

Laboratorio 2

Per ogni esercizio:

Prima di scrivere il codice C++, individuare le variabili e i parametri delle funzioni (e il loro tipo!) che saranno necessari durante l'esecuzione. **Commentare** il codice per facilitare la comprensione dello stesso.

Esercizio 1

Scrivere una **funzione** che, per quanto riguarda un corpo che si muove di moto uniformemente accelerato,

- prende in input tramite i suoi parametri (quindi non deve usare cin!):
 - il tempo trascorso,
 - l'accelerazione,
 - la velocità iniziale,
 - la posizione iniziale,
- e restituisce con return (quindi non deve usare cout!) la posizione attuale.

Scrivere il **main** in modo tale che stampi la posizione di un corpo che si muove di moto uniformemente accelerato ogni **deltaT** secondi per **n** volte utilizzando un ciclo **while** che chiama ripetutamente la funzione descritta prima. cin e cout devono essere utilizzati solo nel **main**. Esempio di esecuzione:

```
Dammi l'accelerazione, la velocita' e la posizione iniziali: 4 6 8
Ogni quanti secondi vuoi stampare la posizione? 2
Per quante volte vuoi stampare la posizione? 10
```

```
Posizione: 8 al tempo 0 secondi
Posizione: 28 al tempo 2 secondi
.....
Posizione: 928 al tempo 20 secondi
```

Esercizio 2

Scrivere una **funzione** che stabilisca se un punto (dato con le sue coordinate x e y), è all'interno di un cerchio (dato con le coordinate del suo centro e la lunghezza del raggio). La **funzione** deve prendere i dati necessari come parametri (non deve usare cin!) e deve restituire un **bool** con return (non deve usare cout!). Usare la funzione nel **main**.

Esercizio 3

Scrivere una **funzione** che visualizzi il triangolo di Floyd dato il numero di righe da stampare come parametro (la funzione non deve usare cin!). Il triangolo di Floyd contiene tutti gli numeri interi a partire da 1 organizzandoli in righe successive in maniera tale che ogni riga contenga un numero in più della precedente. Le prime 5 righe sono:

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

Utilizzare la funzione nel **main**.

Esercizio 4

Scrivere una **funzione** ricorsiva (una funzione che chiama se stessa) che restituisca l'*n*-esimo numero di Fibonacci secondo il seguente schema di ricorrenza:

$$F(n) = \begin{cases} n, & n \leq 1 \\ F(n-1) + F(n-2), & n > 1 \end{cases}$$

dove n deve essere un parametro della funzione (la funzione non deve usare cin e cout!).

Utilizzare la funzione nel **main**, stampando oltre $F(n)$, anche il numero di chiamate della funzione effettuate per calcolare $F(n)$. A questo scopo utilizzare una **variabile globale** (dichiarata quindi al di fuori di qualunque modulo) che serve da contatore delle chiamate.

Esercizio 5

Scrivere una **funzione** non-ricorsiva (cioè una funzione che non chiama sé stessa) che restituisca l' n -esimo numero di Fibonacci. Anche in questo caso n deve essere un parametro della funzione (la funzione non deve usare cin e cout!).

Suggerimento: usare tre variabili locali alla funzione per memorizzare il k -esimo, il $(k-1)$ -esimo ed il $(k-2)$ -esimo numero di Fibonacci, e aggiornare opportunamente le tre variabili in un ciclo che fa variare k da 2 a n .

Utilizzare la funzione nel **main**.

Esercizio 6

Scrivere una **funzione** che prende come argomento due numeri interi **passati per riferimento** e scambia i valori dei due interi. (Cioè, per esempio, se la funzione viene chiamata con x e y che sono uguali a 2 e 3, allora dopo la chiamata y deve essere uguale a 2 e x deve essere uguale a 3.)

Esercizio 7

Scrivere una **funzione** ricorsiva che, dato un numero intero a e un numero intero n non negativo, calcola la potenza n -esima di a secondo il seguente schema di ricorrenza:

$$a^n = \begin{cases} 1, & n = 0 \\ a \cdot a^{n-1}, & n > 0 \end{cases}$$

Scrivere una seconda **funzione** che utilizzi invece il seguente schema di ricorrenza:

$$a^n = \begin{cases} 1, & \text{se } n = 0 \\ a^{\frac{n}{2}} \cdot a^{\frac{n}{2}}, & \text{se } n \text{ è pari} \\ a \cdot a^{\frac{n}{2}} \cdot a^{\frac{n}{2}}, & \text{se } n \text{ è dispari} \end{cases}$$

Non calcolare due volte la stessa quantità: $a^{\frac{n}{2}}$ appare due volte nei prodotti sopra ma questo non vuole dire che bisogna calcolarlo due volte.

Testare le funzioni nel **main**.

Usare due variabili globali per contare il numero di chiamate necessarie per calcolare 2^{16} con il primo e il secondo schema. Qual è la relazione tra il numero n ed il numero di chiamate ricorsive effettuate nei due casi?