

WEB DEVELOPER

Unità Formativa (UF): Fondamenti di Programmazione

Docente: Massimo PAPA

Titolo argomento: Esercizi sulle tabelle (3)

Esercizi sulle tabelle (3)

Indicazioni sulla consegna	2
Struttura del codice sorgente	3
Testo esercizi	5
Primo Esercizio	5
Esempio:	5
Secondo Esercizio	5
Terzo Esercizio	5

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, un progetto per ogni esercizio.

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

Una rivendita di libri usati descrive i propri prodotti secondo le seguenti caratteristiche:

- codice numerico (numero intero)
- numero di pagine
- costo

Per questioni commerciali si ha la necessità di raccogliere in un'archivio tutti i libri in vendita e si vuole la stampa di questi secondo l'ordine crescente di costo per pagina.

Secondo Esercizio

Un mobilificio deve gestire la vendita di tavoli prodotti dai diversi mastri artigiani loro fornitori. Ogni tavolo è di una tipologia, ha un numero massimo di persone che possono stare sedute intorno, ha delle dimensioni (altezza da terra, lunghezza e larghezza del piano) in metri.

Si vuole creare un programma che permetta di memorizzare tutte le diverse tipologie di tavolo che il mobilificio vende, tale applicativo deve permettere la stampa dei tavoli a seconda del mastro artigiano inserito nella query di ricerca, ordinata per la superficie media a persona che il tavolo può offrire.

Terzo Esercizio

Un'applicazione di gestione magazzino lancia periodicamente una procedura di aggiornamento dei propri archivi che esegue due operazioni:

1. eliminazione dei prodotti scaduti
2. aggiornamento del prezzo di prodotti di una particolare categoria merceologica

I prodotti scaduti vengono individuati se un'apposito flag è inizializzato a TRUE. Una volta individuati devono essere eliminati e l'intero archivio deve essere compattato in maniera tale che non vi siano posizioni vuote nella tabella. La regola di compattazione deve essere lo "shiftamento in basso".

L'aggiornamento del prezzo avviene chiedendo in input l'incremento/decremento del prezzo e la categoria merceologica interessata.

Una volta eseguite le due funzionalità la procedura deve effettuare la stampa di tutti i prodotti stampando i seguenti dati:

- Codice prodotto
- Categoria merceologica
- Prezzo

La stampa deve essere ordinata per ordine di categoria merceologica crescente e di prezzo decrescente.

Si determini l'appropriata struttura dati necessaria a descrivere il dato trattato.
La parte di input dati può essere codificata direttamente all'interno del codice.