

WEB DEVELOPER

Unità Formativa (UF): Fondamenti di Programmazione

Docente: Massimo PAPA

Titolo argomento: Esercizi sulle funzioni passaggio di parametri per riferimento

Esercizi sulle funzioni passaggio di parametri per riferimento

Indicazioni sulla consegna	1
Struttura del codice sorgente	1
Testo esercizi	3
Primo Esercizio	3
Secondo Esercizio	3
Terzo Esercizio	3
Quarto Esercizio	4

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

Primo Esercizio

Creare una funzione che ordini (ordine crescente) il contenuto di due variabili che le sono passate. Successivamente creare un programma che inseriti tre numeri da parte dell'utente li stampi in ordine crescente usando la funzione precedentemente creata.

Esempi: Input: A = 4 B = 2 C = 3 Output: A = 2 B = 3 C = 4
 Input: A = 1 B = 4 C = 3 Output: A = 1 B = 3 C = 4
 Input: A = 3 B = 2 C = 3 Output: A = 2 B = 3 C = 3

Secondo Esercizio

Due persone hanno una quantità di denaro x e y. Richiamare una funzione che "livelli" le quantità di denaro in maniera tale da equilibrare tale quantità tra i due.

Esempio: se la persona A possiede 30€ e la persona B 70€, una volta utilizzata la funzione la persona A e B possiederanno entrambe 50 €.

Terzo Esercizio

Creare una funzione che ricevuti tre importi di denaro sposti gli eventuali debiti (si considerino debiti gli importi negativi) sul primo importo.
Creare un main per testare la funzione.

Esempio:
se gli importi sono 5 -1 -2 gli importi dopo l'applicazione della funzione saranno 2 0 0.
se gli importi sono -3 4 -2 gli importi dopo l'applicazione della funzione saranno -5 4 0.
se gli importi sono 4 -2 -2 gli importi dopo l'applicazione della funzione saranno 0 0 0.
se gli importi sono 6 -4 -3 gli importi dopo l'applicazione della funzione saranno -1 0 0.

Quarto Esercizio

Creare una funzione che ricevuti due interi che rappresentano numeratore e denominatore di una frazione, riduca la frazione ai minimi termini.

Creare poi un programma che legga due frazioni le riduca ai minimi termini e verifichi se abbiano lo stesso denominatore.

Suggerimento: cercare sul web l'algoritmo che semplifichi una frazione ai minimi termini oppure l'algoritmo per calcolare l'M.C.D. tra due numeri.