATA *input2)
{
 if ((input1->anno)==(input2->anno))
 {
  if((input1->mese)==(input2->mese))
  {
   if((input1->giorno)<(input2->giorno)) return 1;
   else return 0;
  }
```

```
  else if ((input1->mese)<(input2->mese)) return 1;
  else return 0;
 }
 else if((input1->anno)<(input2->anno)) return 1;
 else return 0;
}



 /*FUNZIONI E PROCEDURE*/
```

Procedure su Liste

```
/*Inizializza il primo elemento della lista al valore NULL*/

/*ATTENZIONE , sono necessarie le seguenti dichiarazioni nel MAIN:
             -tipo ListaDiElementi

             struct EL{
                    int info;
                    struct EL*next;
                    };
                    typedef struct EL ElementoLista;
                    typedef ElementoLista *ListaDiElementi;

             -costante NULL (includere stdlib.h)

             */

 void Inizializza(ListaDiElementi *lista)
 {
      *lista=NULL;
 }

/* Stampa la lista passata per VALORE
   Da notare che questa procedura NON MODIFICA LA LISTA */

void stampalista(ListaDiElementi lis)
{
  while(lis!=NULL)
  {
    printf("%d-->" ,lis->info);
    lis=lis->next;
  }
  printf("//\n");
}

/*Stampa la lista passata per valore ( versione ricorsiva ) */

void stampalistaricorsiva(ListaDiElementi lis)
{
     if(lis!=NULL)
     {
      printf("%d",lis->info);
      stampalistaricorsiva(lis->next);
     }
     else printf("//\n");
}
```

```
/*Inserisce un nuovo elemento in testa alla lista */

void inseriscitestalista(ListaDiElementi *lis, int el)
{
     ListaDiElementi aux;
     aux=malloc(sizeof(ElementoLista));
     aux->info=el;
     aux->next=*lis;
     *lis=aux;
}

/* Cancella il primo elemento della lista */

void cancellaprimo(ListaDiElementi *lista)
{
     ListaDiElementi *aux;
     if(*lista!=NULL)
     {
       aux=*lista;
       *lista=(*lista)->next;
       free(aux);
     }
}

/*Cancella un'intera lista invocando ricorsivamente cancellaprimo */

void cancellalista(ListaDiElementi *lista)
{
     while(*lis!=NULL)
      cancellaprimo(lista);
}

/* Verifica se un elemento appartiene alla lista */
/*Passaggio per indirizzo */

boolean appartiene(ElementoLista el , ListaDiElementi lista)
{
  boolean trovato=false;

  while(lista!=NULL&&!trovato)
  {
    if(lista->info==el)
     trovato=true;
    else
     lista=lista->next;
    return trovato;
 }

}

/* Verifica se un elemento appartiene alla lista */
/*Passaggio per indirizzo */

boolean appartienericorsiva(ElementoLista el , ListaDiElementi lista)
{
 if(lista==NULL)
  return false;
 else if(lista->info==el)
  return true;
 else
  return (appartienericorsiva(el,lis->next));
}
```

```
/*Inserisce un elemento in coda*/

void inseriscicoda(ListaDiElementi *lista,ElementoLista el)
{
 ListaDiElementi ultimo;
 ListaDiElementi aux;

 aux=malloc(sizeof(ElementoLista));
 aux->info=el;
 aux->next=NULL;

 if(*lista=NULL)
  *lista=aux;
 else
 {
  ultimo=*lista;
  while(ultimo->next!=NULL)
   ultimo=ultimo->next;
  ultimo->next=aux;
}
}

/*Inserisce un elemento in coda*/
/* Versione ricorsiva*/

void inseriscicodaricorsiva(ListaDiElementi *lista,ElementoLista el)
{
 if(*lista=NULL)
 {
  *lista=malloc(sizeof(ElementoLista));
  (*lista)->info=el;
  (*lista)->next=NULL;
 }
 else
 inseriscicodaricorsiva(&((*lista)->next),el);
}

/*Cancellazione della prima occorrenza di un elemento*/
/*Versione ricorsiva*/

void CancellaElementoLista(ListaDiElementi *lista, ElementoLista el)
{
 ListaDiElementi prec,corr;
 boolean trovato;

 if(*lista!=NULL)
  if((*lista)->info==el)
  {
   cancellaprimo(lista);
  }
 else
CancellaElementoLista(&((*lista)->next),el);
}

/*Cancellazione di tutte le occorrenze di un elemento*/
/*Versione ricorsiva*/

void CancellaTuttiLista(ListaDiElementi *lista, ElementoLista el)
{
 ListaDiElementi aux;
 if(*lista!=NULL)
  if((*lista)->info==el)
   {
```

```
    cancellaprimo(lista);
    CancellaTuttiLista(lista,el);
   }
  else
   CancellaTuttiLista(&(*lista->next),el);
}

/*Inserimento di un elemento in lista ordinata*/

void inserzioneordinata(ListaDiElementi *lista,ElementoLista el)
{
 if(*lista=NULL)
  InserisciTestaLista(lista,elem);
 else
  if((*lista)->info>=el)
   InserisciTestaLista(lista,el);
  else
   inserzioneordinata(&((*lista)->next),el);
}

/*Controllo lista vuota*/

boolean controllolistavuota(ListaDiElementi lista)
{
 if(lista==NULL) return true;
 else return false;
}
```