

  
UNIVERSITÀ  
DEGLI STUDI DI BARI  
ALDO MORO

Corso di Laurea in Informatica (Track B) - A.A. 2018/2019

# Laboratorio di Informatica

## Linguaggio C

(Parte 1)

docente: Veronica Rossano  
[veronica.rossano@uniba.it](mailto:veronica.rossano@uniba.it)

Sotto a cura del dott. Claudio Muro

## Linguaggio C

- Perchè il linguaggio C ?
  - C è piccolo (numero limitato di istruzioni)
  - Abbastanza semplice da imparare
  - E' efficiente
  - Caratterizzato da un buon livello di portabilità



Sotto a cura del dott. Claudio Muro

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

## Linguaggio C

- C
  - Evoluzione di due precedenti linguaggi di programmazione, BCPL e B (by Ritchie)
  - Usato per scrivere i moderni sistemi operativi
    - Il sistema operativo UNIX è scritto in C
- Standardizzazione
  - Inizialmente esistevano diverse varianti del C, ma non erano compatibili tra di loro. È stato formato un comitato per creare una definizione "non ambigua, e machine-independent"
  - Standard creato nel 1989, aggiornato nel 1999 (la versione più comune, detta C-99) riaggiornato nel 2011



Sotto a cura del dott. Claudio Muro

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

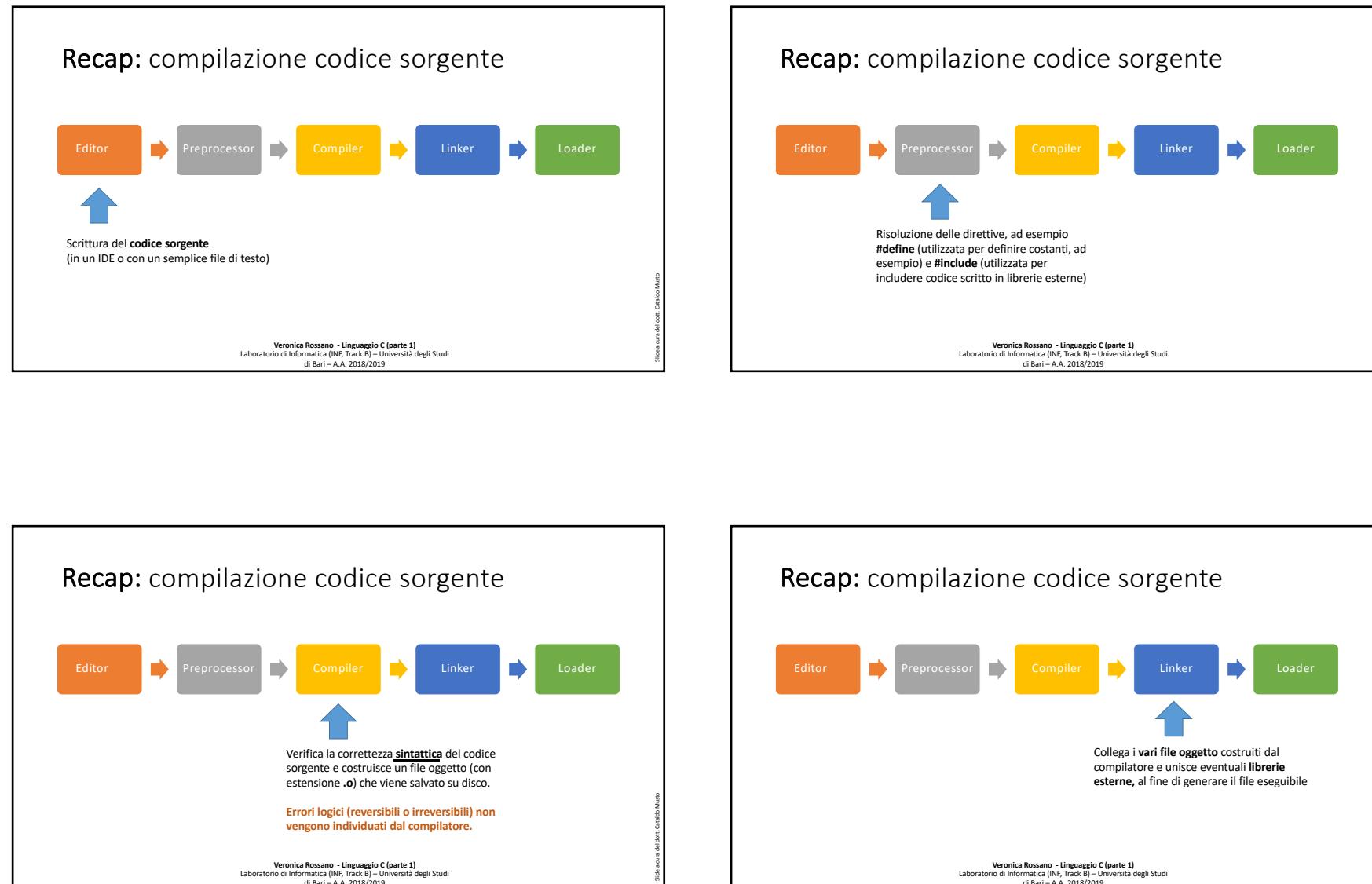
## Recap: compilazione codice sorgente

```

graph LR
    Editor[Editor] --> Preprocessor[Preprocessor]
    Preprocessor --> Compiler[Compiler]
    Compiler --> Linker[Linker]
    Linker --> Loader[Loader]
  
```

Sotto a cura del dott. Claudio Muro

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019



## Recap: compilazione codice sorgente



Carica in memoria e lancia  
**l'eseguibile compilato.** Il processo è  
preso in carico dalla CPU che esegue  
seguenzialmente le istruzioni ed  
eventualmente alloca della memoria  
per creare variabili, file su disco, etc.

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Slide 4 curata da: Carlo Muto

## Linguaggio C: il primo programma

```

1 #include <stdio.h>
3 int main()
4 {
5     int integer1;
6     int integer2;
7     int sum;
8
9     printf("Enter value: ");
10    scanf("%d",&integer1);
11
12    printf("Enter value: ");
13    scanf("%d",&integer2);
14
15    sum = integer1 + integer2;
16    printf("Sum: %d\n", sum);
17
18    return 0;
19 }
  
```

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Slide 4 curata da: Carlo Muto

## Linguaggio C: il primo programma

```

1 #include <stdio.h>
3 int main()
4 {
5     int integer1;
6     int integer2;
7     int sum;
8
9     printf("Enter value: ");
10    scanf("%d",&integer1);
11
12    printf("Enter value: ");
13    scanf("%d",&integer2);
14
15    sum = integer1 + integer2;
16    printf("Sum: %d\n", sum);
17
18    return 0;
19 }
  
```

**Riga 1:** direttive al preprocessore.  
Aggiunge le funzioni per gestire i flussi di input/output

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Slide 4 curata da: Carlo Muto

## Linguaggio C: il primo programma

```

1 #include <stdio.h>
3 int main()
4 {
5     int integer1;
6     int integer2;
7     int sum;
8
9     printf("Enter value: ");
10    scanf("%d",&integer1);
11
12    printf("Enter value: ");
13    scanf("%d",&integer2);
14
15    sum = integer1 + integer2;
16    printf("Sum: %d\n", sum);
17
18    return 0;
19 }
  
```

**Riga 1:** direttive al preprocessore.  
Aggiunge le funzioni per gestire i flussi di input/output

**Riga 3:** il main è la funzione principale

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Slide 4 curata da: Carlo Muto

## Linguaggio C: il primo programma

```

1 #include <stdio.h>
3 int main()
{
5     int integer1;
6     int integer2;
7     int sum;
8
9     printf("Enter value: ");
10    scanf("%d",&integer1);
11
12    printf("Enter value: ");
13    scanf("%d",&integer2);
14
15    sum = integer1 + integer2;
16    printf("Sum: %d\n", sum);
17
18    return 0;
19 }

```

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Riga 1: direttive al preprocessore.  
Aggiunge le funzioni per gestire i flussi di input/output  
  
Riga 3: il main è la funzione principale  
  
Riga 5-7: dichiariamo variabili di tipo intero



Sotto a cura dei docenti Carlo Muto

## Linguaggio C: il primo programma

```

1 #include <stdio.h>
3 int main()
{
5     int integer1;
6     int integer2;
7     int sum;
8
9     printf("Enter value: ");
10    scanf("%d",&integer1);
11
12    printf("Enter value: ");
13    scanf("%d",&integer2);
14
15    sum = integer1 + integer2;
16    printf("Sum: %d\n", sum);
17
18    return 0;
19 }

```

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Riga 1: direttive al preprocessore.  
Aggiunge le funzioni per gestire i flussi di input/output  
  
Riga 3: il main è la funzione principale  
  
Riga 5-7: dichiariamo variabili di tipo intero  
  
Riga 9-10: stampa di una stringa, lettura di un valore e memorizzazione in una variabile.



Sotto a cura dei docenti Carlo Muto

## Linguaggio C: il primo programma

```

1 #include <stdio.h>
3 int main()
{
5     int integer1;
6     int integer2;
7     int sum;
8
9     printf("Enter value: ");
10    scanf("%d",&integer1);
11
12    printf("Enter value: ");
13    scanf("%d",&integer2);
14
15    sum = integer1 + integer2;
16    printf("Sum: %d\n", sum);
17
18    return 0;
19 }

```

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Riga 1: direttive al preprocessore.  
Aggiunge le funzioni per gestire i flussi di input/output  
  
Riga 3: il main è la funzione principale  
  
Riga 5-7: dichiariamo variabili di tipo intero  
  
Riga 9-10: stampa di una stringa, lettura di un valore e memorizzazione in una variabile.  
Perché utilizziamo &?



Sotto a cura dei docenti Carlo Muto

## Linguaggio C: il primo programma

```

1 #include <stdio.h>
3 int main()
{
5     int integer1;
6     int integer2;
7     int sum;
8
9     printf("Enter value: ");
10    scanf("%d",&integer1);
11
12    printf("Enter value: ");
13    scanf("%d",&integer2);
14
15    sum = integer1 + integer2;
16    printf("Sum: %d\n", sum);
17
18    return 0;
19 }

```

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Riga 1: direttive al preprocessore.  
Aggiunge le funzioni per gestire i flussi di input/output  
  
Riga 3: il main è la funzione principale  
  
Riga 5-7: dichiariamo variabili di tipo intero  
  
Riga 9-10: stampa di una stringa, lettura di un valore e memorizzazione in una variabile. Ricordiamo che una variabile è un nome che identifica una locazione di memoria, quindi serve l'operatore & per referenziare l'indirizzo di quella variabile



Sotto a cura dei docenti Carlo Muto

## Linguaggio C: il primo programma

```

1  #include <stdio.h>
3  int main()
4  {
5      int integer1;
6      int integer2;
7      int sum;
8
9      printf("Enter value: ");
10     scanf("%d",&integer1);
11
12     printf("Enter value: ");
13     scanf("%d",&integer2);
14
15     sum = integer1 + integer2;
16     printf("Sum: %d\n", sum);
17
18     return 0;
19 }
```

Sotto a cura del dott. Carlo Muto

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi di Bari – A.A. 2018/2019

**Riga 1:** direttive al preprocessore.  
Aggiunge le funzioni per gestire i flussi di input/output

**Riga 3:** il main è la funzione principale

**Riga 5-7:** dichiariamo variabili di tipo intero

**Riga 9-10:** stampa di una stringa, lettura di un valore e memorizzazione in una variabile. **Ricordiamo che una variabile è un nome che identifica una locazione di memoria, quindi serve l'operatore & per referenziare l'indirizzo di quella variabile**

**Riga 15-17:** somma aritmetica, assegnazione di un valore a una nuova variabile e stampa del valore.

Sotto a cura del dott. Carlo Muto

## Linguaggio C: cenni sulla memoria

Ogni variabile è identificata attraverso tre elementi. Quali?

- Nome
- Tipo
- Valore

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi di Bari – A.A. 2018/2019

Sotto a cura del dott. Carlo Muto

## Linguaggio C: cenni sulla memoria

Ogni variabile è identificata attraverso tre elementi. Quali?

Sotto a cura del dott. Carlo Muto

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi di Bari – A.A. 2018/2019

## Linguaggio C: cenni sulla memoria

Ogni variabile è identificata attraverso tre elementi. Quali?

- Nome
- Tipo
- Valore

Quando dichiariamo una variabile è importante inizializzarla. Perché?

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi di Bari – A.A. 2018/2019

Sotto a cura del dott. Carlo Muto

## Linguaggio C: cenni sulla memoria

`integer1` 45  
`integer2` 72

Nome  
Tipo-Valore

`integer1` 45  
`integer2` 72  
`sum` 117

Nome  
Tipo-Valore

Ogni variabile è identificata attraverso tre elementi. Quali?

- Nome
- Tipo
- Valore

Quando dichiariamo una variabile è importante inizializzarla. Perché?

- Perché un nome di variabile è un alias di una locazione di memoria
- Inizializzando la variabile sovrascriviamo il valore memorizzato in quella locazione di memoria

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi di Bari – A.A. 2018/2019

Slide a cura del dott. Claudio Muto

26/02/19

## Linguaggio C: cenni sulla memoria

`integer1` 45  
`integer2` 72

Nome  
Tipo-Valore

`integer1` 45  
`integer2` 72  
`sum` 117

Nome  
Tipo-Valore

- Il tipo di dato determina **il range e la tipologia di valori** che una variabile può assumere
  - La scelta del tipo di dato più corretto per una variabile è una importante scelta progettuale in fase di risoluzione del problema e codifica dell'algoritmo
  - Ad esempio le variabili di tipo “**short int**” possono assumere valori tra -32768 e 32767 (65536 valori in tutto)
  - Perchè esattamente **65536?**

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi di Bari – A.A. 2018/2019

Slide a cura del dott. Claudio Muto

26/02/19

**NON SARANNO TROPPI**

**TIPI DI DATO?**

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi di Bari – A.A. 2018/2019

Slide a cura del dott. Claudio Muto

26/02/19

## Linguaggio C: tipi di dato

Tipo di dato	Dimensione in Byte (Macchina a 32 bit)	Dimensione in Byte (Macchina a 64 bit)	Range (Macchina a 64 bit)	
char	1	1	-128	127
unsigned char	1	1	0	255
short	2	2	-32768	32767
unsigned short	2	2	0	65535
int	4	4	-2147483648	2147483647
unsigned int	4	4	0	4294967295
long	4	8	-9223372036854775808	9223372036854775808
unsigned long	4	8	0	1844674407370951615
float	4	4	....	....
double	8	8	....	....
Long double	12	16	....	....

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Sede a cura del docente: Cicaldo Muro

## Linguaggio C: tipi di dato

Verificate da soli!

```
#include <stdio.h>

int main(void) {
    char charVariable;
    unsigned char unsignedCharVariable;    short shortVariable;
    unsigned short unsignedShortVariable;   int integerVariable;      unsigned int unsignedIntVariable;
    long longVariable;                     unsigned long unsignedLongVariable;
    float floatVariable;                  double doubleVariable;   long double longDoubleVariable;

    printf("Size of char: %u \n", sizeof(charVariable));
    printf("Size of unsigned char: %u \n", sizeof(unsignedCharVariable));
    printf("Size of short: %u \n", sizeof(shortVariable));
    printf("Size of unsigned short: %u \n", sizeof(unsignedShortVariable));
    printf("Size of integer: %u \n", sizeof(integerVariable));
    printf("Size of unsigned integer: %u \n", sizeof(unsignedIntVariable));
    printf("Size of long: %u \n", sizeof(longVariable));
    printf("Size of unsigned long: %u \n", sizeof(unsignedLongVariable));
    printf("Size of float: %u \n", sizeof(floatVariable));
    printf("Size of double: %u \n", sizeof(doubleVariable));
    printf("Size of long double: %u \n", sizeof(longDoubleVariable));
}
```

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Sede a cura del docente: Cicaldo Muro

## Linguaggio C: tipi di dato

Verificate da soli!

```
#include <stdio.h>

int main(void) {
    char charVariable;
    unsigned char unsignedCharVariable;    short shortVariable;
    unsigned short unsignedShortVariable;   int integerVariable;      unsigned int unsignedIntVariable;
    long longVariable;                     unsigned long unsignedLongVariable;
    float floatVariable;                  double doubleVariable;   long double longDoubleVariable;

    printf("Size of char: %u \n", sizeof(charVariable));
    printf("Size of unsigned char: %u \n", sizeof(unsignedCharVariable));
    printf("Size of short: %u \n", sizeof(shortVariable));
    printf("Size of unsigned short: %u \n", sizeof(unsignedShortVariable));
    printf("Size of integer: %u \n", sizeof(integerVariable));
    printf("Size of unsigned integer: %u \n", sizeof(unsignedIntVariable));
    printf("Size of long: %u \n", sizeof(longVariable));
    printf("Size of unsigned long: %u \n", sizeof(unsignedLongVariable));
    printf("Size of float: %u \n", sizeof(floatVariable));
    printf("Size of double: %u \n", sizeof(doubleVariable));
    printf("Size of long double: %u \n", sizeof(longDoubleVariable));
}
```

**L'operatore  
sizeof( )  
restituisce la  
dimensione di una  
variabile**

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Sede a cura del docente: Cicaldo Muro

**Programming  
is thinking, not typing**

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Sede a cura del docente: Cicaldo Muro

## Problem Solving

- **Prima** di scrivere un programma
  - Comprendere a **fondo il problema (analisi)**
  - Pianificare con cura un **approccio per risolverlo** (approccio **top-down, bottom-up**)
  - Produrre una soluzione in pseudo-codice o con i flow-chart

Programming  
is thinking, not typing

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Sotto a cura del dott. Carlo Muto

## Problem Solving

- **Prima** di scrivere un programma
  - Comprendere a **fondo il problema (analisi)**
  - Pianificare con cura un **approccio per risolverlo** (approccio **top-down, bottom-up**)
  - Produrre una soluzione in pseudo-codice o con i flow-chart
- **Mentre** scrivete un programma
  - Individuate quali “building blocks” sono disponibili (**riuso** del codice)
  - La maggior parte dei programmi segue una struttura “standard”
    - **Definizione e inizializzazione** delle variabili (acquisizione dell’input)
    - **Elaborazione** dei dati
    - **Visualizzazione** dei risultati (output)

Programming  
is thinking, not typing

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Sotto a cura del dott. Carlo Muto

## Programmazione Strutturata

- **Teorema di Bohm e Jacopini:** tutti i programmi possono essere scritti usando tre strutture di controllo fondamentali
  - **Sequenza:**
    - Nativa nel C. I programmi vengono eseguiti sequenzialmente per default
  - **Selezione:**
    - Il C ne ha tre tipi: **if, if..else, switch**
  - **Iterazione:**
    - Il C ne ha tre tipi: **while, do...while e for**

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Sotto a cura del dott. Carlo Muto

## Struttura di Selezione

Usata per scegliere tra diverse alternative

Istruzione	Pseudocodice	Traduzione in C
<b>if</b>		<b>Scrivere un programma che stampi un messaggio differente a seconda che il voto ricevuto sia maggiore o minore di 18</b>
<b>if..else</b>		

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Sotto a cura del dott. Carlo Muto

## Struttura di Selezione

Usata per scegliere tra diverse alternative

Istruzione	Pseudocodice	Traduzione in C
<b>if</b>	Se il voto dello studente è maggiore di 18 Stampa "Promosso"	
<b>if...else</b>	Se il voto dello studente è maggiore di 18 Stampa "Promosso" Altrimenti Stampa "Bocciato"	

Sotto a cura dei doc. Giacomo Muto

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi di Bari – A.A. 2018/2019

## Struttura di Selezione

Usata per scegliere tra diverse alternative

Istruzione	Pseudocodice	Traduzione in C
<b>if</b>	Se il voto dello studente è maggiore di 18 Stampa "Promosso"	if ( voto >= 18 ) puts( "Promosso\n" );
<b>if...else</b>	Se il voto dello studente è maggiore di 18 Stampa "Promosso" Altrimenti Stampa "Bocciato"	if ( voto >= 18 ) { puts( "Promosso\n" ); } else puts( "Bocciato\n" );

Sotto a cura dei doc. Giacomo Muto

**Importante:** l'indentazione rende il programma **più leggibile**

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi di Bari – A.A. 2018/2019

## Struttura di Selezione

Usata per scegliere tra diverse alternative

Istruzione	Pseudocodice	Traduzione in C
<b>switch</b>	Se il voto dello studente è maggiore di 18 Stampa "Promosso" Altrimenti Stampa "Bocciato"	

↑

Lo pseudocodice è analogo a quella dell'istruzione **if...else**  
ma si esprime in modo diverso.

Sotto a cura del docente Giacomo Muto

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi di Bari – A.A. 2018/2019

## Struttura di Selezione

Usata per scegliere tra diverse alternative

Istruzione	Pseudocodice	Traduzione in C
<b>switch</b>	Se il voto dello studente è maggiore di 18 Stampa "Promosso" Altrimenti Stampa "Bocciato"	

↑

Lo pseudocodice è analogo a quella dell'istruzione **if...else**  
ma si esprime in modo diverso.

```
switch(voto) {
    case 0: case 1: case 2:
    case 3: case 4: case 5:
    case 6: case 7: case 8:
    case 9: case 10: case 11:
    case 12: case 13: case 14:
    case 15: case 16: case 17:
        puts("Bocciato\n");
        break;
    default:
        puts("Promosso\n");
}
```

Sotto a cura del docente Giacomo Muto

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi di Bari – A.A. 2018/2019

### Struttura di Selezione

Usata per scegliere tra diverse alternative

Istruzione	Pseudocodice	Traduzione in C
switch	Se il voto dello studente è maggiore di 18 Stampa "Promosso" Altrimenti Stampa "Bocciato"	<pre>switch(voto) {     case 0: case 1: case 2:     case 3: case 4: case 5:     case 6: case 7: case 8:     case 9: case 10: case 11:     case 12: case 13: case 14:     case 15: case 16: case 17:         puts("Bocciato\n");         break;     default:         puts("Promosso\n"); }</pre>

Lo pseudocodice è analogo a quella dell'istruzione `if...else` ma si esprime in modo diverso.

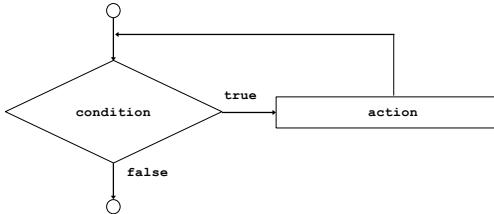
E' una tipologia di **istruzione molto utile** se dobbiamo valutare **diverse alternative di tipo categorico** (non numerico!) – es: città di nascita, provincia di residenza, voto di un esame (in lettere, tipo 'A', 'B', 'C'), gruppo sanguigno, etc.

Sotto a cura del dott. Carlo Muto

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi di Bari – A.A. 2018/2019

### Struttura di Iterazione - while

Utilizzata per esprimere operazioni che si ripetono finchè una determinata condizione resta vera



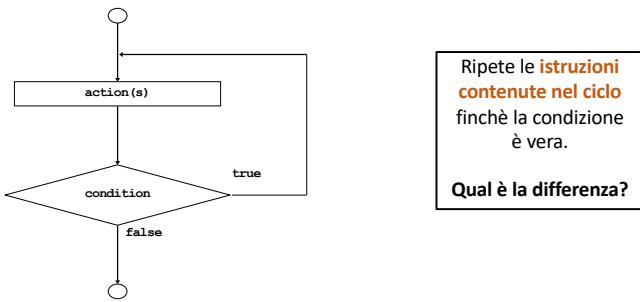
Ripete le **istruzioni contenute nel ciclo** finchè la condizione è vera.

Sotto a cura del dott. Carlo Muto

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi di Bari – A.A. 2018/2019

### Struttura di Iterazione – do .. while

Utilizzata per esprimere operazioni che si ripetono finchè una determinata condizione resta vera.



Ripete le **istruzioni contenute nel ciclo** finchè la condizione è vera.

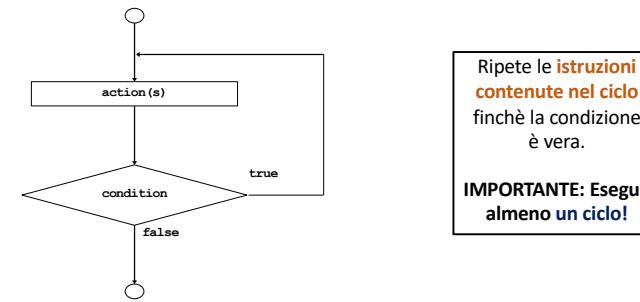
Qual è la differenza?

Sotto a cura del dott. Carlo Muto

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi di Bari – A.A. 2018/2019

### Struttura di Iterazione – do .. while

Utilizzata per esprimere operazioni che si ripetono finchè una determinata condizione resta vera.



Ripete le **istruzioni contenute nel ciclo** finchè la condizione è vera.

**IMPORTANTE: Esegue almeno un ciclo!**

Sotto a cura del dott. Carlo Muto

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi di Bari – A.A. 2018/2019

## Struttura di Iterazione – for

Utilizzata per esprimere operazioni che si ripetono finchè una determinata condizione resta vera.

Segue la stessa struttura del **while**,  
ma utilizza una variabile contatore che  
**viene incrementata dopo il blocco**  
**delle istruzioni (action)**

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Sotto a cura dei doc. Carlo Muto

## Struttura di Iterazione

Utilizzata per esprimere operazioni che si ripetono finchè una determinata condizione resta vera.

Istruzione	Pseudocodice	Traduzione in C
while		<b>Scrivere un programma che conteggi il costo totale dei prodotti in un carrello. Sappiamo che il carrello può contenere esattamente cinque prodotti.</b>
do..while		
for		

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Sotto a cura dei doc. Carlo Muto

## Struttura di Iterazione

Utilizzata per esprimere operazioni che si ripetono finchè una determinata condizione resta vera.

Istruzione	Pseudocodice	Traduzione in C
while	<b>finché(numero_prodotti&lt;5)</b> leggi costo aggiungi il costo al totale	<b>Scrivere un programma che conteggi il costo totale dei prodotti in un carrello. Sappiamo che il carrello può contenere esattamente cinque prodotti.</b>
do..while		
for		

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Sotto a cura dei doc. Carlo Muto

## Struttura di Iterazione

Utilizzata per esprimere operazioni che si ripetono finchè una determinata condizione resta vera.

Istruzione	Pseudocodice	Traduzione in C
while	<b>finché(numero_prodotti&lt;5)</b> leggi costo aggiungi il costo al totale	<b>Scrivere un programma che conteggi il costo totale dei prodotti in un carrello. Sappiamo che il carrello può contenere esattamente cinque prodotti.</b>
do..while	<b>ripeti</b> leggi costo aggiungi il costo al totale <b>finché(numero_prodotti&lt;5)</b>	
for	<b>finché(numero_prodotti&lt;5)</b> leggi costo aggiungi il costo al totale	

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Sotto a cura dei doc. Carlo Muto

## Struttura di Iterazione

Utilizzata per esprimere operazioni che si ripetono finché una determinata condizione resta vera.

Istruzione	Pseudocodice	Traduzione in C
while	<b>finché(numero_prodotto&lt;5)</b> leggi costo aggiungi il costo al totale	
do..while	<b>ripeti</b> leggi costo aggiungi il costo al totale <b>finché(numero_prodotto&lt;5)</b>	
for	<b>finché(numero_prodotto&lt;5)</b> leggi costo aggiungi il costo al totale	

**Scrivere un programma che conteggi il costo totale dei prodotti in un carrello. Sappiamo che il carrello può contenere esattamente cinque prodotti.**

Sotto a cura del dott. Giacomo Muto

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi di Bari – A.A. 2018/2019

## Struttura di Iterazione

Utilizzata per esprimere operazioni che si ripetono finché una determinata condizione resta vera.

Istruzione	Pseudocodice	Traduzione in C
while	<b>finché(numero_prodotto&lt;5)</b> leggi costo aggiungi il costo al totale	
do..while	<b>ripeti</b> leggi costo aggiungi il costo al totale <b>finché(numero_prodotto&lt;5)</b>	
for	<b>finché(numero_prodotto&lt;5)</b> leggi costo aggiungi il costo al totale	

**Come lo rendiamo in linguaggio C?**

Sotto a cura del dott. Giacomo Muto

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi di Bari – A.A. 2018/2019

## Struttura di Iterazione

Utilizzata per esprimere operazioni che si ripetono finché una determinata condizione resta vera.

Istruzione	Pseudocodice	Traduzione in C
while	<b>finché(numero_prodotto&lt;=5)</b> leggi costo aggiungi il costo al totale	<code>while(products&lt;=5){     scanf("%d",&amp;costo);     totale = totale + costo;     products++; }</code>
do..while	<b>ripeti</b> leggi costo aggiungi il costo al totale <b>finché(numero_prodotto&lt;=5)</b>	<code>do {     scanf("%d",&amp;costo);     totale = totale + costo;     products++; } while(products&lt;=5);</code>
for	<b>finché(numero_prodotto&lt;=5)</b> leggi costo aggiungi il costo al totale	<code>for(products=0; products&lt;=5; products++) {     scanf("%d",&amp;costo);     totale = totale + costo; }</code>

Sotto a cura del dott. Giacomo Muto

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi di Bari – A.A. 2018/2019

## Problema 1.1

**Scrivere un programma che conteggi il costo totale dei prodotti in un carrello. Non sappiamo quanti prodotti può contenere il carrello.**

**Input?**  
Costo dei singoli prodotti

**Output?**  
Costo totale della spesa

**Quale tipologia di istruzioni ci serve? Perché?**  
Struttura di **iterazione**. Perché il programma effettua una operazione (ciclica) comandando il costo dei prodotti nel carrello. **Non conosciamo a priori il numero di prodotti da inserire.**

Sotto a cura del dott. Giacomo Muto

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi di Bari – A.A. 2018/2019

## Problema 1.1

Scrivere un programma che conteggi il costo totale dei prodotti in un carrello. Non sappiamo quanti prodotti può contenere il carrello.

**Input?**

Costo dei singoli prodotti

**Output?**

Costo totale della spesa

**Quale tipologia di istruzioni ci serve? Perché?**

Struttura di **iterazione**. Perché il programma effettua una operazione (ciclica) sommando il costo dei prodotti nel carrello. **Non conosciamo a priori il numero di prodotti da inserire.**

**E' necessario introdurre il concetto di iterazione non controllata**

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Sotto a cura del dott. Claudio Muto

## Problema 1.1

Scrivere un programma che conteggi il costo totale dei prodotti in un carrello. Non sappiamo quanti prodotti può contenere il carrello.

**Input?**

Costo dei singoli prodotti

**Output?**

Costo totale della spesa

**Quale tipologia di istruzioni ci serve? Perché?**

Struttura di **iterazione**. Perché il programma effettua una operazione (ciclica) sommando il costo dei prodotti nel carrello. **Non conosciamo a priori il numero di prodotti da inserire.**

**E' necessario introdurre il concetto di iterazione non controllata!**

Non si utilizza un contatore fisso, si utilizza un valore sentinella

Se il valore letto è uguale al valore sentinella, esce dal ciclo.

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Sotto a cura del dott. Claudio Muto

## Struttura di Iterazione (non controllata)

Utilizzata per esprimere operazioni che si ripetono finchè una determinata condizione resta vera.

Istruzione	Pseudocodice	Traduzione in C
while	finchè(sentinella=true) leggi costo aggiungi il costo al totale	while(costo != -1) { scanf("%d",&costo); totale = totale + costo; }
Do..while	ripeti leggi costo aggiungi il costo al totale finchè(sentinella=true)	do { scanf("%d",&costo); totale = totale + costo; } while(costo != -1);
for	finchè(sentinella=true) leggi costo aggiungi il costo al totale	for(costo = 0; costo != -1;){ scanf("%d",&costo); totale = totale + costo; }

Sotto a cura del dott. Claudio Muto

## Struttura di Iterazione (non controllata)

Utilizzata per esprimere operazioni che si ripetono finchè una determinata condizione resta vera.

Istruzione	Pseudocodice	Traduzione in C	
while	finchè(sentinella=true) leggi costo aggiungi il costo al totale	while(costo != -1){ scanf("%d",&costo); totale = totale + costo; }	Il valore sentinella va nella condizione. Si tendono ad utilizzare valori non validi per quella variabile
Do..while	ripeti leggi costo aggiungi il costo al totale finchè(sentinella=true)	do { scanf("%d",&costo); totale = totale + costo; } while(costo != -1);	
for	finchè(sentinella=true) leggi costo aggiungi il costo al totale	for(costo = 0; costo != -1;){ scanf("%d",&costo); totale = totale + costo; }	

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Sotto a cura del dott. Claudio Muto

## Struttura di Iterazione (non controllata)

Utilizzata per esprimere operazioni che si ripetono finché una determinata condizione resta vera.

Istruzione	Pseudocodice	Traduzione in C
<code>while</code>	<code>finché(sentinella=true) leggi costo aggiungi il costo al totale</code>	<code>while(costo != -1){     scanf(%d,&amp;costo);     totale = totale + costo; }</code>
<code>do...while</code>	<code>ripeti leggi costo aggiungi il costo al totale finché(sentinella=true)</code>	<code>do {     scanf(%d,&amp;costo);     totale = totale + costo; } while(costo != -1);</code>
<code>for</code>	<code>finché(sentinella=true) leggi costo aggiungi il costo al totale</code>	<code>for(costo = 0; costo != -1; {     scanf(%d,&amp;costo);     totale = totale + costo; })</code>

Slide a cura del dott. Carlo Muto

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

## Struttura di Iterazione (non controllata)

Utilizzata per esprimere operazioni che si ripetono finché una determinata condizione resta vera.

Istruzione	Pseudocodice	Traduzione in C
<code>while</code>	<code>finché(sentinella=true) leggi costo aggiungi il costo al totale</code>	<code>while(costo != -1){     scanf(%d,&amp;costo);     totale = totale + costo; }</code>
<code>do...while</code>	<code>ripeti leggi costo aggiungi il costo al totale finché(sentinella=true)</code>	<code>do {     scanf(%d,&amp;costo);     totale = totale + costo; } while(costo != -1);</code>
<code>for</code>	<code>finché(sentinella=true) leggi costo aggiungi il costo al totale</code>	<code>for(costo = 0; costo != -1; {     scanf(%d,&amp;costo);     totale = totale + costo; })</code>

**Suggerimento:** `while` e `do...while` si tendono a preferire per le iterazioni non controllate, mentre il `for` è la struttura più semplice da adottare quando sappiamo a priori il numero di iterazioni da eseguire è noto.

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Slide a cura del dott. Carlo Muto

### Problema 1.1

Scrivere un programma che conteggi il costo totale dei prodotti in un carrello. Non sappiamo quanti prodotti può contenere il carrello.

**Proviamo a codificare una soluzione su Repl.it**

Slide a cura del dott. Carlo Muto

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

### Problema 1.1

Scrivere un programma che conteggi il costo totale dei prodotti in un carrello. Non sappiamo quanti prodotti può contenere il carrello.

**Proviamo a codificare una soluzione su Repl.it**



**(My Repls → Simbolo '+' in basso → Linguaggio C)**

Slide a cura del dott. Carlo Muto

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

## Soluzione (con `while` + valore sentinella)

```

1  #include "stdio.h"
2
3  int main(void) {
4      int costo = 0;           // variabile di tipo int
5      int totale = 0;
6
7      while(costo != -1) {
8          printf("Inserisci il costo del prodotto: ");
9          scanf("%d", &costo);
10         totale = totale + costo;
11     }
12
13     printf("Totale: %d", totale);
14 }
```

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Sotto a cura dei doc. Carlo Muto

## Soluzione (con `while` + valore sentinella)

```

1  #include "stdio.h"
2
3  int main(void) {
4      int costo = 0;           // variabile di tipo int
5      int totale = 0;
6
7      while(costo != -1) {
8          printf("Inserisci il costo del prodotto: ");
9          scanf("%d", &costo);
10         totale = totale + costo;
11     }
12
13     printf("Totale: %d", totale);
14 }
```

**E' una soluzione corretta?**

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Sotto a cura dei doc. Carlo Muto

## Soluzione (con `while` + valore sentinella)

```

1  #include "stdio.h"
2
3  int main(void) {
4      int costo = 0;           // variabile di tipo int
5      int totale = 0;
6
7      while(costo != -1) {
8          printf("Inserisci il costo del prodotto: ");
9          scanf("%d", &costo);
10         totale = totale + costo;
11     }
12
13     printf("Totale: %d", totale);
14 }
```

gcc version 4.6.3  
Inserisci il costo del prodotto: 2  
Inserisci il costo del prodotto: 3  
Inserisci il costo del prodotto: 6  
Inserisci il costo del prodotto: -1  
Totale: 10

No, perché il valore sentinella viene comunque aggiunto al totale

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Sotto a cura dei doc. Carlo Muto

**Soluzione (con `while` + valore sentinella)**

Bisogna applicare i principi della programmazione difensiva: il programma deve funzionare correttamente anche davanti a input non corretti

```

1  #include "stdio.h"
2
3  int main(void) {
4      int costo = 0;           // variabile di tipo int
5      int totale = 0;
6
7      while(costo != -1) {
8          printf("Inserisci il costo del prodotto: ");
9          scanf("%d", &costo);
10         totale = totale + costo;
11     }
12
13     printf("Totale: %d", totale);
14 }
```

sentinella viene comunque aggiunto al totale

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Sotto a cura dei doc. Carlo Muto

## Soluzione (con `while` + valore sentinella)

```

1 #include <stdio.h>
2
3 int main(void) {
4     int costo = 0;           // variabile di tipo int
5     int totale = 0;
6
7     printf("Inserisci il costo del prodotto: ");
8     scanf("%d", &costo);
9
10    while(costo != -1) {
11        totale = totale + costo;
12        printf("Inserisci il costo del prodotto: ");
13        scanf("%d", &costo);
14    }
15    printf("Totale: %d", totale);
16 }
```

Programma  
(parzialmente)  
Corretto

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Sotto a cura del dott. Cataldo Musto

## Soluzione (con `while` + valore sentinella)

```

1 #include <stdio.h>
2
3 int main(void) {
4     int costo = 0;           // variabile di tipo int
5     int totale = 0;
6
7     printf("Inserisci il costo del prodotto: ");
8     scanf("%d", &costo);
9
10    while(costo != -1) {
11        totale = totale + costo;
12        printf("Inserisci il costo del prodotto: ");
13        scanf("%d", &costo);
14    }
15    printf("Totale: %d", totale);
16 }
```

Se il primo input inserito  
è -1 il programma non  
entra nel ciclo e stampa  
zero: CORRETTO

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Sotto a cura del dott. Cataldo Musto

## Soluzione (con `while` + valore sentinella)

```

1 #include <stdio.h>
2
3 int main(void) {
4     int costo = 0;           // variabile di tipo int
5     int totale = 0;
6
7     printf("Inserisci il costo del prodotto: ");
8     scanf("%d", &costo);
9
10    while(costo != -1) {
11        totale = totale + costo;
12        printf("Inserisci il costo del prodotto: ");
13        scanf("%d", &costo);
14    }
15    printf("Totale: %d", totale);
16 }
```

Mancano ancora dei  
controlli sul valore  
inserito per la variabile  
«costo»

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Sotto a cura del dott. Cataldo Musto

## Domande?

Laboratorio di Informatica  
docente: Cataldo Musto  
[cataldo.musto@uniba.it](mailto:cataldo.musto@uniba.it)



Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Sotto a cura del dott. Cataldo Musto

## Esercitazione 0

- **Obiettivo:** utilizzare i concetti base di programmazione strutturata, combinando **istruzioni di iterazione e selezione** per risolvere problemi complessi. Dimostrare anche di avere conoscenza relativa alla **tipizzazione**.
- *Assignment su Repl.it*
  - [Esercitazione di autovalutazione](#).

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Sotto a cura dei dott. Claudio Muto

## Esercitazione 0

- Il Body Mass Index (BMI) è calcolato come il rapporto tra il peso e il quadrato dell'altezza (espresso in metri) di un individuo. Gli individui sovrappeso hanno un BMI superiore a 25
- $$\text{BMI} = \text{peso} / (\text{altezza\_in\_metri} * \text{altezza\_in\_metri})$$
1. Scrivere un algoritmo che generi random età, altezza e peso di dieci individui e calcoli il BMI di ciascuno di essi e lo mostri sullo schermo.
  2. L'algoritmo deve calcolare il massimo di altezza e peso e la media del BMI , distinguendo over 40 e under 40
  3. L'algoritmo deve anche stampare in output i valori massimi di altezza e peso (per le due categorie)
  4. L'algoritmo deve stampare un messaggio sullo schermo del tipo «il campione è mediamente sovrappeso» oppure «il campione è mediamente sottopeso» a seconda che il BMI medio sia maggiore o minore di 25 (sia per gli under 40 che per gli over).

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Sotto a cura dei dott. Claudio Muto

## Esercitazione 0 (cont.)

- **INPUT**
  - Età, Altezza, Peso, generati random per 10 individui
    - Attenzione a generare valori in un intervallo corretto!
- **OUTPUT**
  - Altezza massima over 40, Altezza massima under 40
  - Peso massimo over 40, Peso massimo under 40
  - BMI medio over 40, BMI medio under 40
- **NOTE**
  - **Porre attenzione allo stile di programmazione:** nomenclatura, costanti simboliche e indentazione
  - Porre attenzione a scrivere commenti significativi
  - Porre attenzione alla **programmazione difensiva:** il programma funziona anche nei casi "limite" ?

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Sotto a cura dei dott. Claudio Muto

## Esercitazione 0 (cont.)

### • SUGGERIMENTI

Per la generazione di un numero random in un range tra MIN e MAX utilizzare il seguente frammento di codice

```
#include <stdlib.h> // libreria che include la funzione rand()
#include <time.h> // libreria per randomizzare il seed
int seed = time(NULL);
srand(seed);
int randomValue = rand() % (MAX-MIN+1) + MIN;
```

Se vogliamo generare un numero tra 10 e 30 la funzione sarà  
`int randomValue = rand() % 21 + 10;`

Perchè l'operatore % ci restituisce un valore tra 0 e 20, cui sommeremo 10 che è il valore minimo possibile.

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Sotto a cura dei dott. Claudio Muto

## Esercitazione 0 (cont.)

- **ESTENSIONI**

Modificare l'algoritmo utilizzando un valore sentinella invece di un ciclo con valore predefinito

**Svolgete l'assignment su Repl.it**

**Termine ultimo: Sabato 2 Marzo**

Veronica Rossano - Linguaggio C (parte 1)  
Laboratorio di Informatica (INF, Track B) – Università degli Studi  
di Bari – A.A. 2018/2019

Foto: G. Susto - Ufficio stampa USMB