

WEB DEVELOPER

Unità Formativa (UF): Fondamenti di Programmazione

Docente: Massimo PAPA

Titolo argomento: Esercizi sugli array

Esercizi sugli array

Indicazioni sulla consegna	2
Struttura del codice sorgente	3
Testo esercizi	5
Primo Esercizio	5
Secondo Esercizio	5
Terzo Esercizio	5
Quarto Esercizio	5
Quinto Esercizio	5
Sesto Esercizio	5
Settimo Esercizio	5
Ottavo Esercizio	5
Nono Esercizio	6
Decimo Esercizio	6

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

Memorizzare in un array di 100 posti i primi cento numeri naturali in ordine inverso.

Secondo Esercizio

Dopo aver letto e memorizzato N numeri reali in un array, N dato in input, calcolare la somma di quelli negativi e, successivamente, sovrascrivere il loro valore con zero. Stampare la somma calcolata e l'array modificato.

Terzo Esercizio

Leggere un array di interi di N posizioni, N dato in input. Leggere un ulteriore numero intero M e dire quanti numeri memorizzati nell'array sono inferiori e quanti superiori di M .

Quarto Esercizio

Leggere un array di interi di N posizioni, N dato in input. Dire se i valori memorizzati nell'array sono in ordine crescente oppure no.

Quinto Esercizio

Leggere un array di N posizioni, N dato in input. Dire se ogni valore memorizzato nell'array è minore del quadrato del numero nella cella precedente.

Sesto Esercizio

Leggere un array di numeri reali di N posizioni, N dato in input. Stampare il numero che compare più volte all'interno dell'array, qualora ci siano più numeri che compaiono lo stesso numero di volte stampare quello che compare per primo.

Settimo Esercizio

Leggere un array di N interi, N dato in input. Stampare solo i numeri che appaiono nell'array una volta soltanto.

Esempio: se l'array contiene 1, 2, 3, 1, 2, 4 il programma stamperà 3, 4.

Ottavo Esercizio

Leggere un array di N interi, N dato in input. Verificare se c'è almeno un valore presente un numero di volte identico al suo valore.

Nono Esercizio

Leggere un array di N numeri reali, N dato in input. Dire se ogni valore presente nell'array è minore della somma dei numeri presenti nelle celle che lo precedono.

Decimo Esercizio

Leggere un array di N interi, N dato in input. Verificare se contiene tutti valori diversi.