

## Alcuni elementi di programmazione in linguaggio C

- Dichiarazione di una variabile intera:

```
int n;
```

- Inizializzazione di una variabile intera:

```
n=10;
```

- Le due precedenti istruzioni si possono condensare in:

```
int n=10;
```

- Ciclo **for** (iterazione determinata) in Linguaggio C:

```
for (Exp1;Exp2;Exp3){  
    blocco di istruzioni;  
}
```

1. Si valuta Exp1;

2. Se Exp2 è vera:

2.1 si esegue il blocco di istruzioni;

2.2 si esegue Exp3;

2.3 torna al punto 2.

3. Se Exp2 è falsa, il ciclo termina.

- Ciclo **for** (iterazione determinata) nella versione più usuale:

```
int i;  
for (i=inizio; i<fine;i=i+passo){  
    blocco di istruzioni;  
}
```

dove *inizio*, *fine* e *passo* sono costanti intere.

- Ciclo **while**:

```
while (B){  
    blocco di istruzioni;  
}
```

- Istruzione **printf**:

```
printf("Buongiorno!"); //stampa la stringa "Buongiorno!"  
  
printf("Il valore di i e' %d",i);  
//al posto di %d verrà stampato il valore attuale della variabile i;
```

- Istruzione **scanf**:

```
scanf("%d",&i);  
//se i è una variabile intera precedentemente dichiarata, l'istruzione  
//assegna ad essa un valore intero che l'utente deve immettere da  
//tastiera.
```

- Scambio del valore di due variabili:

```
#include<stdio.h>  
int main(void) {  
  
    int a=23;  
    int b=49;  
  
    printf("La variabile a vale  %d\n",a);  
    printf("La variabile b vale  %d\n\n",b);  
  
    int x; //variabile di appoggio  
    x=a;  
    a=b;  
    b=x;  
    printf("La variabile a vale  %d\n",a);  
    printf("La variabile b vale  %d\n\n",b);  
}
```

- Stampa del valore di un numero a due cifre con le sue cifre invertite

```
#include<stdio.h>  
int main(void) {  
    //invertire un numero a due cifre;  
    int n=56;  
    printf("\nIl numero inserito è %d\n",n);  
    printf("\nIl numero invertito è %d%d\n\n",n%10,n/10);  
}
```

- Calcolo del numero di banconote e monete da 20, 10, 5, 2 e 1 Euro per coprire un importo

```
#include<stdio.h>
/*numero di banconote da 5, 10, 20 euro e
monete da 1 e 2 euro necessarie per pagare un importo.
*/
int main(void){
    //inserire un importo intero da pagare.
    int n=98;

    int banc20=n/20;
    n=n%20;

    int banc10=n/10;
    n=n%10;

    int banc5=n/5;
    n=n%5;

    int mon2=n/2;
    n=n%2;

    int mon1=n;

    printf("\nbanconote da venti: %d\n",banc20);
    printf("\nbanconote da dieci: %d\n",banc10);
    printf("\nbanconote da cinque: %d\n",banc5);
    printf("\nmonete da due: %d\n",mon2);
    printf("\nmonete da uno: %d\n\n",mon1);

}
```

- Somma dei primi n numeri naturali

```
#include<stdio.h>

int main(void){

    int n=20;
    int s=0;

    int i;
    for (i=1;i<=n;i=i+1)
        s=s+i;

    printf("\nLa somma dei primi %d numeri e': %d\n\n",n,s);

}
```

- Calcolo dell'MCD fra due interi

```
#include<stdio.h>

int main(void) {

    int a=30;
    int b=90;
    int r=a%b;

    while(r!=0) {
        a=b;
        b=r;
        r=a%b;
    }
    printf("\nMCD=%d\n\n",b);
}
```

- Calcolare e stampare la più piccola potenza di due maggiore di un numero immesso dall'utente

```
#include<stdio.h>

/*minore potenza di due maggiore di un numero
scelto dall'utente
*/

int main(void) {
    //inserire un numero.
    int n;
    scanf("%d",&n);

    int i=1;
    while(i<=n) {
        i=i*2;
        // printf("%d\n",i); serve per visualizzare le potenze
    }

    printf("La potenza richiesta eè %d\n\n",i);
}
```

- Calcolare e stampare il più piccolo quadrato perfetto maggiore di un numero scelto dall'utente

```
#include<stdio.h>

/*minore quadrato maggiore di un numero
scelto dall'utente
*/

int main(void){

    printf("inserire un numero: ");
    int n;
    scanf("%d",&n);

    printf("\n\n");

    int i=0;
    while(i*i<=n)
        i=i+1;

    printf("Il più piccolo quadrato maggiore di %d e' %d\n\n",n,
i*i);
}
```

- Stampare i quadrati perfetti minori o uguali di un numero scelto dall'utente

```
#include<stdio.h>
//quadrati minori di un numero immesso dall'utente

int main(void){
    printf("Immettere un numero intero positivo: ");
    int n;
    scanf("%d",&n);

    int i;
    for(i=1;i*i<n;i=i+1)
        printf("\n\n%d\t",i*i);

    return 0;
}
```

- Dichiarazione di un vettore di interi:

```
int v[n];
//viene dichiarato un vettore di interi di nome v di lunghezza n.

#include<stdio.h>

int main(void) {
// DICHIARAZIONE VETTORI

    int a[5] = {1,2,3,4,3};
//ogni componente è una variabile intera

    a[4]=90;
//stampo con un ciclo
    int i;
    for(i=0;i<=4;i=i+1)
        printf("%d ",a[i]);
    printf("\n\n");

//Ma non e' comodo. Quindi:
    int n;
    printf("\n\ndimensione vettore: ");
    scanf("%d",&n);

//inizializzazione con ciclo su scanf

    for(i=0;i<n;i=i+1)
        scanf("%d",&a[i]);

    for(i=0;i<n;i=i+1)
        printf("%d ",a[i]);
    printf("\n\n");
}
```

- Inizializzazione e stampa di un vettore tramite l'uso di due specifiche funzioni (inizializza e stampa). Calcolo della somma delle sue componenti tramite l'uso di una funzione (somma\_comp).

```
#include<stdio.h>

//PROTOTIPI
void inizializza(int [],int);
void stampa(int [],int);
int somma_comp(int [], int);

int main(void){
    int n;
    printf("\n\ndimensione vettore: ");
    scanf("%d",&n);
    int a[n];

//chiamata a inizializza

    inizializza(a,n);

    stampa(a,n);

    int somma=somma_comp(a,n);
    printf("La somma delle componenti e': %d\n\n",somma);
}

//Definizione funzioni

void inizializza(int v[],int N){

    int i;
    for(i=0;i<=N-1;i=i+1)
        scanf("%d",&v[i]);
    return;
}

void stampa(int v[],int N){

    int i;
    for(i=0;i<=N-1;i=i+1)
        printf("%d ",v[i]);

    printf("\n");
    return;
}

int somma_comp(int v[], int N){
```

```

    int s=0;
    int i;
    for (i=0;i<N;i=i+1)
        s=s+v[i];

    return s;
}

```

- Simulazione di una pila tramite un vettore. Si usano le funzioni PUSH e POP per inserimento ed estrazione, rispettivamente. La funzione TOP legge chi sarà il prossimo estratto. La funzione stampa\_pila visualizza la pila.

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

```
//PROTOTIPI
void PUSH(int [],int,int);
int POP(int [],int);
int TOP(int [], int);
void stampa_pila(int []);
```

```
int top=0;
```

```
int main(void) {
    int n;
    printf("\ndimensione massima pila: ");
    scanf("%d",&n);
    int a[n];
    printf("\n\n\n");

    int r; //variabile per estrazione

    PUSH(a,n,543);
    PUSH(a,n,145);
    PUSH(a,n,83);
    stampa_pila(a);

    r=POP(a,n);
    if(r!=-999)
        printf("Impossibile estrarre: Pila vuota\n");
    else
        printf("elemento estratto: %d\n\n",r);

    stampa_pila(a);
}
```



```

}

void PUSH(int v[], int n, int x){

    if(top==n-1){
        printf("Impossibile inserire: pila piena\n\n");
        return;
    }

    v[top]=x;
    top=top+1;
    return;
}

int POP(int v[], int n){

    if(top==0){
//        printf("Impossibile estrarre: pila vuota\n\n");
        return -999;
    }

    int estratto=v[top-1];
    top=top-1;
    return estratto;
}

int TOP(int v[], int n){

    if(top==0){
        printf("Impossibile estrarre: pila vuota\n\n");
        return -999;
    }

    return v[top-1];
}

void stampa_pila(int v[]){
    if(top==0){
        printf("pila vuota");
        return;
    }
    int i;
    for(i=top-1;i>=0;i=i-1)
        printf("%d\n",v[i]);
}

```

- Altri esempi di semplici programmi

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

int main(void){

    //numero di cifre di un numero

    int n;
    printf("Inerire un numero intero positivo: n=");
    scanf("%d",&n);

    int num_cifre=0;

    while(n!=0){
        n=n/10;
        num_cifre=num_cifre+1;
    }

    printf("Il numero di cifre e' %d",num_cifre);
}
```

---

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

int main(void){

    // somma delle cifre di un numero

    int n;
    printf("Inserire un intero positivo: n=");
    scanf("%d",&n);

    int somma_cifre=0;

    while(n!=0){
        somma_cifre=somma_cifre+n%10;
        n=n/10;
        printf("siamo qui");
    }

    printf("La somma delle cifree': %d",somma_cifre);
}
```

---

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

```

int main(void){

//somma componenti di posto pari e posto dispari di un vettore;
//notare come una componente di un vettore possa essere trattata
//come un variabile semplice.

    int n;
    printf("Immettere dimensione del vettore: n=");
    scanf("%d",&n);
    int v[n]; //si dichiara il vettore.

//con un ciclo for si inizializza il vettore:
    int i;
    for(i=0;i<n;i=i+1)
        scanf("%d",&v[i]);

    int x=0,y=0;

    for(i=0;i<n;i=i+2)
        x=x+v[i];

    for(i=1;i<n;i=i+2)
        y=y+v[i];

    printf("somma componenti di posto pari: %d\n",x);
    printf("Somma componenti di posto dispari: %d\n",y);
}

```

---

```

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

```

```

int main(void){

// numero di punti fissi, eccedenze e deficienze di un vettore

    int n;
    printf("Immettere dimensione del vettore: n=");
    scanf("%d",&n);
    int v[n];
    int i;
    int pf=0,def=0,ecc=0;

    for(i=0;i<n;i=i+1)
        scanf("%d",&v[i]);

    for(i=0;i<=n-1;i=i+1){
        if(v[i]==i+1)
            pf=pf+1;
        if(v[i]<i+1)
            def=def+1;
        if(v[i]>i+1)

```

```

        ecc=ecc+1;
    }

    printf("Ci sono %d punti fissi\n",pf);
    printf("Ci sono %d deficienze\n",def);
    printf("Ci sono %d eccedenze\n",ecc);
}
-----

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

int main(void){

// Massimi relativi di un vettore. Una componente di un vettore intero
// è un massimo relativo se è maggiore di quella successiva e
// di quella precedente.

    int n;
    printf("Immettere dimensione del vettore: n=");
    scanf("%d",&n);
    int v[n];
    int i;
    int max_rel=0;

    for(i=0;i<n;i=i+1)
        scanf("%d",&v[i]);

    for(i=1;i<=n-2;i=i+1){
        if(v[i]>v[i+1]&&v[i]>v[i-1])
            max_rel=max_rel+1;
    }

    if(max_rel>0)
        printf("Ci sono %d massimi relativi\n",max_rel);
    else
        printf("Non ci sono massimi relativi");
}

```