

## 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

```
{  
    scanf(A);  
    scanf(B);  
    somma = A + B;  
    printf(somma);  
}
```

## if-else

Prof. Giuseppe Ascia

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_if;
else
    istruzione_else;
```

dove *istruzione\_if* e *istruzione\_else* possono essere istruzioni singole, sequenze di istruzioni o nulle.

La clausola *else* può essere opzionale.

Fondamenti di Informatica

3

## Visualizzare il valore assoluto della differenza tra due numeri

Prof. Giuseppe Ascia

```
{
    scanf(A);
    scanf(B);
    differenza = A - B;

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

    printf(VA);
}
```

Fondamenti di Informatica

4

## Visualizzazione del massimo tra due numeri

Prof. Giuseppe Ascia

```
{
    scanf(a);
    scanf(b);
    max = a;
    if (b > a)
        max = b;

    printf(max);
}
```

Fondamenti di Informatica

5

## Realizzare una delle 3 operazioni (+,-,\*)

Prof. Giuseppe Ascia

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

Fondamenti di Informatica

6

## while

Prof. Giuseppe Ascia

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.

Fondamenti di Informatica

7

## Leggere e sommare 10 numeri

Prof. Giuseppe Ascia

```
{
    conta = 0;
    somma = 0;

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

Fondamenti di Informatica

8

## Leggere 10 numeri da tastiera

---

Prof. Giuseppe Ascia

```
{
  contatore=0;
  while (contatore <10)
  {
    scanf(numero);
    contatore = contatore +1;
  }
}
```

Fondamenti di Informatica

9

## Leggere e sommare i primi N numeri naturali

---

Prof. Giuseppe Ascia

```
{
  i=0;
  somma=0;

  scanf(N);
  while (i <N)
  {
    somma= somma+ i;
    i = i +1;
  }

  printf(somma);
}
```

Fondamenti di Informatica

10

## Visualizzare il massimo tra 20 numeri

Prof. Giuseppe Ascia

```
{
    conta = 1;
    scanf(a);
    max=a;

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

Fondamenti di Informatica

11

## Visualizzare il minimo comune multiplo tra due numeri positivi

Prof. Giuseppe Ascia

```
{
    printf("Inserisci il valore di a (>0)");
    scanf(a);
    printf("Inserisci il valore di b (>0)");
    scanf(b);
    mcma=a;
    mcmb=b;

    while(a!=b)
        if(a<b)
            mcma=mcma+a;
        else
            mcmb=mcmb+b;

    printf(mcma);
}
```

Fondamenti di Informatica

12

## Calcolare il minimo di una sequenza di numeri positivi chiusa da un numero non positivo

Prof. Giuseppe Ascia

```
{
printf("Inserire il primo numero");
scanf(numero);
if( numero>0)
{ minimo=numero;
while(numero>=0)
{
printf("Inserire un altro numero");
scanf(numero);
if(numero > 0 && numero < minimo)
minimo=numero;
}
printf(minimo);
}
else
printf("Il primo numero non e' positivo");
}
```

Fondamenti di Informatica

13

## do-while

Prof. Giuseppe Ascia

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.

Fondamenti di Informatica

14

## Leggere e sommare 100 numeri

Prof. Giuseppe Ascia

```
{
    conta = 0;
    somma = 0;
    do
    {
        scanf(numero);
        somma = somma + numero;
        conta = conta + 1;
    }
    while (contatore < 100);
    printf(somma);
}
```

Fondamenti di Informatica

15

## Visualizzare il minimo tra 30 numeri

Prof. Giuseppe Ascia

```
{
    conta = 0;
    scanf(a);
    min=a;
    do
    {
        printf("Inserisci un numero");
        scanf(a);
        if(a<min)
            min=a;
        conta=conta+1;
    }
    while(conta<30);
    printf(min);
}
```

Fondamenti di Informatica

16



## Visualizzare il massimo di una sequenza chiusa dallo zero

Prof. Giuseppe Ascia

```
{
    scanf(a);
    max=a;
    if(a!=0)
        do
        {
            printf("Inserisci un numero");
            scanf(a);
            if(a>max && a!=0)
                max=a;
        } while(a!=0);
    else
        printf("Errore");
    printf(max);
}
```

Fondamenti di Informatica

17

## for

Prof. Giuseppe Ascia

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.

Fondamenti di Informatica

18

## Leggere e sommare 20 numeri

Prof. Giuseppe Ascia

```
{
    somma = 0;

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

Fondamenti di Informatica

19

## Leggere gli elementi di un vettore di 10 elementi e visualizzarli in ordine inverso

Prof. Giuseppe Ascia

```
{
    for(indice=0;indice<10;indice++)
        scanf(A[indice]);

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

Fondamenti di Informatica

20

## Leggere gli elementi di un vettore A di 10 elementi e visualizzare il valore minimo

Prof. Giuseppe Ascia

```
{
  for(i=0;i<10;i++)
    scanf(A[i]);

  min=A[0];

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

  printf(min);
}
```

Fondamenti di Informatica

21

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

Prof. Giuseppe Ascia

```
{
  for(i=0;i<10;i++)
    scanf(A[i]);

  imin=0;

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

  printf(imin);
}
```

Fondamenti di Informatica

22

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

Prof. Giuseppe Ascia

```
{
    for(i=0;i<10;i++)
        scanf(A[i]);

    for(i=0;i<10;i++)
        printf(A[i]);

    printf("Inserisci il primo indice ");
    scanf(j);
    printf("Inserisci il secondo indice");
    scanf(j);

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

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

Fondamenti di Informatica

23

## Scambia l'elemento minimo con il massimo di un vettore di 10 elementi

Prof. Giuseppe Ascia

```
{
    for(i=0;i<10;i++)
        scanf(A[i]);
    for(i=0;i<10;i++)
        printf(A[i]);
    imin=0;
    imax=0;
    for(i=1;i<10;i++)
        if(A[i]<A[imin]) imin=i;
        else if(A[i]>A[imax]) imax=i;

    temp=A[imin];
    A[imin]=A[imax];
    A[imax]=temp;

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

Fondamenti di Informatica

24

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

Prof. Giuseppe Ascia

```
{
  for(i=0;i<10;i++)
    scanf(A[i]);
  for(i=0;i<10;i++)
    printf(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(A[i]);
}
```

Fondamenti di Informatica

25

## break

Prof. Giuseppe Ascia

- 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(A[i]);
```

Fondamenti di Informatica

26

## switch

Prof. Giuseppe Ascia

- 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;
}
```

Fondamenti di Informatica

27

## Realizzare una delle 4 operazioni (+, -, \*, /)

Prof. Giuseppe Ascia

```
{ printf("Inserire il primo operando");
  scanf(a);
  printf("Inserire il secondo operando");
  scanf(b);
  printf("Quale operazione? (+,-,*,/)");
  scanf(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(risultato);
}
```

Fondamenti di Informatica

28