POLITECNICO DI MILANO

Strutture di Controllo

Informatica B

Daniele Loiacono

C vs MATLAB

Strutture di controllo in Matlab

```
if (<expr1>)
        statement;
else if (<expr2>)
        statement;
else
        statement;
```

C

```
if <expr1>
    statements
elseif <expr2>
    statements
else
    statements
end
```

MATLAB

Strutture di controllo in Matlab

```
for (<iniz>;<term>;<agg>)
{
        statement1;
        …
        statementN;
}
```

C

```
for variable = init:step:end
        statement
        ...
        statement
end
```

MATLAB

Strutture di controllo in Matlab

```
while (<expr>)
{
        statement1;
        …
        statementN;
}
```

C

```
while <expr>
        statement
        …
        statement
end
```

MATLAB

Condizioni in C

Vero e falso in C

- In C non esiste un tipo di dato specifico per rappresentare i concetti **vero** e **falso**
- Una condizione assume un valore **intero** pari a
  - 0 se la condizione è **falsa**
  - 1 se la condizione è **vera**
- In generale, ogni valore **diverso da zero** è considerato **vero**
  - (3) → VERO
  - (1) → VERO
  - (a – a) → FALSO

Operatori relazionali

- Gli operatori relazionali operano su valori numerici e carattere

  a OP *b*

- *a* e *b* possono essere espressioni aritmetiche, variabili, o costanti
- OP può essere
  - > maggiore
  - < minore
  - >= maggiore o uguale
  - <= minore o uguale
  - == uguale
  - != diverso

Operatori relazionali: esempi

```
if (5>=4) /* vero */
    ...

if (3!=3) /* falso */
    ...

if (2e12 < 3.42) /* falso */
    ...

if ('c'=='d') /* falso */
...

if ('a' < 'c') /* vero */
    ...
```

Operatori logici

- Gli operatori logici consentono di costruire delle condizioni complesse a partire da condizioni più semplici
- I quattro operatori logici principali sono: AND, OR, NOT e XOR
- AND, OR e XOR sono operatori binari, NOT è un operatore unario
- Gli operatori vengono definiti attraverso una **tavola della verità**:

A	B	A AND B	A OR B	A XOR B
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

A	NOT A
0	1
1	0

Operatori logici in C

- Sintassi operatori logici
  - <condizione> && <condizione> AND
  - <condizione> || <condizione> OR
  - ! <condizione> NOT

- Dove <condizione>:
  - può contenere a sua volta sotto-condizioni definite sia attraverso operatori relazionali che logici
  - dovrà avere comunque un valore intero: se tale valore è 0 verrà valutata come **falsa**, altrimenti verrà valutata come **vera**.

Operatori logici: esempio

```
int a;
char c;

if ( a>=5 && a<9) /* vero se 5 <= a < 9 */

if ( !( c >= 'a' && c<='z' ) )
    printf ("%c non è una lettera minuscola\n",c);

if ( c < 'a' || c >'z' )
    printf ("%c non è una lettera minuscola\n",c);
```

Ordine tra gli operatori

- Un'espressione viene valutata nel seguente ordine:
  - Operatori ++ e –
  - !
  - operatori aritmetici
  - operatori relazionali
  - &&
  - ||
- È possibile utilizzare le parentesi tonde per specificare la precedenza desiderata
  - Non si possono usare altri tipi di parentesi per questo scopo
  - È possibile inserire diversi livelli di parentesi (tonde) uno dentro l'altro

Esempio

```c
/* Programma assicurazione.c */
#include <stdio.h>

int main()
{
    int anni, cc,  giovane,altaCC;
    printf("Inserire l'eta': ");    scanf("%d",&anni);
    printf("Inserire cilindarata: ");    scanf("%d",&cc);
    giovane = (anni<=20);
    altaCC = (cc>1400);
    if (giovane && altaCC)
        printf("Incremento: 70 per cento\n");
    if (giovane && !altaCC)
        printf("Incremento: 40 per cento\n");
    if (!giovane && altaCC)
        printf("Incremento: 10 per cento\n");
    if (!giovane && !altaCC)
        printf("Incremento: nessuno\n");

    return 0;
}
```

Assegnamento ≠ Confronto

- Assegnamento

```
int a = 0, b = 4;
a = b;
printf( "%d", a );
```

- Confronto

```
int a = 0, b = 4;
if ( a == b )
  printf( "uguali" );
```

```
int a = 0, b = 4;
if ( a = b )
  printf( "uguali" );
```

1) a=4
2) (a=b) → 4 → vero

Costrutti condizionali

Istruzione if: esempio

```
#include<stdio.h>
int main()
{
    char c;
    printf("Inserire il carattere maiuscolo: ");
    scanf("%c", &c);
    printf("La traduzione e' %c\n",c+32);

    return 0;
}
```

- Se il carattere inserito non è una lettera maiuscola?

Istruzione if: esempio

```
#include<stdio.h>
int main()
{
    char c;
    printf("Inserire il carattere maiuscolo: ");
    scanf("%c", &c);
    if (c>='A' && c<='Z')
        printf("La traduzione e' %c\n",c+32);
    return 0;
}
```

- Se c è una lettera maiuscola eseguo la traduzione, altrimenti?

Istruzione condizionale if-else

- Consente di scegliere fra due alternative nel flusso di esecuzione

```
if (anni>18)
    printf("Hai diritto al voto\n");
else
    printf("Non puoi votare\n");
```

```
if ( a==b )
    printf("a e b sono uguali\n");
else
    printf("a e b sono diversi\n");
```

```
if (<condizione>)
        statement;
else
        statement;
```

- **Semantica**: il primo statement viene eseguito **solo** se la condizione è **vera,** il secondo statement viene eseguito solo se lacondizione è **falsa**

Istruzione if-else: esempio

```c
#include<stdio.h>

int main()
{
    char c;
    printf("Inserire il carattere maiuscolo: ");
    scanf("%c", &c);
    if (c>='A' && c<='Z')
        printf("La traduzione e' %c\n",c+32);
    else
        printf ("Il carattere inserito non e' una lettera maiuscola\n");

    return 0;
}
```

Statement composti

- Scrivere un programma che, letti due numeri individua quello maggiore e quello minore

```
#include<stdio.h>

int main()
{
    float max, min, temp;
    printf("Inserire il primo numero: ");
    scanf("%f", &max);
    printf("Inserire il secondo numero: ");
    scanf("%f", &min);
    if (max<min)
        temp = min;
        min = max;
        max = temp;
    printf("max=%f - min=%f\n",max,min);
    return 0;
}
```

Statement composti

- Scrivere un programma che, letti due numeri individua quello maggiore e quello minore

```
#include<stdio.h>

int main()
{
    float max, min, temp;
    printf("Inserire il primo numero: ");
    scanf("%f", &max);
    printf("Inserire il secondo numero: ");
    scanf("%f", &min);
    if (max<min)
    {
        temp = min;
        min = max;
        max = temp;
    }
    printf("max=%f - min=%f\n",max,min);
    return 0;
}
```

Scegliere fra molte alternative

```
#include<stdio.h>

int main()
{
    int pesoMoneta;
    scanf("%d",&pesoMoneta);
    if (pesoMoneta==9)
        printf ("5 centesimi\n");
    if (pesoMoneta==16)
        printf ("10 centesimi\n");
    if (pesoMoneta==19)
        printf ("20 centesimi\n");
    if (pesoMoneta==35)
        printf ("50 centesimi\n");
     return 0;
}
```

Se non si verifica nessuna condizione?

Scegliere fra molte alternative

```
#include<stdio.h>

int main()
{
    int pesoMoneta;
    scanf("%d",&pesoMoneta);
    if (pesoMoneta==9)
        printf ("5 centesimi\n");
        else if (pesoMoneta==16)
            printf ("10 centesimi\n");
            else if (pesoMoneta==19)
                printf ("20 centesimi\n");
                else if (pesoMoneta==35)
                    printf ("50 centesimi\n");
                    else
                        printf("Non riconusciuta!");

    return 0;
}
```

Poco leggibile

Scegliere fra molte alternative

```
#include<stdio.h>

int main()
{
    int pesoMoneta;
    scanf("%d",&pesoMoneta);
    if (pesoMoneta==9)
        printf ("5 centesimi\n");
    else if (pesoMoneta==16)
        printf ("10 centesimi\n");
    else if (pesoMoneta==19)
        printf ("20 centesimi\n");
    else if (pesoMoneta==35)
        printf ("50 centesimi\n");
    else
        printf("Non riconusciuta!");

    return 0;
}
```

Esiste un costrutto specifico per le selezioni multiple!

Cicli

Ripetere le istruzioni più volte…

- Supponiamo di voler trasformare 3 lettere maiuscole in minuscole:

```
…
printf("Inserire il carattere maiuscolo: ");
scanf("%c", &c);
if (c>='A' && c<='Z')
   printf("La traduzione e' %c\n",c+32);
printf("Inserire il carattere maiuscolo: ");
scanf("%c", &c);
if (c>='A' && c<='Z')
   printf("La traduzione e' %c\n",c+32);
printf("Inserire il carattere maiuscolo: ");
scanf("%c", &c);
if (c>='A' && c<='Z')
   printf("La traduzione e' %c\n",c+32);
…
```

È proprio necessario ripetere più volte lo stesso pezzo di codice?

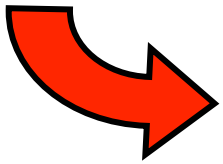
Il ciclo for

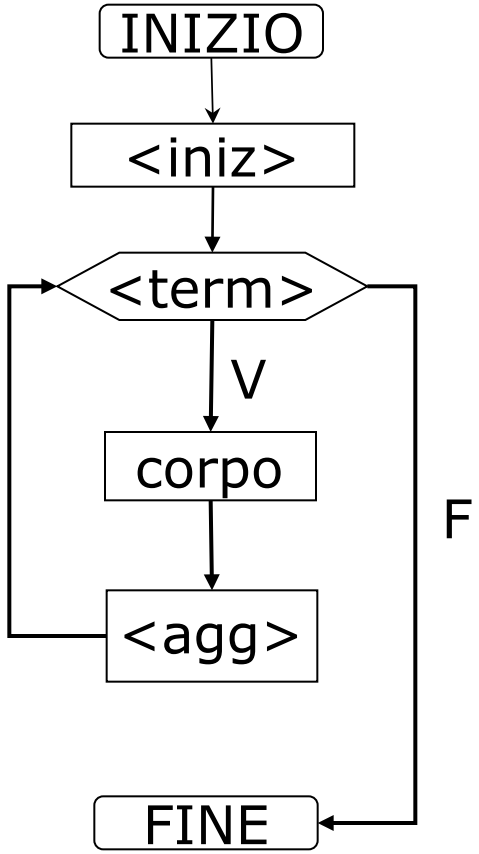
```
for (<iniz>;<term>;<agg>)
{
        statement1;
        …
        statementN;
}
```

Corpo del ciclo

Il ciclo for

```
for (<iniz>;<term>;<agg>)
{
        corpo
}
```





Esempio

```
int cont;
for (cont = 0; cont < N; cont ++)
{
        …
}
```

Il ciclo for: esempio

- Scrivere un programma per trasformare tre lettere maiuscole in minuscole

```
#include<stdio.h>

int main()
{
    char c;
    int cont;

    for (cont = 0; cont < 3; cont ++)
    {
        printf("Inserire il carattere maiuscolo: ");
        scanf("%c", &c);
        if (c>='A' && c<='Z')
            printf("La traduzione e' %c\n",c+32);
    }

    return 0;
}
```

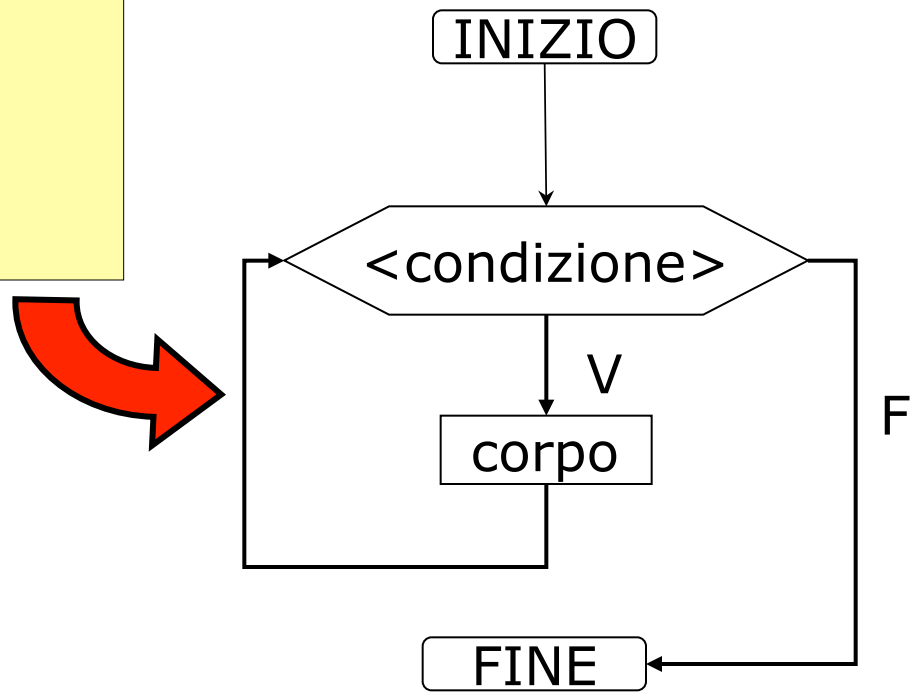
Il ciclo for: esempio (2)

```c
#include<stdio.h>

int main()
{
    char c;
    int cont,N;
    printf("Quante lettere vuoi convertire? ");
    scanf("%d",&N);
    for (cont = 0; cont < N; cont ++)
    {
        printf("Inserire il carattere maiuscolo: ");
        scanf("%c", &c);
        if (c>='A' && c<='Z')
            printf("La traduzione e' %c\n",c+32);
        else
            cont--;
    }
     return 0;
}
```

Il ciclo for consente di eseguire un numero di iterazioni definito run-time!

Il ciclo while

```
while (<condizione>)
{
        corpo
}
```



Esempio

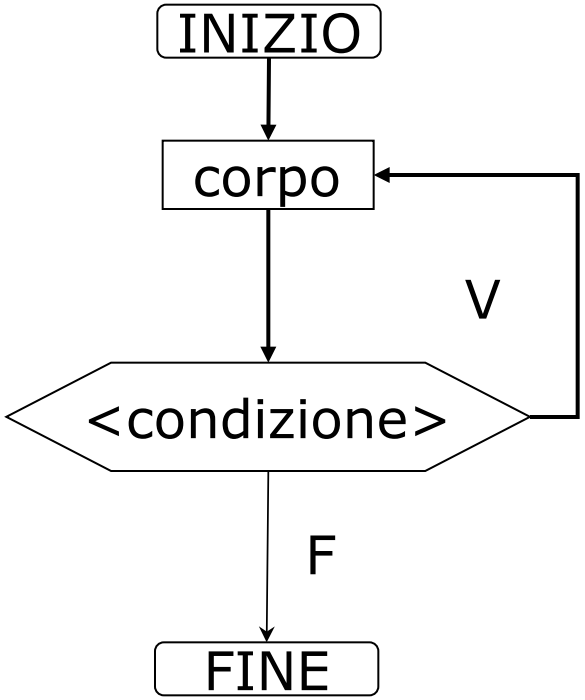
```
int cont = 0;
while (cont < N)
{
        …
        cont ++;
}
```

Il ciclo while: esempio

```
#include<stdio.h>

int main()
{
    char c;

    printf("Inserire una lettera maiuscola: ");
    scanf("%c", &c);

    while (c>='A' && c<='Z')
    {
      printf("La traduzione e' %c\n",c+32);
     printf("Inserire una lettera maiuscola: ");
     scanf("%c", &c);

    }
    return 0;
}
```

Il ciclo do-while

```
do
{
        corpo
} while (<condizione>);
```



Esempio

```
int cont = 0;
do
{
        …
        cont ++;
} while (cont < N);
```

Il ciclo do-while: esempio

```
#include<stdio.h>

int main()
{
    char c;

    do
    {
        printf("Inserire una lettera maiuscola: ");
        scanf("%c", &c);
        if (c>='A' && c<='Z')
            printf("La traduzione e' %c\n",c+32);
    }while (c>='A' && c<='Z');

    return 0;
}
```

Istruzioni break e continue

- L'istruzione **break** all'interno di un ciclo lo interrompe immediatamente:

```
for (i=0; i<10; i++) {
    scanf("%d",&x);
    if (x < 0)
        break;
}
```

- L'istruzione **continue** passa direttamente all'iterazione seguente

```
for (i=0; i<10; i++) {
    scanf("%d",&x);
    if (x < 0)
        continue;
}
```