




**UNIVERSITÀ DEGLI STUDI
DEL SANNIO** Benevento
DING
DIPARTIMENTO DI INGEGNERIA

CORSO DI "PROGRAMMAZIONE I"

Prof. Franco FRATTOLILLO
Dipartimento di Ingegneria
Università degli Studi del Sannio


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 1



Teorema di Bohm e Jacopini

- Tutti i programmi possono essere scritti in termini di tre strutture di controllo:
 - La sequenza, che permette di eseguire le istruzioni secondo l'ordine sequenziale in cui sono scritte
 - La selezione, che permette di scegliere, in base al valore di una condizione, quale istruzione eseguire
 - L'iterazione, che permette di eseguire ripetutamente un blocco di istruzioni


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 2



Linguaggio naturale

- se...allora...altrimenti....
- qualora.....manel caso in cui...
- ripeti fino a quando...
- mentre.....
- nel primo caso...., nel secondo caso....,..... nell'ennesimo caso.....
- nei linguaggi di programmazione questi costrutti sintattici vengono chiamati "strutture di controllo"

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 3



Strutture di controllo del flusso

- Le strutture di controllo del flusso specificano l'ordine secondo il quale le operazioni devono essere effettuate
- istruzioni decisionali
- istruzioni cicliche

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 4

Istruzioni decisionali

- if
- if-else
- switch

La struttura di selezione singola if

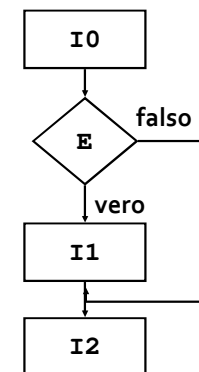
- Usata per scegliere tra diverse azioni alternative
- Esempio (Pseudocodice):
 - Se il voto di uno studente è maggiore o uguale a 18
 - Stampa "Promosso"
- Se la condizione è true, viene eseguita l'istruzione di stampa e il programma prosegue con la successiva istruzione
- Se la condizione è false, l'istruzione di stampa è ignorata e il programma prosegue con la successiva istruzione
- L'indentazione delle istruzioni migliora la leggibilità del programma
- Esempio:
 - `if (voto >= 18)`
 - `printf("Promosso\n");`

if

- Sintassi:
if (espressione)
 istruzione
- Semantica:
 - Valuta espressione
 - Se l'espressione è vera (cioè assume valore diverso da zero) viene eseguita istruzione
 - In ogni caso continua eseguendo la prima istruzione successiva alla struttura di controllo

if: diagramma di flusso

I0;
if (E) I1;
I2;



la struttura introduce due possibili sequenze:

1. la sequenza (se **E** è vera):

I0 I1 I2

2. la sequenza (se **E** è falsa):

I0 I2

Esempio

```
#include <stdio.h>
main() {
    int i;
    printf("Inserisci un intero: ");
    scanf("%d", &i);
    if (i<100)                               Stampa "Minore di 100" se il
        printf("Minore di 100\n");           numero letto è minore di 100
    }
```

Operatori relazionali

== uguale
 != diverso
 < minore
 <= minore o uguale
 > maggiore
 >= maggiore o uguale


- Restituiscono 0 quando il risultato è falso, 1 quando il risultato è vero


Operatori logici

- Un operatore unario
 - not !
 - !x vale 0 quando x è pari ad un valore diverso da 0, vale 1 altrimenti
- Due operatori binari:
 - and &&
 - a && b vale 1 se entrambi a e b sono pari ad un valore diverso da 0, vale 0 se almeno uno tra a e b vale 0
 - or ||
 - a || b vale 1 se almeno uno tra a e b vale un valore diverso da 0, vale 0 se sia a che b valgono 0

Attenzione

== diverso da =
 assegnamento
 uguaglianza






if + istruzione composta

```
#include <stdio.h>
main() {
    int i;
    int min100=0;
    printf("Inserisci un intero: ");
    scanf("%d", &i);
    if (i<100) {
        min100=1;
        printf("Minore di 100 ? : %d\n", min100);
    }
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 13




Esercizio

- Scrivere un programma che richieda in ingresso un valore intero e stampa il suo valore assoluto
- Valore assoluto di un numero n:
 - Se $n < 0$: $-n$
 - Se $n \geq 0$: n
- Esempio:

9	→	9
-27	→	27

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 14



Soluzione

```
#include<stdio.h>
main( ) {
    int n;
    printf("Inserisci un intero: ");
    scanf("%d", &n);
    if (n<0) n=-n;
    printf("Risultato: %d", n);
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 15



Istruzioni decisionali

- if
- if-else
- switch

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 16

if-else

- Sintassi

```

if (espressione)
    istruzione_1
else
    istruzione_2
  
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 17

if-else: semantica

- Viene valutata espressione
- Se l'espressione è vera (cioè assume valore diverso da zero), viene eseguita istruzione_1
- Se l'espressione è falsa (cioè assume valore nullo), viene eseguita istruzione_2

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 18

if-else: diagramma di flusso

```

I0;
if (E) I1;
else I2;
I3;
  
```

la struttura introduce due possibili sequenze:

1. la sequenza (se E è vera):
I0 I1 I3
2. la sequenza (se E è falsa):
I0 I2 I3

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 19

Esempio

```

#include <stdio.h>
main( ) {
    int i;
    printf("Inserisci un intero: ");
    scanf("%d", &i);
    if (i<100)
        printf("Minore di 100\n");
    else
        printf("Maggiore o uguale a 100\n");
}
  
```

Stampa "**Minore di 100**"
 se il numero letto è minore di 100
 Stampa "**Maggiore o uguale a 100**"
 se il numero letto è maggiore o uguale di 100

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 20

if-else + istruzione composta

```
#include <stdio.h>
main() {
    int i, mag100=0, min100=0;
    printf("Inserisci un intero: ");
    scanf("%d", &i);
    if (i<100) {
        min100=1;
        printf("Minore di 100 ? : %d\n", min100);
    }
    else {
        mag100=1;
        printf("Maggiore o uguale a 100 ? : %d\n", mag100);
    }
}
```

Esempio

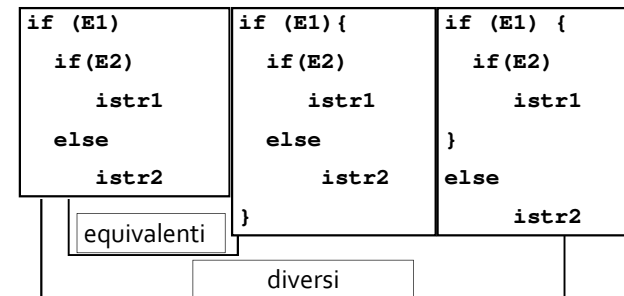
```
#include<stdio.h>
main() {
    int x;
    scanf("%d", &x);
    if (x==0) {      // if (!x)
        printf("pippo "); printf("pluto");
    }
    else
        printf("minnie");
}
```


if annidati

- Le istruzioni che compaiono nell'if non sono soggette ad alcuna restrizione, per cui possono essere anch'esse if:
 - if annidati

Ambiguità in caso di if annidati

- else è relativa al più vicino degli if che lo precedono ed in cui manchi l'else






Esercizio

- Scrivere un programma che richiede in ingresso due valori interi e stampa il maggiore

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
25




Soluzione

```

#include <stdio.h>
main() {
    int n1, n2;
    printf("Inserisci due interi separati dallo spazio: ");
    scanf("%d", &n1);
    scanf("%d", &n2); // scanf("%d %d", &n1, &n2);
    if (n1>n2)
        printf("Il maggiore è: %d", n1);
    else
        printf("Il maggiore è: %d", n2);
}

```


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
26



Esercizio

- Scrivere un programma che richiede in ingresso un valore intero e stampa "pari" se il numero è pari, "dispari" altrimenti
- Esempio
 - 7 ➔ Il numero è dispari
 - 90 ➔ Il numero è pari

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
27




Soluzione

```

#include <stdio.h>
main() {
    int n;
    printf("Inserisci un intero: ");
    scanf("%d", &n);
    if (n%2==0)
        printf("Il numero è pari");
    else
        printf("Il numero è dispari");
}

```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
28




Esercizio

• Scrivere un programma per il calcolo delle radici di un'equazione di secondo grado con coefficienti reali: a, b, c

$$ax^2 + bx + c = 0$$

a e b sono 0 ➔ degenerare
 a==0 ➔ I grado
 delta < 0 ➔ determinante negativo


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 29



Soluzione...

```
#include <stdio.h>
#include <math.h>
main() {
    float a, b, c, r1, r2, delta;
    printf("Questo programma risolve un'equazione di II grado: \n");
    printf("ax^2 + bx + c: \n");
    printf("Inserisci il coefficiente a: ");
    scanf("%f", &a);
    printf("Inserisci il coefficiente b: ");
    scanf("%f", &b);
    printf("Inserisci il coefficiente c: ");
    scanf("%f", &c);
```


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 30



Soluzione (cont.)

```
if (a==0 && b==0)
    printf("equazione degenerare\n");
else
    if (a==0) {
        r1=-c/b; printf("equazione di I grado \n"); printf("radice = %f \n", r1);
    }
    else {
        delta = b*b - 4*a*c;
        if (delta<0)
            printf("determinante negativo\n");
        else {
            delta = sqrt(delta); r1 = (-b + delta) / (2*a); r2 = (-b - delta) / (2*a);
            printf("equazione di II grado\n"); printf("radici = %f e %f \n", r1, r2);
        }
    }
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 31



Esercizio

- In accordo alle regole del Calendario Gregoriano, un anno è bisestile se è un multiplo di 4 e non è un secolo, oppure un secolo multiplo di 400
- Scrivere un programma che verifichi se l'anno inserito da tastiera è bisestile
- Esempio
 - 1992 bisestile (multiplo di 4)
 - 1800 no (multiplo di 4, secolo non multiplo di 400)
 - 2000 bisestile (secolo multiplo di 400)

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 32

Soluzione

```
#include<stdio.h>
main() {
    int anno, ris;
    printf("Inserisci l'anno: ");
    scanf("%d", &anno);
    ris = ((anno % 4 == 0) && (anno % 100 != 0)) || (anno % 400 == 0);
    if (ris == 1)
        printf("Anno bisestile");
    else
        printf("Anno NON bisestile");
}
```

Esercizio

- Scrivere un programma che richiede in ingresso tre valori interi e ne stampa il maggiore

Soluzione

```
#include <stdio.h>
main() {
    int n1, n2, n3;
    printf("Inserisci 3 interi separati dallo spazio: ");
    scanf("%d", &n1); scanf("%d", &n2); scanf("%d", &n3);
    if (n1>n2)
        if (n1>n3)
            printf("Il maggiore è: %d", n1);
        else
            printf("Il maggiore è: %d", n3);
    else
        if (n2>n3)
            printf("Il maggiore è: %d", n2);
        else
            printf("Il maggiore è: %d", n3);
}
```

... oppure

```
#include <stdio.h>
main() {
    int n1, n2, n3;
    printf("Inserisci 3 interi separati dallo spazio: ");
    scanf("%d", &n1); scanf("%d", &n2); scanf("%d", &n3);
    if ((n1>n2) && (n1>n3))
        printf("Il maggiore è: %d", n1);
    else if (n2>n1 && n2>n3)
        printf("Il maggiore è: %d", n2);
    else
        printf("Il maggiore è: %d", n3);
}
```

... oppure

```
#include <stdio.h>
main() {
    int n1, n2, n3, max;
    printf("Inserisci 3 interi separati dallo spazio: ");
    scanf("%d", &n1); scanf("%d", &n2); scanf("%d", &n3);
    max=n1;
    if (n2 > max) max=n2;
    if (n3 > max) max=n3;
    printf("Il maggiore è: %d", max);
}
```

Istruzione else-if

- Quando occorre effettuare una scelta plurima, è possibile utilizzare l'istruzione else-if
- Sintassi:


```
if (espressione)
    <istruzione/i>
else if (espressione)
    <istruzione/i>
else if (espressione)
    <istruzione/i>
else
    <istruzione/i>
```

Esempio

- Questo codice riconosce il carattere digitato a video assumendo che i caratteri possibili si dividono in lettere, in cifre ed in simboli
- ```
c = getchar();
if (c >= '0' && c <= '9')
 printf("E' una cifra\n");
else if (c >= 'a' && c <= 'z')
 printf("E' una lettera minuscola\n");
else if (c >= 'A' && c <= 'Z')
 printf("E' una lettera maiuscola\n");
else
 printf("E' un simbolo\n");
```

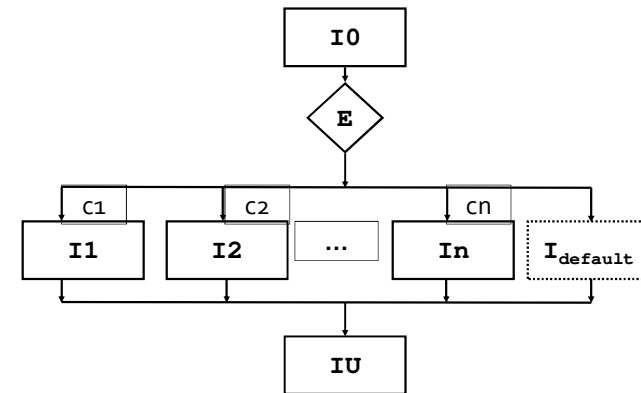
## Istruzioni decisionali

- if
- if-else
- switch

## switch

- Struttura di scelta plurima che controlla se un'espressione assume un valore all'interno di un certo insieme di costanti e si comporta di conseguenza

## switch



## switch: sintassi

```

switch (espressione) {
 case costante1: sequenzaDiIstruzioni1; break;
 case costante2: sequenzaDiIstruzioni2; break;
 ...
 case costanteN: sequenzaDiIstruzioniN; break;
 default: sequenzaDiIstruzioneDefault;
}

```

## Esempio

```

#include<stdio.h>
main() {
 int num;
 scanf("%d", &num);
 switch (num) {
 case 1: printf("*\n"); break;
 case 2: printf("**\n"); break;
 case 3: printf("***\n"); break;
 default: printf("! \n");
 }
}

```

## Osservazione

- I valori ammessi come possibili scelte devono essere costanti

## Esempio

```
#include<stdio.h>
main() {
 int num;
 int uno=1;
 int due=2;
 scanf("%d", &num);
 switch (num) {
 case uno: printf("pippo1\n"); break;
 case due: printf("pippo2\n"); break;
 default: printf("pippo_default\n");
 }
}
```

SBAGLIATO  
Valori non costanti

## Osservazione

- break non è strettamente indispensabile
- Se non è presente, viene eseguita sequenzialmente ogni istruzione a partire dal case che è stato raggiunto

## Esempio senza break

```
#include<stdio.h>
main() {
 int num;
 scanf("%d", &num);
 switch (num) {
 case 1: printf("**\n");
 case 2: printf("***\n");
 case 3: printf("****\n");
 default: printf("!\n");
 }
}
```

## Un altro esempio

```

#include<stdio.h>
main(){
char car;
scanf("%c", &car);
switch (car) {
case 'a': case 'e': case 'i':
case 'o': case 'u':
printf("Vocale minuscola\n"); break;
case 'A': case 'E': case 'I':
case 'O': case 'U':
printf("Vocale maiuscola\n"); break;
default: printf("Non e' vocale\n");
}
}

```

Possono esserci più  
etichette per una stessa  
sequenza di istruzioni

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    49

## Esercizio

- Scrivere un programma che visualizza il seguente menu:
  - Menu di prova
  - Immettere dati
  - Visualizzare dati
  - Modificare dati
  - Scelta:
- quindi aspetta l'immissione di un carattere da parte dell'utente e visualizza una scritta corrispondente alla scelta effettuata, del tipo: In esecuzione l'opzione 1
- Se la scelta non è tra quelle proposte (1,2,3), deve essere visualizzata la scritta "Opzione inesistente"

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    50

## Soluzione

```

#include<stdio.h>
main(){
int num;
printf("Menu di prova\n");
printf("1) Immettere dati\n"); printf("2) Visualizzare dati\n"); printf("3) Modificare dati\n");
printf("Scelta: ");
scanf("%d", &num);
switch (num){
case 1: printf("In esecuzione l'opzione 1"); break;
case 2: printf("In esecuzione l'opzione 2"); break;
case 3: printf("In esecuzione l'opzione 3"); break;
default: printf("Opzione inesistente");
}
}

```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    51

## Esercizio

- Scrivere un programma che riceve in ingresso un mese (numero intero) e stampa quanti giorni ha quel mese

|                   |   |               |
|-------------------|---|---------------|
| • 4,6,9,11        | ➔ | 30            |
| • 1,3,5,7,8,10,12 | ➔ | 31            |
| • 2               | ➔ | 28            |
| • m ∉ [1..12]     | ➔ | non esistente |

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    52

## Soluzione

```

main() {
 int num; printf("Inserisci un mese (numero): ");
 scanf("%d", &num);
 switch (num) {
 case 1: printf("31 giorni"); break;
 case 2: printf("28 giorni"); break;
 case 3: printf("31 giorni"); break;
 case 4: printf("30 giorni"); break;
 case 5: printf("31 giorni"); break;
 case 6: printf("30 giorni"); break;
 case 7: printf("31 giorni"); break;
 case 8: printf("31 giorni"); break;
 case 9: printf("30 giorni"); break;
 case 10: printf("31 giorni"); break;
 case 11: printf("30 giorni"); break;
 case 12: printf("31 giorni"); break;
 default: printf("Mese inesistente");
 }
}

```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 53

## ... oppure

```

#include <stdio.h>
main() {
 int num;
 printf("Inserisci un mese (numero): "); scanf("%d", &num);
 switch (num) {
 case 4: case 6: case 9: case 11: printf("30 giorni"); break;
 case 1: case 3: case 5: case 7:
 case 8: case 10: case 12: printf("31 giorni"); break;
 case 2: printf("28 giorni"); break;
 default: printf("Mese inesistente");
 }
}

```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 54

## Strutture di controllo del flusso

- Le strutture di controllo del flusso specificano l'ordine secondo il quale le operazioni devono essere effettuate
- istruzione composta
- istruzioni decisionali
- istruzioni cicliche

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 55

## Istruzioni cicliche

- Consentono di realizzare cicli di elaborazione, ossia l'esecuzione ripetuta di una sequenza di istruzioni

il numero di volte per il quale  
viene ripetuta la esecuzione  
della sequenza è noto a priori

il numero di volte per il quale  
la sequenza viene ripetuta non è  
noto a priori, ma è condizionato  
dal verificarsi in esecuzione di  
un evento assegnato

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 56

## Esempi

- Calcolare le paghe dei dipendenti di una azienda
  - noto il numero  $N$  dei dipendenti, ripetere  $N$  volte la sequenza di calcolo della paga
- Superare l'esame di "Programmazione I"
  - la sequenza sostenere l'esame va eseguita e rieseguita fino a quando non si viene promosso (voto  $\geq 18$ )

## Istruzioni cicliche

- for
- while
- do-while

## Esempio

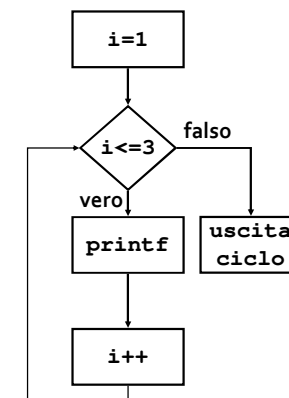
- Vogliamo visualizzare tre volte la scritta:  
Ciao, mondo.


il numero di volte per il quale viene ripetuta l'esecuzione della sequenza è noto a priori: **3**

```
printf("Ciao, mondo.\n");
printf("Ciao, mondo.\n");
printf("Ciao, mondo.\n");
```

## Usando il for

```
#include<stdio.h>
main() {
 int i;
 for (i=1; i<=3; i++)
 printf("Ciao, mondo.\n");
}
```





## Sintassi del for

- for (inizializzazione; condizione; incremento) istruzione


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    61



## Semantica

- for (inizializzazione; condizione; incremento) istruzione
- inizializzazione
  - Viene eseguita una volta sola
  - Serve per impostare le variabili
  - Può anche non essere presente; in questo caso, dopo la parentesi tonda aperta viene subito il punto e virgola


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    62



## Semantica (cont.)

- for (inizializzazione; condizione; incremento) istruzione
- condizione
  - Viene valutata ogni volta prima di eseguire le istruzioni del ciclo
  - Se condizione è vera, si esegue ancora istruzione
  - Se condizione è falsa, si esce dal ciclo for passando all'istruzione successiva del programma
  - Può anche non essere presente, ed allora lo spazio tra i due punti e virgola rimane vuoto; il compilatore valuta vera la condizione assente, quindi continua ad eseguire istruzione
    - si realizza un loop infinito

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    63



## Semantica (cont.)

- for (inizializzazione; condizione; incremento) istruzione
- incremento
  - Viene eseguita alla fine di ogni ciclo
  - Modificano, ad esempio, le variabili contatore (incremento o decremento)
    - in generale: espressione
  - Può anche non essere presente, ed allora, dopo il secondo punto e virgola, viene subito la parentesi tonda chiusa

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    64



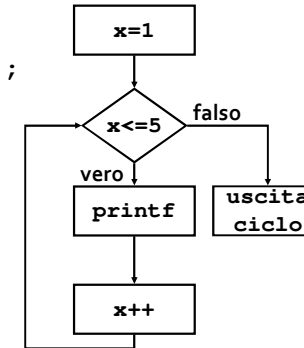
## Importante!!

- **for** (inizializzazione; condizione; incremento) istruzione
- inizializzazione: eseguita una volta sola
- condizione: valutata ogni volta prima di eseguire le istruzioni del ciclo
- incremento: eseguita alla fine di ogni ciclo più in generale è un'espressione

## Esempio

```
int x;
for(x=1; x<=5; x++)
 printf("%d\n", x);
```

```
1
2
3
4
5
```



| x | x<=5 |
|---|------|
| 1 | si   |
| 2 | si   |
| 3 | si   |
| 4 | si   |
| 5 | si   |
| 6 | no   |

## Esempio

```
int x;
for(x=10; x>=0; x-=2) printf("%d\n", x);
```

```
10
8
6
4
2
0
```

| x  | x>=0 |
|----|------|
| 10 | si   |
| 8  | si   |
| 6  | si   |
| 4  | si   |
| 2  | si   |
| 0  | si   |
| -2 | no   |

## Esempio: loop infinito

```
int x;

for (x=0; ;) printf("loop");

for (; ;) printf("loop");

for (x=0; x==0;) printf("loop");
```

```
looplooplooplooplooploop
looplooplooplooplooploop
looplooplooplooplooploop
looplooplooplooplooploop
looplooplooplooplooploop
looplooplooplooplooploop
```

### Attenzione!!!

- Iterazione infinita

```
for (i=5; i>=5; i++) ...
```

- Attenzione!! ... Il compilatore non segnala errore

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 69

### Osservazioni ...

- Poiché la condizione viene valutata prima di ogni ciclo, il for permette anche di non eseguire alcuna iterazione del ciclo

Non stampa nulla

- Esempio

```
int x;
for (x=9; x<6; x++) printf("%d\n", x);
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 70

### ... Osservazioni

- Inizializzazione ed incremento, nella sintassi del for, possono contenere più istruzioni che dovranno essere separate da virgola

- Esempio

```
int x, y;
for (x=0, y=0; x+y<10; x++, y+=3) printf("%d\n", x+y);
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 71

### Esempio

```
int x, y;
for (x=0, y=0; x+y<10; x++, y+=3)
 printf("%d\n", x+y);
```

Cosa stampa?

| x | y | x+y<10 |
|---|---|--------|
| 0 | 0 | si     |
| 1 | 3 | si     |
| 2 | 6 | si     |
| 3 | 9 | no     |

0

4

8

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 72

### Esempio di uso del for

- Calcolare la somma di 5 numeri interi immessi dall'utente:

```
#include <stdio.h>
main() {
 int i, somma, numero;
 somma=0;
 for (i=1; i<=5; i++) {
 printf("Inserisci il numero: ");
 scanf("%d", &numero);
 somma=somma + numero;
 }
 printf("Somma = %d\n", somma);
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 73

### Esempio (cont.)

- È spesso utile far apparire il numero d'ordine d'inserimento:

```
#include <stdio.h>
main() {
 int i, somma, numero;
 somma=0;
 for (i=1; i<=5; i++) {
 printf("Inserisci il numero %d: ", i);
 scanf("%d", &numero);
 somma=somma + numero;
 }
 printf("Somma = %d\n", somma);
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 74

### Esercizio

```
#include<stdio.h>
main() {
 int i, n=3;
 for (i = 0; i < n; i++)
 printf("*");
 for (i = 0; i < n; i++)
 printf("!!");
}
```

\*\*\*!!!


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 75

### Cosa stampa?

```
#include<stdio.h>
main() {
 int x;
 for (x=1; x<=3; x++) {
 printf("%d\n", x);
 x=x+2;
 }
}
```

1


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 76



## Cosa stampa?

```
#include <stdio.h>
main() {
 int a, b;
 for(a = 5; a <= 7; a++) {
 for(b = 1; b <= 3; b++)
 printf("%d ", b);
 printf("%d ", a);
 }
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    77



## Soluzione


| a | a<=7 |
|---|------|
| 5 | si   |
| 6 | si   |
| 7 | si   |
| 8 | no   |

| b | b<=3 |
|---|------|
| 1 | si   |
| 2 | si   |
| 3 | si   |
| 4 | no   |

```
for(a = 5; a <= 7; a++) {
 for(b = 1; b <= 3; b++)
 printf("%d ", b);
 printf("%d ", a);
}
```

1 2 3 5 1 2 3 6 1 2 3 7


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    78



## E questo programma?

```
#include <stdio.h>
main() {
 int a, b;
 for(a = 5; a <= 7; a++) {
 for(b = 1; b <= 3; b++) {
 printf("%d ", b);
 printf("%d ", a);
 }
 }
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    79



## Soluzione

1 5 2 5 3 5 1 6 2 6 3 6 1 7 2 7 3 7

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    80

### Riassumendo ...

- L'indice del ciclo interno scorre più velocemente dell'indice del ciclo esterno:  

```
for(i = 1; i <= n; i++) {
 ...
 for(j = 1; j <= m; j++) {
 ...
 }
 ...
}
```
- L'indice  $j$  conta da 1 a  $m$  per ogni diverso valore dell'indice  $i$

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    81

### Esercizio

- Determinare il valore delle variabili  $w$  e  $z$  dopo l'esecuzione del seguente frammento:  

```
int i, j, w, z;
w=0; z=0;
for (i=1; i<=3; i++) {
 for (j=1; j<=3; j++)
 w=w+1;
 z=z+1;
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    82

### Soluzione

|     |      |   |
|-----|------|---|
| $w$ | vale | 9 |
| $z$ | vale | 3 |
| $i$ | vale | 4 |
| $j$ | vale | 4 |


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    83

### Un altro esempio di uso del for

- Scrivere una matrice di asterischi formata da  $r$  righe e  $c$  colonne, con  $r$  e  $c$  letti da input
- Esempio  
 $r=3$   $c=2$   

```
**
**
**
```


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    84



Soluzione

```
#include<stdio.h>
main() {
 int r, c, i, j;
 scanf("%d %d", &r, &c);
 for (i = 0; i < r; i++) {
 for (j = 0; j < c; j++) printf("*");
 printf("\n");
 }
}
```


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
85



Somma dei primi n numeri

```
#include <stdio.h>
main() {
 int n, somma, i;
 printf("Inserisci n: ");
 scanf("%d", &n);
 somma=0;
 for (i=1; i<=n; i++) somma=somma+i;
 printf("La somma dei primi n num = %d", somma);
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
86




Somma dei primi n numeri

```
#include <stdio.h>
main() {
 int n;
 printf("Inserisci n: ");
 scanf("%d", &n);
 printf("Somma = %d", n*(n+1)/2);
}
```

Gauss      **$n(n+1)/2$**


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
87



Somma di n numeri

```
#include <stdio.h>
main() {
 int n, somma=0, num, i;
 printf("Inserisci n: ");
 scanf("%d", &n);
 printf("Ora inizia l'inserimento di %d numeri\n", n);
 for (i=1; i<=n; i++) {
 printf("Inserisci il numero %d: ", i);
 scanf("%d", &num);
 somma=somma+num;
 }
 printf("La somma = %d", somma);
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
88




## Massimo di n numeri

- Se uso int
- Minimo =  $-2^{31}$  = -2147483648
- Massimo =  $2^{31} - 1$  = 2147483647

```
#include <limits.h>
INT_MIN
INT_MAX
```


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    89



## Massimo di n numeri

```
#include <stdio.h>
#include <limits.h>
main() {
 int n, max=INT_MIN, num, i;
 printf("Inserisci n: ");
 scanf("%d", &n);
 printf("Ora inizia l'inserimento di %d numeri\n", n);
 for (i=1; i<=n; i++) {
 printf("Inserisci il numero %d: ", i);
 scanf("%d", &num);
 if (max < num) max=num;
 }
 printf("Il massimo= %d", max);
}
```


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    90



## Osservazione

- Soluzione alternativa ad inizializzare *max* al minimo numero negativo accettato da una variabile intera:
  - inizializzare *max* al primo numero letto dall'esterno, all'inizio del ciclo

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    91



## Esercizio


- Scrivere un programma che richiede all'utente un naturale *n* e calcola il fattoriale di *n*, indicato con *n!*

$$0! = 1$$

$$n! = n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 2 \cdot 1 \quad n > 0$$
- Esempio:
 
$$4! = 4 \cdot 3 \cdot 2 \cdot 1 = 24$$

$$6! = 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 720$$

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    92



## Fattoriale di n

```
#include <stdio.h>
main() {
 int n, fatt, i;
 printf("Inserisci n: ");
 scanf("%d", &n);
 fatt=1;
 for (i=n; i>=1; i--)
 fatt=fatt*i;
 printf("Il fattoriale di n = %d", fatt);
}
```


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    93



## Istruzioni cicliche

- for
- while
- do-while


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    94



## while

- Permette di ottenere la ripetizione ciclica di un'istruzione sotto il controllo di una condizione di terminazione

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    95



## while

- Sintassi  
while (espressione) istruzione
- Semantica
- espressione deve essere valutata ogni volta prima di eseguire istruzione
- se la valutazione di espressione è vera, allora istruzione viene eseguita, altrimenti no, e si esce dal ciclo

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    96



### while: diagramma di flusso

```

IO;
while (E) I1;
I2;

```

- le possibili sequenze sono:

```

IO I2
IO I1 I2
IO I1 I1 I2
IO I1 I1 I1 I2
...

```

```

graph TD
 I0[I0] --> E{E}
 E -- vero --> I1[I1]
 I1 --> E
 E -- falso --> I2[I2]

```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 97

### Analisi

- Poiché la valutazione dell'espressione è effettuata all'inizio, il ciclo stesso può essere eseguito:
  - zero volte
  - un numero finito di volte
  - infinite volte

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 98

### Esempio

```

int x=12;
int y= 1;
while (y<x) y=y*2;

```

- Quanto vale y alla fine dell'esecuzione ?

**y=16**

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 99

### Soluzione

| y  | x  |
|----|----|
| 1  | 12 |
| 2  |    |
| 4  |    |
| 8  |    |
| 16 |    |

```

int x=12;
int y=1;
while (y<x) y=y*2;

```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 100

## Errori frequenti: ripetizioni infinite

- Esempi

```
...
while (x > 0) a = b + c;
```

- poiché il valore di x non viene in alcun caso modificato.....

```
...
while (i > n) {
 ...
 i = i + 1; }
```

- i varia, ma aumenta .....

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 101

## Esempio

```
#include<stdio.h>
main() {
 int x=0;
 while (x==0)
 printf("loop");
}
```

loop infinito

```
looplooplooplooplooploop
looplooplooplooplooploop
looplooplooplooplooploop
looplooplooplooplooploop
looplooplooplooplooploop
looplooplooplooplooploop
looplooplooplooplooploop
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 102

## Convergenza del ciclo

- Assicurarsi, in prima istanza, che nel corpo del while venga alterato il valore di almeno una delle informazioni coinvolte in E
- Assicurarsi, in ogni caso, che, in un numero finito di iterazioni, E diventi falsa
- Assicurarsi che a tutte le informazioni usate nell'espressione E sia stato assegnato un valore prima del while

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 103

## Esempio

- Somma i numeri interi in ingresso finché non viene inserito lo zero

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 104

## Soluzione

```
#include<stdio.h>
main() {
 int num=1;
 int somma=0;
 while (num !=0) {
 printf("Inserisci un numero: ");
 scanf("%d", &num);
 somma=somma+num;
 }
 printf("La somma = %d", somma);
}
```

alla variabile **num** è assegnato il valore **1** per far in modo che il ciclo venga eseguito almeno una volta; ovviamente qualunque numero diverso da zero va bene

## Soluzione alternativa

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

alla variabile **num** è assegnato il primo numero inserito dall'utente fuori dal ciclo

## Relazione tra for e while

```
for (iniz; cond; inc) {
 istruzioni;
}
```

• è equivalente a:

```
iniz;
while (cond) {
 istruzioni;
 inc;
}
```

```
scanf("%d", &n);
while (n!=0) {
 somma=somma+n;
 printf("Inserisci il numero: ");
 scanf("%d", &n);
}
```

## Esempio

```
int n, i, k;
n=4;
for (i=2; i<=n; i++) {
 k=1;
 while (k<10) k=k*i;
 printf("valore di k = %d\n", k);
}
```

Cosa stampa?

valore di k = 16

valore di k = 27

valore di k = 16

### Esempio

```
int n, i, k;
n=4;
for (i=2; i<=n; i++) {
 k=1;
 while (k<10) k=k*i;
 printf("valore di k = %d\n", k);
}
```

Cosa succede se assegno **1** al posto di **2**?

va in loop

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 109

### Esempio

```
int n, i, k;
n=3;
k=7;
while (k>0) {
 for (i=1; i<=n; i++)
 k=k-i;
 printf("Valore di k = %d\n", k);
}
```

Cosa succede se assegno **-7** al posto di **7**?

Valore di k = 1  
Valore di k = -5

non esegue il ciclo **while**

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 110

### Calcolare il massimo comun divisore


- Algoritmo di Euclide:
- Siano dati due numeri x e y > 0
  - Se x==y → MCD(x,y)=x
  - Se x>y → MCD(x-y,y)
  - Se x<y → MCD(x, y-x)
- Esempio
  - 15 9 restituisce 3

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 111

### Soluzione

```
#include<stdio.h>
main() {
 int x, y, mcd;
 printf("Digita due numeri separati da spazio: ");
 scanf("%d %d", &x, &y);
 while (x!=y)
 if (x>y)
 x=x-y;
 else
 y=y-x;
 mcd = x;
 printf("Il MCD =%d", mcd);
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 112



Esercizio


- Siano  $i$  e  $x$  variabili intere. Data l'istruzione:  

```
for(i=1; i<=x; i++)
 printf("%d\n", i*x);
```
- scrivere un programma equivalente usando il while

Se  $x=3$

3  
6  
9

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
113




Soluzione

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

i=1;
while (i<=x) {
 printf("%d\n", i*x);
 i=i+1;
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
114



Sono equivalenti?

```
int x=1, y=1;
while (x>0) {
 x=x-y;
 y=y/2;
}
```


```
int x, y=1;
for (x=1; x>0; x=x-y)
 y=y/2;
```

NO!

loop

$x=0, y=0$

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
115




Sono equivalenti!

```
int x=1, y=1;
while (x>0) {
 x=x-y;
 y=y/2;
}
```

```
int x, y=1;
for (x=1; x>0;) {
 x=x-y;
 y=y/2;
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT
116



## for - while

- Entrambi controllano la condizione di terminazione all'inizio del ciclo
- Impiegare un while invece di un for, o viceversa, dipende largamente dalle preferenze personali ...


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    117



## Istruzioni cicliche

- for
- while
- do-while


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    118



## do-while

- for e while controllano la condizione di terminazione all'inizio del ciclo
- do-while controlla la condizione al termine di ogni iterazione
  - consente di eseguire un ciclo condizionale da 1 ad infinite volte

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    119



## do-while

- Sintassi
 

```
do
 istruzione
while (espressione);
```
- Semantica
  - espressione deve essere valutata ogni volta dopo aver eseguito istruzione
  - se la valutazione di espressione è vera, allora il ciclo viene ripetuto, cioè si riesegue istruzione, altrimenti si esce dal ciclo

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    120

### do-while: diagramma di flusso

```

I0;
do {
 I1;
} while (E);
I2;

```

• le possibili sequenze sono:

I0 I1 I2  
 I0 I1 I1 I2  
 I0 I1 I1 I1 I2  
 ...

```

graph TD
 I0[I0] --> I1[I1]
 I1 --> E{E}
 E -- vero --> I1
 E -- falso --> I2[I2]

```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 121

### Cosa stampa?

```

#include<stdio.h>
main() {
 int n=12, k=5;
 do {
 n=n/2;
 k=k-1;
 } while (n>=k);
 printf("%d %d\n", n, k);
}

```

**1 2**

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 122

### Perché?

```

int n=12, k=5;
do {
 n=n/2;
 k=k-1;
} while (n>=k);
printf("%d %d\n", n, k);

```

**n** ~~12~~ ~~6~~ ~~3~~ **1**

**k** ~~5~~ ~~4~~ ~~3~~ **2**

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 123

### Esempio


```

int num;
do {
 scanf("%d", &num);
} while (num>10);

```

Legge numeri finché  
non viene inserito un  
numero **<=10**


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 124



## Riprendiamo l'esempio

- Somma numeri interi in ingresso finché non viene inserito lo zero


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    125



## Soluzione con do-while

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

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    126




## Relazione tra do-while e while

```
do {
 istruzioni;
} while (espressione);
```

- .... è equivalente a ??

```
istruzioni;
while (espressione) {
 istruzioni;
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    127



## Esercizio

- Scrivere un programma che legge un carattere ed aspetta che l'utente digiti s oppure n. In questo caso stampa OK

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio    Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT    128



## Problema???

```
#include<stdio.h>
main() {
 char car;
 do {
 printf("Rispondi s oppure n: ");
 scanf("%c", &car);
 } while ((car != 's') && (car != 'n'));
 printf("OK");
}
```

## Teorema di Bohm e Jacopini

- Problema: quante e quali strutture di controllo occorrono per descrivere un qualsiasi tipo di algoritmo?
- Teorema di Bohm e Jacopini:
- qualsiasi algoritmo può essere descritto usando solo tre strutture di controllo:
  - una di sequenza
  - una di decisione
  - una di ciclo

## Operatore ternario

- L'operatore ternario è una forma compatta di if-else
- Sintassi  
espressione ? espressione1 : espressione2;
- Semantica
- Se espressione è vera, viene valutata solo espressione1, altrimenti viene valutata espressione2

## Esempio

- Assegnare il minimo tra i valori di y e z a x
- ```
if (y < z)
    x = y;
else
    x = z;
```
- è equivalente a
 $x = (y < z) ? y : z;$

Stile di programmazione

- Evitare programmi prolissi
- Non usare variabili più del necessario
- Memorizzare i risultati di computazioni intermedie quando tali risultati devono essere riusati

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 133

Esempio

- Programma che calcola il rapporto tra la somma e la differenza di due variabili e lo memorizza in r

prolisso	semplice
somma = m+n ;	
diff = m-n ;	r = (m+n) / (m-n) ;
r = somma/diff ;	

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 134

Esempio

- Se un'espressione compare in più punti, è meglio memorizzarla in una variabile piuttosto che ricalcolarla

s = (m+n) + 5* (m+n) * (m+n) ;	sum = m+n ;
t = (m+n) / (m-n) ;	s = sum +5* sum * sum ;
...	t = sum / (m-n) ;
u = 3* (m+n) ;	...
	u = 3* sum ;

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 135

Esempio

- m e n non vengono alterate nel corpo del ciclo

↓

sup = m-n+1 ; while ((i < m-n+1) && (ris < max)) ris = ris * i ; 	Evita di ricalcolare sempre l'espressione m-n+1
--	--

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 136

break

- break consente l'uscita immediata da una istruzione switch, while, for, do-while
- Esempio

```
...
for (i=1; ; i++) {
    if (i==4) break;
    printf("%d\n", i);
}
printf("dopo il break");
```

1
2
3
dopo il break

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 137

break

```
...
for (i=1; ; i++) {
    if (i==4) break;
    printf("%d\n", i);
}
printf("dopo il break");
```

- Meglio evitare questo tipo di implementazione
- Come la si può riscrivere?

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 138

continue

- continue, a differenza di break, si applica ai cicli (while, for, do-while), ma non allo switch
- Interrompe l'esecuzione di un ciclo, ma anziché uscire dal ciclo definitivamente, determina l'esecuzione della successiva iterazione
 - non è molto usata


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 139

Esempio di continue

```
int i=0;
while(i<=3) {
    i=i+1;
    printf("ciao\n");
    continue;
    printf("istr1 dopo continue\n");
    printf("istr2 dopo continue\n");
}
```

ciao
ciao
ciao
ciao


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 140



Esercizio

- Scrivere un programma che legge numeri finché non viene inserito lo zero, e conta quanti numeri pari e dispari sono stati inseriti
- Esempio
input : 1 16 23 2 7 5 0 → dispari=4
pari=2


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 141



Scomponiamo il problema

- Leggere un numero *num*
- Incrementare pari o dispari in base al valore di *num*
- Ripetere finché non leggo zero


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 142



Leggere un numero

```
printf("Inserisci un numero: ");
scanf("%d", &num);
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 143



Incrementare pari o dispari

```
if (num%2 == 0)
    pari = pari+1;
else
    dispari = dispari+1;
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 144

Ciclo

```
do {
    printf("Inserisci un numero: ");
    scanf("%d", &num);
    if (num%2 == 0)
        pari=pari+1;
    else
        dispari=dispari+1;
} while (num != 0);
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 145

Programma completo

```
#include<stdio.h>
main() {
    int num; int dispari=0, pari=-1;
    do {
        printf("Inserisci un numero: ");
        scanf("%d", &num);
        if (num%2 == 0)
            pari=pari+1;
        else
            dispari=dispari+1;
    } while (num != 0);
    printf("Pari=%d e Dispari=%d", pari, dispari);
}
```

lo zero non va contato

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 146

Esercizio

- Scrivere un programma che riceve in input un numero naturale n ed un carattere car e stampa un triangolo rettangolo di altezza e base n , utilizzando il simbolo car
- Esempio

$n = 5$ e $car = \&$	$n = 3$ e $car = \$$
&	\$
&&	\$\$
&&&	\$\$\$
&&&&	
&&&&&	

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 147

Scomponiamo il problema

- Stampare la riga i -esima con car e andare a capo

```
for (j=1; j<=i; j++)
    printf("%c", car);
printf("\n");
```

Se $i=1$, stampa &
...
Se $i=4$, stampa &&&&

&
&&
&&&
&&&&
&&&&&

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 148

... ora dobbiamo stampare tutte n le righe

```
for (i=1; i<=n; i++) {
    for (j=1; j<=i; j++)
        printf("%c", car);
    printf("\n");
}
```











Programma completo


```
#include<stdio.h>
main() {
    int n, i, j; char car;
    printf("Inserisci un numero ed un carattere: ");
    scanf("%d %c", &n, &car);
    for (i=1; i<=n; i++) {
        for (j=1; j<=i; j++)
            printf("%c", car);
        printf("\n");
    }
}
```

Fibonacci

- Il problema di Fibonacci, o problema dei conigli, consiste nel determinare quante coppie di conigli ci saranno dopo N mesi, nelle seguenti ipotesi:
 - al mese 0 c'è una coppia di conigli neonati
 - un coniglio diventa fertile dopo un mese dalla nascita
 - ogni coppia di conigli fertile genera ogni mese una nuova coppia di conigli
 - non c'è mortalità di conigli
- Sviluppare un programma che risolve il problema di Fibonacci

Fibonacci

		Coppie
Mese 0		1
Mese 1		1
Mese 2	 + 	2
Mese 3	2  + 	3
Mese 4	3  + 2 	5
Mese 5	5  + 3 	8



Soluzione


- Sequenza

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144,...

- Ogni nuovo numero rappresenta la somma dei due numeri che lo precedono

$\text{fib}(0) = 1$
 $\text{fib}(1) = 1$
 $\text{fib}(n) = \text{fib}(n-1) + \text{fib}(n-2)$


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 153



Programma

```
#include <stdio.h>
main() {
    int n, f0, f1, i, prossimo;
    f0=1; f1=1;
    printf("Inserisci n: ");
    scanf("%d", &n);
    for (i=1; i<n; i++) {
        prossimo=f0+f1;
        f0=f1;
        f1=prossimo;
    }
    printf("Fibonacci di %d= %d", n, f1);
}
```


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 154



Numero magico

- Generare un numero a caso e chiedere all'utente un numero fino a quando non è uguale a quello generato casualmente

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 155



Numeri casuali

- `rand()` definita in `stdlib.h`
- per la generazione di numeri pseudo-casuali nell'intervallo compreso tra 0 e `RAND_MAX` (costante definita in `stdlib.h`)

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 156

Soluzione

```
#include <stdio.h>
#include <stdlib.h>
main() {
    int guess, magic;
    magic = rand();
    do {
        printf("Indovina il numero magico: ");
        scanf("%d", &guess);
        if (guess == magic)
            printf("Bravo!!\n");
        else
            printf("Sbagliato. Riprova.\n");
    } while (magic != guess);
}
```

genera un numero casuale

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 157

Altra soluzione

```
#include <stdio.h>
#include <stdlib.h>
main() {
    int guess, magic;
    magic = rand() % 10;
    do {
        printf("Indovina il numero magico: ");
        scanf("%d", &guess);
        if (guess == magic)
            printf("Bravo!!\n");
        else
            printf("Sbagliato. Riprova.\n");
    } while (magic != guess);
}
```

...così è più semplice indovinare

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 158

Approfondimenti ...

- I numeri pseudo-casuali ottenuti per via algoritmica, in realtà, casuali non lo sono affatto: se non si inizializza l'algoritmo con un opportuno seme, la sequenza di valori risulta essere sempre la stessa. La funzione utilizzata allo scopo è:
 srand(seme) definita in stdlib.h
- Perché sia efficace, il seme deve poter variare. Si usa perciò invocarla così:
 srand(time(NULL));
- "time" restituisce l'ora corrente ed è perciò differente ogni volta che la si invoca

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 159

Soluzione

```
#include <stdio.h>
#include <stdlib.h>
main() {
    int guess, magic;
    srand(time(NULL));
    magic = rand() % 10;
    do {
        printf("Indovina il numero magico: ");
        scanf("%d", &guess);
        if (guess == magic)
            printf("Bravo!!\n");
        else
            printf("Sbagliato. Riprova.\n");
    } while (magic != guess);
}
```

inizializza il generatore di numeri casuali

genera un numero casuale

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio
Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 160

Cosa stampa ?

```
#include<stdio.h>
main( ) {
    int product=5, x=5;
    product *= ++x;
    printf("product = %d, x = %d", product, x);
}
```

product = 30, x = 6

Cosa stampa ?

```
#include<stdio.h>
main( ) {
    int product=5, x=5;
    product *= x++;
    printf("product = %d, x = %d", product, x);
}
```

product = 25, x = 6

Cosa stampano ?

```
main() {
    int result=5, x=5;
    result = ++x + x ;
    printf("result = %d, x = %d", result, x);
}
```

result = 12, x = 6

```
main() {
    int result=5, x=5;
    result = x++ + x ;
    printf("result = %d, x = %d", result, x);
}
```

result = 10, x = 6

Crittografia

- Una azienda vuole trasmettere dei dati sulla linea telefonica, ma i suoi responsabili sono preoccupati dal fatto che i propri telefoni potrebbero essere spiati. Tutti i loro dati sono trasmessi come interi di quattro cifre. Vi hanno quindi chiesto di scrivere un programma che crittografi i loro dati in modo che possano essere trasmessi con maggior sicurezza. Il vostro programma dovrà leggere un intero di 4 cifre e crittografarlo nel seguente modo: sostituite ogni cifra con la somma di quella cifra + 7 modulo 10. In seguito, scambiare la prima cifra con la terza e la seconda cifra con la quarta. Visualizzate quindi l'intero crittografato. Scrivere poi un programma che riceve in input un intero crittografato di 4 cifre e provvede a decifrarlo, in modo da ricostruire il numero originale

- Esempio:

• originale = 3219 → crittografato = 8609

Osservazione

• $-7\%10 = -7$

• $-17\%10 = -7$

originale --> crittografato

```
#include <stdio.h>
main() {
    int num_originale, num_critto, prima, seconda, terza, quarta;

    printf("Inserisci il numero da crittografare: ");
    scanf("%d", &num_originale);

    prima = num_originale/1000;
    seconda = (num_originale%1000)/100;
    terza = (num_originale%100)/10;
    quarta = num_originale%10;

    prima = (prima + 7) %10;
    seconda = (seconda + 7) %10;
    terza = (terza + 7) %10;
    quarta = (quarta + 7) %10;
    num_critto = terza*1000+quarta*100+prima*10+seconda;
    printf("Il numero crittografato = %d", num_critto);
}
```

crittografato --> originale

```
#include <stdio.h>
main() {
    int num_originale, num_critto, prima, seconda, terza, quarta;
    printf("Inserisci il numero crittografato: ");
    scanf("%d", &num_critto);
    prima = num_critto/1000;
    seconda = (num_critto%1000)/100;
    terza = (num_critto%100)/10;
    quarta = num_critto%10;
    prima = (prima - 7) %10;
    if (prima < 0) prima = 10 + prima;
    seconda = (seconda - 7) %10;
    if (seconda < 0) seconda = 10 + seconda;
    terza = (terza - 7) %10;
    if (terza < 0) terza = 10 + terza;
    quarta = (quarta - 7) %10;
    if (quarta < 0) quarta = 10 + quarta;
    num_originale = terza*1000+quarta*100+prima*10+seconda;
    printf("Il numero originale= %d", num_originale);
}
```

Stelline

- Scrivere un programma che visualizzi il seguente disegno

```

      *
    * * *
  * * * * *
* * * * * * *
  * * * * *
    * * *
      *
```

Stelline

9x9

				*				
			*	*	*			
		*	*	*	*	*		
	*	*	*	*	*	*	*	
*	*	*	*	*	*	*	*	*
	*	*	*	*	*	*	*	
		*	*	*	*	*		
			*	*	*			
				*				

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 169

Soluzione

```

#include <stdio.h>
main() {
  int i, j, stelle=1, spazi=4;
  for (i=1; i<=9; i++) {
    for (j=1; j<=spazi; j++) printf(" ");
    for (j=1; j<=stelle; j++) printf("*");
    for (j=1; j<=spazi; j++) printf(" ");
    if (i<5) {
      stelle += 2;  spazi -= 1;
    }
    else {
      stelle -= 2;  spazi += 1;
    }
    printf("\n");
  }
}

```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 170

Altra soluzione

```

#include <stdio.h>
main() {
  int i, j;
  for (i=0; i<=4; i++) {
    for (j=0; j<4-i; j++) printf(" ");
    for (j=5-i; j<= 5+i; j++ ) printf("*");
    for (j=6+i; j<=8; j++) printf(" ");
    printf("\n");
  }
  for (i=3; i>=0; i--) {
    for (j=0; j<4-i; j++) printf(" ");
    for (j=5-i; j<=5+i; j++ ) printf("*");
    for (j=6+i; j<=8; j++) printf(" ");
    printf("\n");
  }
}

```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 171

Numero palindromo

- Un palindromo è un numero che, da sinistra a destra o da destra verso sinistra, si legge nello stesso modo. Scrivere un programma che legga in input un intero e determini se si tratta o meno di un palindromo
- Esempio:

12321 ➔ palindromo
 12345 ➔ no

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 172

Soluzione ...

```
#include <stdio.h>
#include <math.h>
main( ) {
    int num, temp, bd, cifre, uguali=1;
    printf("Inserisci un numero: "); scanf("%d", &num);
    temp=num;  cifre=1;
    /* conto il numero di cifre */
    while (temp>9) {
        temp=temp/10;  cifre++;
    }
    bd=pow(10,cifre-1);
    while (uguali && num>9) {
        uguali = ((num/bd) == (num%10));
        num = (num%bd)/10;  cifre -=2;
        bd= pow(10,cifre-1);
    }
}
```

... soluzione

```
if (uguali)
    printf("numero palindromo\n");
else
    printf("numero non palindromo\n");

} /* chiude il main */
```

Cosa stampa il seguente programma?

```
#include<stdio.h>
main( ) {
    int i, j, k;
    for (i=1; i<=5; i++) {
        for (j=1; j<=3; j++) {
            for (k=1; k<=4; k++)
                printf("*");
            printf("\n");
        }
    }
}
```

Soluzione

```
****
****
****
****
****
****
****
****
****
****
****
****
****
****
****
```

15 righe, ogni riga ha 4 *