

Laboratorio di Informatica

Flusso del controllo: Iterazione Strutture dati: array

Pagina 1

Flusso del controllo

- Le istruzioni di un programma vengono normalmente eseguite in sequenza
- Spesso è però necessario alterare l'ordine sequenziale per:
 - Scegliere tra azioni alternative (selezione)
 - `if`, `if - else`, `switch`
 - Iterare un blocco di istruzioni (iterazione)
 - `while`, `for`

Pagina 2

Teorema di Bohm e Jacopini

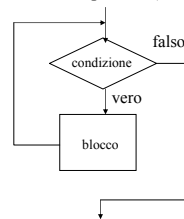
Tutti i programmi possono essere scritti in termini di tre strutture di controllo:

- **Sequenza:** permette di eseguire le istruzioni secondo l'ordine in cui sono scritte
- **Selezione:** permette di scegliere, in base al valore di una condizione, il blocco di istruzioni da eseguire
- **Iterazione:** permette di eseguire ripetutamente un blocco di istruzioni

Pagina 3

Iterazione: while ... do

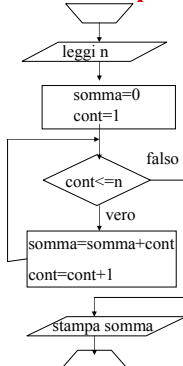
Fino a quando (mentre) la condizione è vera esegui blocco



`while (condizione)`
`blocco`

Pagina 4

while ... do: somma dei primi n numeri naturali



```
leggi n
somma=0
cont=1
while (cont<=n)
    somma=somma+cont
    cont=cont+1
stampa somma
```

N. B.: se $n = 0$ il primo controllo fa stampare somma senza eseguire nemmeno una volta il ciclo

Pagina 5

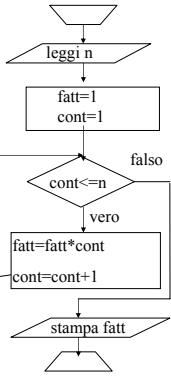
while ... do Esercizio: fattoriale di n

Scrivere un algoritmo che calcoli il fattoriale di un numero n chiesto all'utente

N.B. $n! = n * (n-1) * (n-2) \dots * 1$
 $0! = 1$

Pagina 6

while ... do: fattoriale di n



leggi n

fatt=1

cont=1

while (cont<=n)

fatt=fatt*cont

cont=cont+1

stampa fatt

N. B.: se n = 0 il primo controllo fa stampare fatt senza eseguire nemmeno una volta il ciclo

Si noti il costrutto:

expr 1

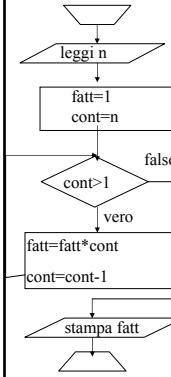
while (expr2)

istruzioni

expr3

Pagina 7

while ... do: fattoriale di n



leggi n

fatt=1

cont=n

while (cont>1)

fatt=fatt*cont

cont=cont-1

stampa fatt

N. B.: se n = 0 il primo controllo fa stampare fatt senza eseguire nemmeno una volta il ciclo

Si noti il costrutto:

expr 1

while (expr2)

istruzioni

expr3

Pagina 8

C: while

- La forma generale è:

```
while (<espressione>)  
<istruzione>
```
- Se espressione è vera allora viene eseguita *istruzione* altrimenti il controllo passa all'istruzione successiva
- Istruzione può anche essere composta:

```
while (<espressione>)  
{  
<istruzioni>  
}
```

Pagina 9

C-while:

somma dei primi n numeri naturali

Leggi n

```
int cont = 0;
```

```
while (cont<=n)
```

```
{
```

```
    somma = somma + cont;
```

```
    cont = cont+1;
```

```
}
```

```
printf("somma è %i", somma);
```

Pagina 10

while ... do Esercizio: fattoriale di n

Scrivere un programma in C che calcoli il fattoriale di un numero n chiesto all'utente usando l'istruzione while

Pagina 11

Iterazione: for

```
for (expr1, expr2, expr3)
```

```
istruz
```

```
expr1;
```

```
while (expr2){
```

```
    istruz
```

```
    expr3;}
```

```
for (cont = 1; cont <= n; cont++)
```

```
    somma = somma + cont;
```

```
printf("somma è %i", somma);
```

```
cont = 1;
```

```
while (cont<=n){
```

```
    somma = somma + cont;
```

```
    cont = cont+1;}
```

```
printf("somma è %i", somma);
```

Il C utilizza ++ (--) per incrementare (decrementare) i valori della variabile di uno
i++ equivalente a i=i+1 i-- equivalente a i=i-1

Pagina 12

for Esercizio: fattoriale di n

Scrivere un programma in C che calcoli il fattoriale di un numero n chiesto all'utente usando l'istruzione for

Pagina 13

Array monodimensionali

I programmi fanno spesso uso di *dati omogenei*, ad esempio la somma di una lista di numeri.

Invece che definire una variabile per ogni dato si utilizza una struttura dati derivata: **gli array**.

Possono essere pensati come sequenze di elementi di un certo tipo (es. char, int, ...)

1	3	66	1	23
---	---	----	---	----

 array di 5 interi

Pagina 14

Array monodimensionali

Dichiarazione: simile a una variabile ma con la lunghezza dell'array tra parentesi quadre

Es: `int a[3];`

Es: `#define N 3`
`int a[N];`

`<tipo> <NomeArray> [<lunghezza>]`

Inizializzazione: il valore di inizializzazione viene racchiuso tra parentesi graffe:

Es: `int a[N]={0};`
Es: `int a[N]={5,6,7};`

Pagina 15

Array monodimensionali

1	3	66	1	23
---	---	----	---	----

i = 0 1 2 3 4

Indice: varia tra 0 e lunghezza - 1
per accedere all'elemento i-esimo `a[i]`

Scorrimento: essendo nota la lunghezza del vettore si usa solitamente l'istruzione `for`

Pagina 16

Array monodimensionali: esempio

```
/* Somma degli elementi di un vettore. */
```

```
#include <stdio.h>
#include <conio.h>
#define N 5
```

```
int main(void){
    int i, somma=0;
    int a[N]={2,4,6,3,8};
```

```
    for (i = 0; i<N; i++){
        somma = somma + a[i];
    }
    printf("la somma dei numeri in a è: %d", somma);
    getch();
    return 0;}
```

Pagina 17

Array monodimensionali: esempio

```
/* Somma degli scarti degli elementi di un vettore. */
```

```
#include <stdio.h>
#include <conio.h>
#define N 3
```

```
int main(void){
    int i;
    double somma=0,somma_scarti=0;
    double media=0;
    int a[N]={2,4,6};
    double scarti[N]={0};

    for (i = 0; i<N; i++){
        somma = somma + a[i];
        media=somma/N;
    }

    for (i = 0; i<N; i++){
        scarti[i]=a[i]-media;
        somma_scarti = somma_scarti+scarti[i];
    }
```

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

    printf("media: %lf \n", media);
```

```
    printf("scarti: \n");
    for (i = 0; i<N; i++){
        printf("%lf ", scarti[i]);
    }
```

```
    printf("la somma degli scarti di a è: %lf \n", somma_scarti);
    getch();
    return 0;}
```

Pagina 18

Stringhe

- Una stringa è un array di caratteri
- Dichiarazione e inizializzazione:
`char <nome_stringa> [<lunghezza>]=“”`
- Delimitatore: ogni stringa termina con un
marcatore: `\0`

“abcd”=

a	b	c	d	\0
---	---	---	---	----