

WEB DEVELOPER

Unità Formativa (UF): Fondamenti di Programmazione

Docente: Massimo PAPA

Titolo argomento: Esercizi sulle funzioni (firma funzioni)

Esercizi sulle funzioni (usare la firma delle funzioni)

Indicazioni sulla consegna	1
Struttura del codice sorgente	1
Testo esercizi	3
Primo Esercizio	3
Secondo Esercizio	3
Terzo Esercizio	3
Quarto Esercizio	3

Indicazioni sulla consegna

Implementare gli algoritmi risolutivi dei seguenti esercizi codificandoli in linguaggio c++. Partire analizzando il problema seguendo la seguente traccia:

- Quali sono gli input del problema?
- Quali sono gli output?
- Suddividi il problema in problemi più semplici e se lo ritieni opportuno descrivi la soluzione di ogni sottoproblema con un flow-chart
- Andrai a rappresentare ogni algoritmo che risolve un sottoproblema come una funzione
- La funzione restituisce un valore? Se sì di che tipo?
- Ogni funzione accetta una lista di parametri formali? Se sì di che tipo?
- Esegui la codifica partendo dalla funzione main, al suo interno vai a richiamare le funzioni che poi andrai successivamente a definire.
- Continua la codifica dichiarando e definendo tutte le funzioni prima della funzione main che richiama tutte le altre. In testa alle funzioni scrivi un commento che descrive la funzione stessa, cosa restituisce e quali sono i parametri formali.
- Per ogni funzione dichiara le eventuali variabili locali una per ogni riga, indicando per ognuna una riga di commento
- Scrivi la sezione dell'inizializzazione variabili
- Verifica la codifica utilizzando input di test, cercando di provare anche i casi limite.

Preferibilmente carica la cartella zippata del progetto DevCpp, un progetto per ogni esercizio.

Attenzione: non utilizzare array o matrici in nessuno dei seguenti esercizi.

Struttura del codice sorgente

Relativamente alle indicazioni di scrittura del codice, utilizza il seguente schema generale:

```
/*  
    Autore:  Nome Cognome  
    Data:   gg/mm/aaaa  
  
    Titolo: Testo esercizio  
  
*/
```

```
#include <iostream>
using namespace std;

//
// Funzioni:
//

/* Funzione: fun
   Template per costruire le funzioni
   Parametri:
       int Param1 -> descrizione Param1
       float Param2 -> descrizione Param2
   Valore di ritorno:
       int -> descrizione valore di ritorno
*/
/*int fun(int param1, float param2) {
    int retValue; // Valore di ritorno della funzione

    return retValue;
}*/

/* Programma principale
   Descrizione sintetica funzionalità
   del programma principale.
*/
int main() {
    // Dichiarazione variabili

    // Inizializzazioni variabili

    // Sezione di input Dati

    // Elaborazione
    // Eventuali sotto processi di Elaborazione
    // ---
    // ---

    // Sezione di output

    return 0;
}
```

Testo esercizi

ATTENZIONE: Risolvere i seguenti esercizi utilizzando le funzioni con, esclusivamente, le variabili globali.

Primo Esercizio

Scrivere una funzione che dati due numeri interi x e y , calcoli la somma di tutti i numeri dispari compresi tra x e y . Costruire un programma che richiami n volte la funzione su n differenti intervalli (n dato in input).

Suggerimento: utilizzare la funzione di generazione di numeri pseudocasuali.

Secondo Esercizio

Si legga una sequenza di numeri. Terminare la lettura quando si incontra un numero dispari. In output si stampa quanti numeri sono stati letti e quanti fra essi sono risultati diversi da zero.

Terzo Esercizio

Scrivi un programma che esegua il calcolo delle soluzioni di un'equazione di secondo grado nella forma $ax^2 + bx + c = 0$. Il programma deve mostrare il seguente menù di scelta:

1. Inserimento coefficienti (a, b, c)
2. Calcolo del discriminante ($\Delta = b^2 - 4ac$)
3. Calcolo delle soluzioni
4. Uscita

Quando si seleziona una voce di calcolo stampare il risultato calcolato.

Ogni voce deve essere gestita da una funzione. ~~Le funzioni devono utilizzare le variabili globali.~~

Quarto Esercizio

Creare una funzione che data l'altezza calcoli in quanto tempo un oggetto cade al livello del suolo. Si faccia in modo che la funzione possa operare con la gravità del pianeta in cui ci si trova (usare una variabile globale).

Si crei una funzione per impostare il pianeta in cui ci si trova:

- passando 0 ci si troverà sulla Terra ($g = 9,81 \text{ m/s}^2$),

- passando 1 ci si troverà sulla Luna ($g = 1,62 \text{ m/s}^2$),
- passando 2 ci si troverà su Marte ($g = 3,71 \text{ m/s}^2$),

passando altri valori ci si troverà sempre sulla Terra.

Creare un programma in cui si chiede all'utente il pianeta in cui ci si trova (puoi implementare un menù di scelta), l'altezza (in metri) . Stampare in output il tempo (in secondi) che impiegherà per arrivare al suolo.

Suggerimento: si utilizzi la seguente formula per calcolare il tempo di caduta dell'oggetto:
 $\text{tempo} = \sqrt{2 \cdot \text{altezza} / g}$ g è l'accelerazione di gravità