

Sequenze di controllo C:

- ✓ sequenza
- ✓ if-else
- ✓ while
- ✓ do-while
- ✓ for
- ✓ break
- ✓ switch

Sequenza

Una sequenza di istruzioni è racchiusa sempre tra parentesi graffe.

Per esempio: leggere e sommare due numeri

```
# include <stdio.h>

main() {
    int A,B,somma;
        scanf ("%d", &A) ;
        scanf ("%d", &B) ;
        somma = A + B;
        printf("%d", somma) ;
    }
```

if-else

Questa istruzione serve a indicare quale azione intraprendere sulla base del valore di una certa condizione.

La forma generale dell'istruzione **if** è la seguente:

```
if (espressione)
    istruzione_espr_vera;
else
    istruzione_espr_falsa;
```

dove *istruzione_espr_vera* e *istruzione_espr_falsa* possono essere istruzioni singole, sequenze di istruzioni o nulle.

La clausola *else* può essere opzionale.

Visualizzazione del valore assoluto della differenza

```
# include <stdio.h>

main() {
    int A,B,VA,differenza;

    scanf("%d",& A);
    scanf("%d",& B);
    differenza = A - B;

    if (differenza > 0)
        VA = differenza;
    else
        VA = -differenza;

    printf("%d",VA);
}
```

Massimo tra due numeri

```
# include <stdio.h>

main() {
int a,b, max;

    scanf("%d",a);
    scanf("%d",b);
    max = a;
    if (b > a)
        max = b;

    printf("%d",max);
}
```

Tre operazioni

```
# include <stdio.h>
main() {
int a,b, op, risultato;

    printf("Inserire il primo operando");
    scanf("%d",&a);
    printf("Inserire il secondo operando");
    scanf("%d",&b);
    printf("Quale operazione? (+,-,*,/)");
    scanf("%d",&op);
    if (op == '+')
        risultato = a + b;
    else if (op == '-')
        risultato = a - b;
    else risultato = a * b;
    printf("%d",risultato);
}
```

while

Questa istruzione permette la ripetizione di un'azione fino a quando una certa condizione è vera.

Non appena la condizione diventa falsa, viene eseguita l'istruzione successiva al while.

La forma generale dell'istruzione while è la seguente:

```
while (condizione)
    istruzione;
```

dove *istruzione* può essere un'istruzione vuota, una singola istruzione o un blocco di istruzioni.

L'istruzione eseguita deve essere tale che dopo un numero finito di volte la condizione diventi falsa.

Leggere e sommare 10 numeri

```
# include <stdio.h>

main(){
    conta = 0;
    somma = 0;

    while (conta < 10)
    { printf("Inserisci un numero");
      scanf("%d",&numero);
      somma = somma + numero;
      contatore = contatore + 1;
    }
    printf("%d",somma);
}
```

Visualizzare il massimo tra 20 numeri

```
# include <stdio.h>
main() {
    conta = 1;
    scanf("%d", &a);
    max=a;

    while(conta<20)
    {
        printf("Inserisci un numero");
        scanf("%d", &a);
        if(a>max)
            max=a;
        conta=conta+1;
    }
    printf("%d", max);
}
```

Calcolare la somma di una sequenza di numeri chiusa da uno zero

```
# include <stdio.h>
main()
{
    int numero, somma;
    printf("Inserire il primo numero");
    scanf("%d", &numero);
    somma=numero;
    while(numero!=0)
    {
        printf("Inserire un altro numero");
        scanf ("%d", &numero);
        somma=somma+numero;
    }
    printf("La somma è : %d", somma);
}
```

do-while

Questa istruzione permette la ripetizione di un'azione fino a quando una certa condizione è vera.

Non appena la condizione diventa falsa, viene eseguita l'istruzione successiva al do-while. La forma generale dell'istruzione do-while è la seguente:

```
do
    istruzione;
while (condizione);
```

dove *istruzione* può essere un'istruzione vuota, una singola istruzione o un blocco di istruzioni.

Mentre nel ciclo while la condizione viene valutata prima dell'esecuzione dell'istruzione, nel ciclo do-while la condizione viene valutata dopo l'esecuzione dell'istruzione.

Leggere e sommare n numeri

```
# include <stdio.h>
main() {
    int conta, somma, n;

    scanf("%d", &n);
    conta = 0;
    somma = 0;

    do
    {
        scanf("%d", &numero);
        somma = somma + numero;
        conta = conta + 1;
    }
    while (contatore < n);

    printf("%d", somma);
}
```

Visualizzare il minimo tra 30 numeri

```
# include <stdio.h>
main() {
    int conta,min,a;
    conta = 0;
    scanf("%d",&a);
    min=a;

    do
    {
        printf("Inserisci un numero");
        scanf("%d",&a);
        if(a<min)
            min=a;
        conta=conta+1;
    }
    while(conta<30);
    printf("%d",min);
}
```

Massimo di una sequenza chiusa dallo zero

```
# include <stdio.h>
main() {
    int conta,max,a;

    scanf("%d", &a);
    max=a;
    if(a!=0)
    {
        do
        { printf("Inserisci un numero");
          scanf("%d", &a);
          if(a>max && a!=0)
              max=a;
        } while(a!=0);
        printf("%d",max);
    }
    else
        printf("Il primo numero è uno zero");
}
```

for

Questa istruzione permette di eseguire un'*istruzione* fino a quando la *condizione* è vera. Viene usato normalmente per i cicli di cui si conosce il numero di ripetizioni.

La forma generale dell'istruzione for è la seguente:

```
for (inizializzazione; condizione; incremento)
istruzione;
```

L'inizializzazione normalmente è una istruzione di assegnamento di una variabile di controllo usata nella condizione.

Nella sezione incremento viene definito come deve variare la variabile di controllo ad ogni ciclo.

Leggere e sommare 20 numeri

```
# include <stdio.h>
main() {
    int indice, somma;
        somma = 0;

    for (indice = 0; indice < 20; indice++)
    {
        printf("Inserire un numero");
        scanf("%d", &numero);
        somma = somma + numero;
    }
    printf("%d", somma);
}
```


Leggere gli elementi di un vettore e stamparli in ordine inverso

```
# include <stdio.h>
main() {
    int indice;
    int A[10];

    for(indice=0;indice<10;indice++)
        {printf(" Inserisci un numero ");
         scanf("%d",&A[indice]);
        }

    for(indice=9;indice>0;indice--)
        printf("%d",A[indice]);
}
```

Leggere gli elementi di un vettore A di 10 elementi e visualizza l'indice del minimo

```
# include <stdio.h>
main() {
    int A[10];
    int I,imin;

    for(i=0;i<10;i++)
        {printf(" Inserisci un numero ");
         scanf("%d",&A[i]);
        }

    imin=0;
    for(i=1;i<10;i++)
        if(A[i]<A[imin])
            imin=i;

    printf("%d",imin);
}
```

Scambia gli elementi A[j] e A[k] di un vettore di 10 elementi

```
#include <stdio.h>
main() {
int A[10], I, j;
for(i=0; i<10; i++)
scanf("%d", &A[i]);

for(i=0; i<10; i++)
printf("%d", A[i]);

printf("Inserisci il primo indice ");
scanf("%d", &i);
printf("Inserisci il secondo indice");
scanf("%d", &j);

temp=A[i];
A[i]=A[j];
A[j]=temp;

for(i=0; i<10; i++)
printf("%d", A[i]);
}
```

Leggere e invertire gli elementi di un vettore A di 10 elementi

```
#include <stdio.h>
main() {
int A[10], i, inf, sup, temp;
for(i=0; i<10; i++)
scanf("%d", &A[i]);
for(i=0; i<10; i++)
printf("%d", A[i]);
inf=0;
sup=9;
while(inf<sup)
{temp=A[inf];
A[inf]=A[sup];
A[sup]=temp;
inf++;
sup--;}
for(i=0; i<10; i++)
printf("%d", A[i]);
}
```

break

- Questa istruzione permette l'interruzione dell'esecuzione dell'istruzione corrente.
- Tipicamente viene usata per interrompere l'esecuzione di un ciclo in corrispondenza di una certa condizione.
- Es.

```
for (i=0;i<10;i++)  
    if (A[i]<0) break;  
    else printf("%d",A[i]);
```

switch

- Questa istruzione è utilizzata in sostituzione dell'istruzione if-else, quando bisogna valutare più di due valori di una espressione e per ciascun valore una istruzione deve essere eseguita

```
switch(espressione) {  
    case val1:    istruzione1;  
                 break;  
    case val2:    istruzione2;  
                 break;  
    ...  
    default:     istruzioneN;  
}
```

Realizzazione delle 4 operazioni (+,-,*,/)

```
#include <stdio.h>
main(){
int a,b,resultato;
char op;
printf("Inserire il primo operando");
scanf("%d",&a);
printf("Inserire il secondo operando");
scanf("%d",&b);
printf("Quale operazione? (+,-,*,/)");
fflush(stdin);
scanf("%c",&op);
switch (op) {
case '+': risultato = a + b;
break;
case '-': risultato = a - b;
break;
case '*': risultato = a * b;
break;
default: if (b!=0) risultato = a / b;
}
if (op== '/' && b==0) printf("Operazione errata");
else printf("%d",resultato);
}
```