

# **Sperimentazioni di Fisica I**

## **mod. A – Laboratorio 3**

### **Diagrammi di Flusso**

*Dipartimento di Fisica e Astronomia “G. Galilei”,  
Università degli Studi di Padova*

# The programming process

Il processo di scrittura di un programma si articola nei seguenti punti:

1. **Definire** quello che un **programma deve fare**;
2. **Visualizzare** la fase di esecuzione del programma;
3. Utilizzare tools che permettono creare un **modello del programma**;
4. **Verificare** che il **modello** sia libero da errori di logica
5. **Scrivere** il programma sorgente
6. **Compilare** il programma
7. **Correggere** eventuali **errori** riscontrati in fase di compilazione
8. **Linkare** il programma e creare l'eseguibile
9. **Eseguire** il programma e **verificarne il funzionamento** con parametri di test
10. Correggere eventuali errori rilevati in fase di esecuzione.  
Se necessario ripetere i punti **4-10**.
11. **Validare** i **risultati** del programma

È molto **importante** iniziare sempre con una corretta **pianificazione**.

# Definire scopo del programma

## Example

Scrivere un programma che, dato il raggio, calcoli la lunghezza della circonferenza e l'area del cerchio.

**Purpose** Calcolare la **lunghezza della circonferenza** e **l'area del cerchio**.

**Input** Il **raggio** del cerchio,  $r$ .

**Processing** Calcola la lunghezza della circonferenza  $2\pi \cdot r$  e l'area del cerchio,  $\pi \cdot r^2$ .

**Output** Stampa un messaggio con area e lunghezza della circonferenza.

# Visualizzare funzionalità del programma

Prima di scrivere il programma bisogna **visualizzare il suo funzionamento nella propria mente**: immaginare quello che viene mostrato sullo schermo del computer mentre il programma funziona.

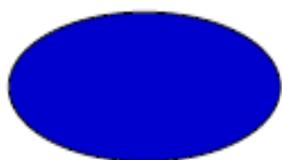
## Possibile soluzione:

```
Inserire il raggio del cerchio: 1
```

```
La lunghezza della circonferenza e': 6.2832
```

```
L'area del cerchio e': 3.1416
```

# Flowchart symbols



START o END del programma



Direzione del flusso logico

---

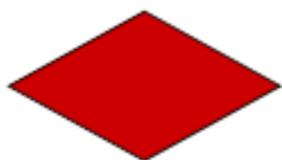


INPUT oppure OUTPUT



Operazioni (esempio calcoli aritmetici)

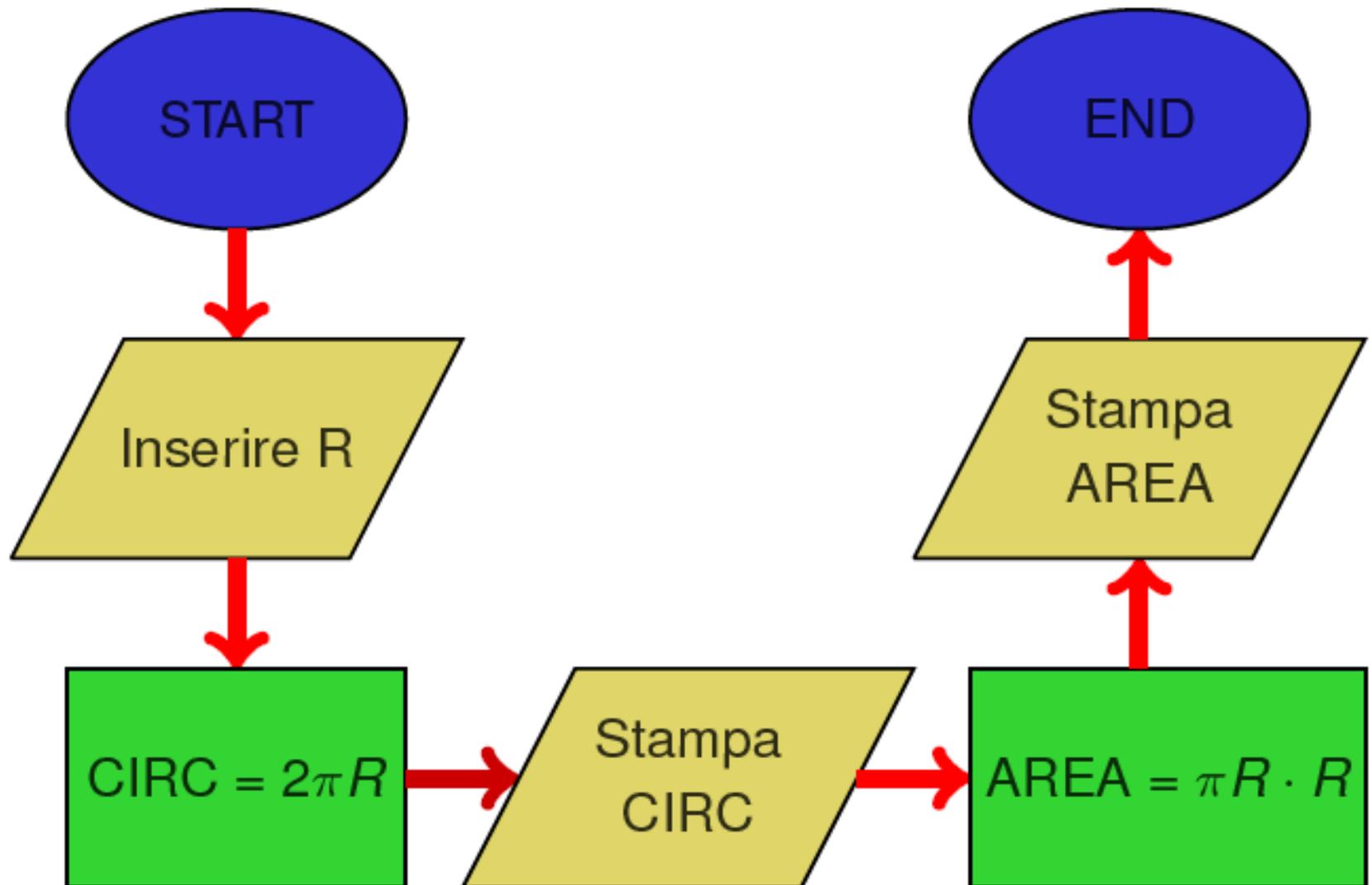
---



Decision or Branch

# Esempio di flowchart

Ecco il **flowchart** del programma esempio.



# Pseudocode

Incrocio tra il linguaggio umano e un linguaggio di programmazione.  
Il programmatore si focalizza sugli step logici del programma senza preoccuparsi della sintassi o dei dettagli della programmazione.

## High Level Pseudocode

- Prendi il raggio
- Calcola la lunghezza della circonferenza
- Mostra la lunghezza della circonferenza
- Calcola l'area del cerchio
- Mostra l'area del cerchio

## Low Level Pseudocode

- Output "Inserisci raggio del cerchio:"
- Input raggio
- Calcola lunghezza circonferenza come  $2 \cdot \pi \cdot r$  e salvala nella variabile circonferenza
- Output "Lunghezza della circonferenza:"
- Output il valore della circonferenza
- Calcola l'area del cerchio come  $\pi \cdot r^2$  e salvala nella variabile area
- Output "Area del cerchio:"
- Output il valore dell'area

# La scrittura del codice

Una volta verificato il modello del programma (flowchart o pseudocode) e corretto, si passa alla scrittura del codice in **C++**.

```
#include <iostream>
#include <cmath>
int main( )
{
    using namespace std;
    cout << "Inserire Raggio: ";
    double raggio;
    cin >> raggio;
    double circ = 2. * M_PI * raggio;
    cout << "Circonferenza: " << circ << endl;
    double area = M_PI * raggio * raggio;
    cout << "Area: " << area << endl;
    return 0;
}
```

# Comandi per Compilare

**Comando generale:**

```
g++
```

**Per utilizzare le features dello standard C++11 :**

- dalla versione GCC 4.7 in poi:

```
g++ -std=c++11
```

- per le versioni precedenti:

```
g++ -std=c++0x
```