

**LABORATORIO DI
FONDAMENTI DI INFORMATICA**

**DIAGRAMMI DI FLUSSO E
CODIFICA IN C DEGLI ALGORITMI**

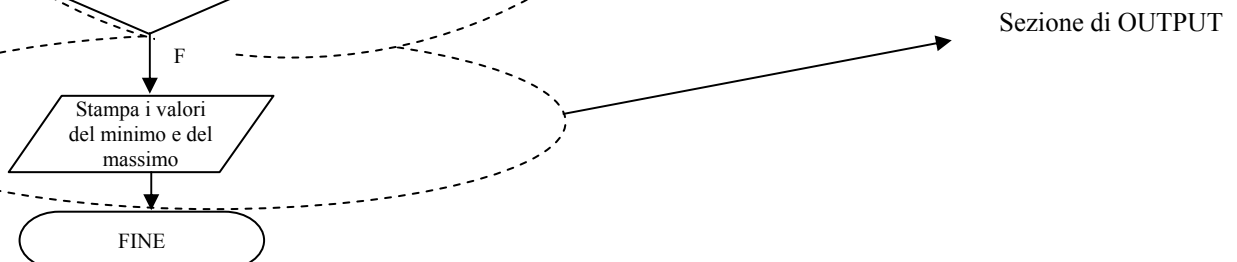
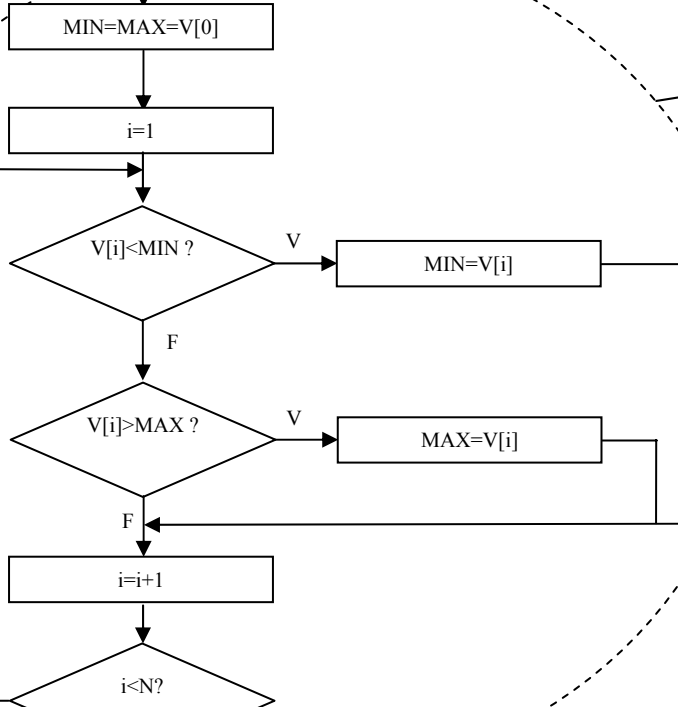
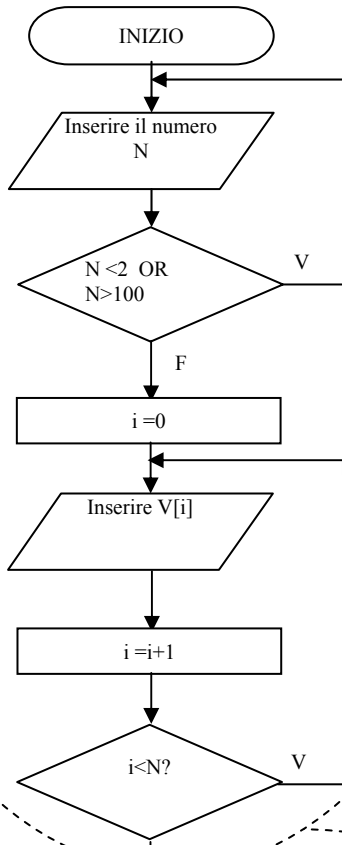
**1.INGEGNERIA ELETTRICA A.A. 2004-005
COPIA CONTROLLATA N 1**

ING. G. GIORDANO

Acquisiti N **numeri interi** (con N minore o uguale di $MX=100$) nel vettore V , determinare il massimo ed il minimo. In figura è stato rappresentato il diagramma di flusso relativo all'algorithmo risolutivo, mentre nelle pagine seguenti la codifica in linguaggio C.

N.B.

Nella codifica in C il codice è ottimizzato, considerando che l'acquisizione degli elementi del vettore successivi al primo, può essere fatta contestualmente alla ricerca del valore del massimo e del minimo, effettuata sia con il do-while sia con il for.



```

/* RICERCA DEL MINIMO E DEL MASSIMO UTILIZZANDO IL COSTRUTTO DO-WHILE */
#include <stdio.h>
#include <conio.h>
#define DIM_MAX 100
void main ()
{
    int i,N,MIN,MAX,V[DIM_MAX];
    char c;
do
    {
do
        /* occorrono almeno due elementi perché abbia senso la ricerca
del minimo e del massimo */
        {
            printf("\n");
printf(" N = ");
            scanf("%d",&N);
        }
while (N<2 || N>DIM_MAX);
printf("V[0]=");
scanf("%d", &V[0]); /*inserisco solo il primo elem. del vettore*/
MIN=MAX=V[0];
i=1;
do
    {
printf("V[%d]=",i);
scanf("%d", &V[i]);/*inserisco gli altri elem. del vettore*/
if (V[i] < MIN)
MIN = V[i];
else
if (V[i] > MAX)
MAX = V[i];
i = i+1;
    }
while (i<N);
/* output su monitor */
printf (" Il massimo vale %d\n", MAX);
printf (" Il minimo vale %d\n", MIN);
/* opzione */
printf(" Si vuole rieseguire l'algoritmo ? s/n ");}
        while (c= getch() != '\n');
    }
}

```

```

/* RICERCA DEL MINIMO E DEL MASSIMO UTILIZZANDO IL CICLO FOR */
#include <stdio.h>
#include <conio.h>
#define DIM_MAX 100
void main ()
{
    int i,N,MIN,MAX,V[DIM_MAX];
    char c;
do
    {
        do
            {
                printf("\n");
                printf(" N = ");
                scanf("%d",&N);
            }
        while (N<2 || N>DIM_MAX);
        printf("V[0]=");
        scanf("%d", &V[0]); /*inserisco solo il primo elem. del vettore*/
        MIN=MAX=V[0];
        for (i=1;i<N;i++)
            {
                printf("V[%d]=",i);
                scanf("%d", &V[i]);/*inserisco gli altri elem. del vettore*/
                if (V[i] < MIN)
                    MIN = V[i];
                else
                    if (V[i] > MAX)
                        MAX = V[i];
            }
        printf (" Il massimo vale %d\n", MAX);
        printf (" Il minimo vale %d\n", MIN);
        printf(" Si vuole rieseguire l'algoritmo ? s/n ");}
        while (c= getch() != '\n');
}

```

```
/* CALCOLO DELLA MEDIA DI UNA SEQUENZA DI VALORI INTRODOTTI IN UN VETTORE */
```

```
#include <stdio.h>
#include <conio.h>
#define DIM_MAX 100
void main ()
{
    int i,N,V[DIM_MAX],SOMMA;
    float MEDIA;
    char c;

do
{
do
    {
        printf("\n");
        printf(" N = ");
        scanf("%d",&N);
    }
while (N<2 || N>DIM_MAX);
SOMMA = 0;
for (i=0;i<N;i++)
{
printf("V[%d]=",i);
scanf("%d", &V[i]);/*inserisco gli altri elem. del vettore*/
SOMMA = SOMMA + V[i];
}
MEDIA = (float)SOMMA/N;/* cast della variabile somma */
printf (" La media vale %f\n", MEDIA);
/* le precedenti due righe di codice possono essere sostituite dalla seguente :
printf (" La media vale %f\n", (float)SOMMA/N); */

printf("\n");
printf(" Si vuole rieseguire l'algoritmo ? s/n ");}
    while (c= getch() != 'n' );

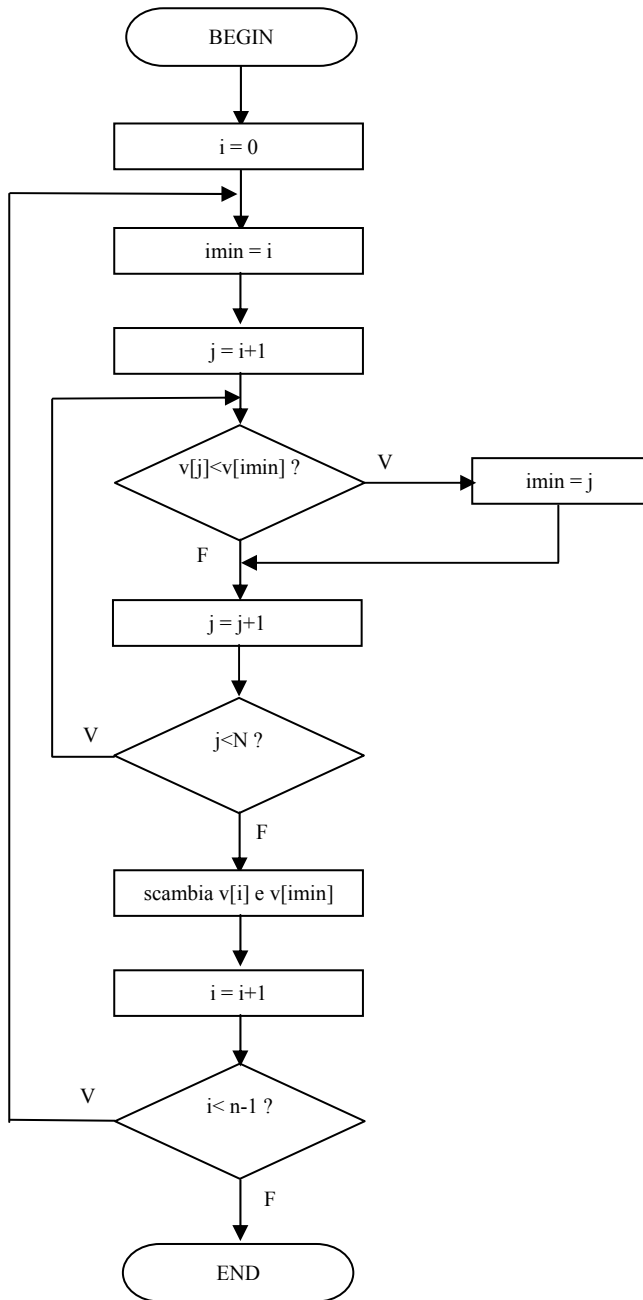
/* output su stampante standard < stdprn >
se si aggiunge la seguente riga di codice la stampa avviene su stampante
fprintf(stdprn, "media = %f\n", MEDIA); */
}
```

```
/* ALGORITMO DEL FATTORIALE */

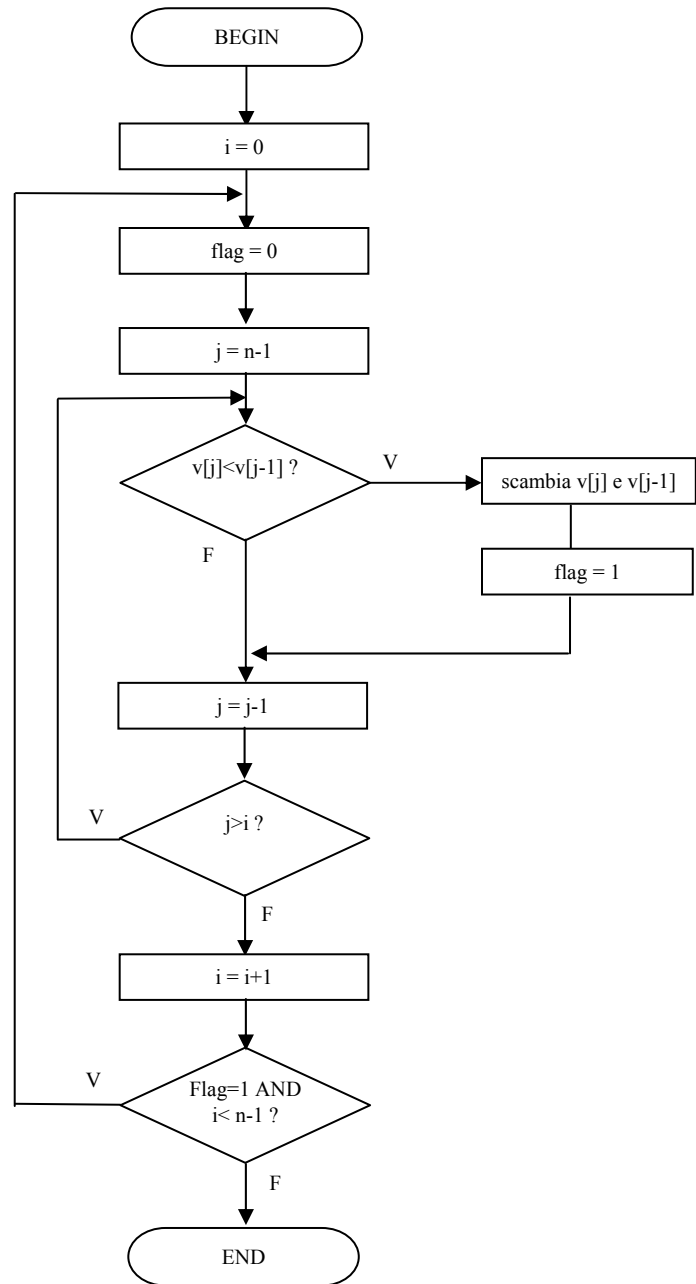
#include <stdio.h>
#include <conio.h>

void main()
{
    int N,AUX;
    float FAT;
    do
    {
        do
        {
            printf("Inserire il numero N non negativo di cui si vuole il fattoriale : ");
            scanf("%d",&N);
        }
        while(N < 0);
        FAT=1;
        for (AUX=1;AUX<=N;AUX++)
            FAT=FAT*AUX;
        printf(" Il fattoriale di %d vale %.0f \n", N, FAT);
        printf(" Vuoi rieseguire il programma? s/n \n");
    }
    while(getch()!='n');
}
```

Ordinamento con il metodo di selezione



Ordinamento con il metodo Bubble Sort



```
/* ORDINAMENTO DI UN VETTORE */
```

```
#include <stdio.h>
#include <conio.h>
#define DIM_MAX 100
void main()
{
    int N,i,imin,j,temp,V[DIM_MAX];
    do
    {
        do
        {
            printf("Immettere dimensione del vettore N= : ");
            scanf("%d",&N);
        }
        while(!(N >= 2 && N <= 100));
        printf(" Bravo, ci devono essere almeno due elementi \n");
    for (i=0;i<N;i++)
        {
            printf(" Introduci V[%d] = ", i);
            scanf("%d",&V[i]);
        }
    for (i=0;i<N-1;i++)
    {
        imin=i;
        for (j=i+1;j<N;j++)
            if (V[j]<V[imin])
                imin=j;
        temp=V[i];
        V[i]=V[imin];
        V[imin] = temp;
    }
    printf("il vettore ordinato: \n");
    for (i=0;i<N;i++)
        printf(" V[%d] = %d \n", i, V[i]);
    printf(" Vuoi rieseguire il programma? s/n \n");
    }
    while(getch()!='n');
}
```



```
/* ORDINAMENTO PER FUSIONE DI DUE VETTORI ORDINATI */
```

```
#include <stdio.h>
#include <conio.h>
#define DIM_MAX_V 100
#define DIM_MAX_W 50
#define DIM_MAX_Z DIM_MAX_V+DIM_MAX_W
void main()
{
    int N,M,i,j,k,V[DIM_MAX_V], W[DIM_MAX_W], Z[DIM_MAX_Z];
    do
    {
        do
        {
            printf("Immettere dimensione del vettore V : ");
            scanf("%d",&N);
        }
        while(!(N >= 2 && N <= DIM_MAX_V));
        for (i=0;i<N;i++)
        {
            printf(" Introduci V[%d] = ", i);
            scanf("%d",&V[i]);
        }
        do
        {
            printf("Immettere dimensione del vettore W : ");
            scanf("%d",&M);
        }
        while(!(M >= 2 && M <= DIM_MAX_W));
        for (i=0;i<M;i++)
        {
            printf(" Introduci W[%d] = ", i);
            scanf("%d",&W[i]);
        }
        /* fusione dei due vettori ordinati */
        i=0; j=0; k=0;
        do
        {
            if (V[i]<W[j])
                Z[k++]=V[i++];
            else
```

```
if (V[i]==W[j])
{
Z[k++]=V[i++];
j++;
}
else
Z[k++]=W[j++];
}
while(i<N && j<M);
if(i<N)
for(;i<N;Z[k++]=V[i++])
;
else
for(;j<M;Z[k++]=W[j++])
;
for (i=0;i<k;i++)
printf(" Z[%d] = %d \n", i, Z[i]);
printf(" Vuoi rieseguire il programma? s/n \n");
}
while(getch()!='n');
```

}

```
/* CONVERSIONE DA BASE 16 A BASE 10*/
```

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
void main()
{
    char codice[80],cifra[80];
    int i,j,k;
    do
    {
        float risultato = 0;
        printf(" \n Inserisci il codice esadecimale: ");
        scanf("%s",codice);
        for(i=0; ((codice[i]) != '\0'&& (codice[i]) != '.'); i++)
        {
            if(codice[i]>=48&&codice[i]<=57)
                cifra[i]= codice[i]-48;
            else
                if (codice[i]>=65&&codice[i]<=70)
                    cifra[i]=codice[i]-55;
        }
        j=i;
        i=i+1;
        for(; (codice[i]) != '\0'; i++)
        {
            if(codice[i]>=48&&codice[i]<=57)
                cifra[i-1]= codice[i]-48;
            else
                if (codice[i]>=65&&codice[i]<=70)
                    cifra[i-1]=codice[i]-55;
        }
        for(k=0;k<i-1;k++)
            risultato= risultato+(cifra[k]*pow(16,(j-k-1)));
        printf(" valore decimale = %f",risultato);
        printf(" \n vuoi rieseguire ?");
    }
    while(getch() != 'n');
```

/* INTRODUZIONE AI PUNTATORI */

```
int a=5; /* dichiarazione di una variabile intera e contemporanea
inizializzazione*/
int *p; /* dichiarazione di una variabile di tipo puntatore ad intero */
/* in questo caso il "simbolo" * ha solo valore sintattico in quanto specifica
che tipo di variabile si sta dichiarando: il puntatore è p */
p=&a; /* il puntatore p è inizializzato con l'indirizzo della variabile a */
/* a questo punto a e *p costituiscono il cosiddetto alias entrambi valgono 5 */
```

```
Es: printf(" la varibile intera a che si trova all'indirizzo %p vale %d ",p,*p);
```

/* Esempio di funzioni definite dall'utente : parametri passati per indirizzo */

```
#include <stdio.h>
void scambia(int *, int *);
/* dichiarazione: gli argomenti sono di tipo puntatore */

void main()
{
    int a,b;
    printf("a e b valgono rispettivamente\n");
    scanf("%d %d",&a,&b);
    scambia(&a,&b);
    /* la funzione scambia è chiamata passandole gli indirizzi */
    printf("a=%d e b=%d\n",a,b);
}

void scambia(int *px, int *py)
```

/* visto che px e py sono variabili formali di tipo puntatore deputate a contenere indirizzi possono contenere &a &b e quindi si può effettuare lo scambio lavorando sui valori reali di a e di b e non sulle loro copie per mezzo dell'operatore unario di dereferenziazione. In questo caso, infatti, px e py contengono una copia degli indirizzi di a e b ma *px e *py sono gli originali e non le copie di a e b */

```
{
    int temp;
    temp = *px;
    *px = *py;
    *py = temp;
}
```

```
/* la funzione calcolaMedia non restituisce risultati, essi sono memorizzati  
nelle variabili di cui è passato l'indirizzo */
```

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

```
void calcolaMedie (int, int, float *, float *);
```

```
void main ()
```

```
{  
    int x, y;  
    float ma, mg; /* media aritmetica e media geometrica */  
    do  
    {  
        printf ("immetti due interi: \n");  
        scanf ("%d%d", &x, &y);  
        calcolaMedie (x, y, &ma, &mg);  
        printf ("media aritmetica = %.2f\nmedia geometrica = %.2f\n", ma, mg);  
        printf("rieseguire?");  
    }  
    while(getch()!='n');  
}
```

```
void calcolaMedie (int a, int b, float *puntMarit, float *puntMgeom)
```

```
{  
    *puntMarit = (a + b)/2.0; /* 2.0 per avere divisione frazionaria  
                             * e non divisione intera */  
    if (a*b < 0)  
        *puntMgeom = -1;  
    else  
        *puntMgeom = sqrt (a * b);  
}
```