

VERIFICA DI TPSIT SU C

(funzioni ricorsive, puntatori, allocazione dinamica della memoria, struct e file)

ESERCIZIO 1

Scrivete un programma che chieda all'utente di immettere una serie di n numeri interi (che verranno memorizzati in un array) e poi li ordini invocando la funzione `selectionSort`.

Quando le viene dato un array di n elementi, la `selectionSort` deve fare le seguenti cose:

- cercare l'elemento più grande all'interno del array e poi spostarlo nell'ultima posizione del array stesso.
- chiamarsi ricorsivamente.

ESERCIZIO 2

Scrivi un programma in C che esegua le seguenti operazioni su un array di numeri interi utilizzando esclusivamente i puntatori e l'aritmetica dei puntatori:

- Chiedi all'utente quanti numeri vuole inserire, quindi crea un array di n interi
- Scrivi una funzione `riempiArray` che farà inserire all'utente i valori all'interno dell'array. La funzione accetta un puntatore all'array e la sua dimensione.
- Scrivi due funzioni, `trovaMinimo` e `trovaMassimo`, che accettino come parametro un puntatore all'array e la sua dimensione, e restituiscano rispettivamente il valore minimo e massimo.
- Scrivi una funzione `calcolaMedia` che accetti come parametro un puntatore all'array e la sua dimensione, e restituisca la media dei valori dell'array.
- Stampa i valori minimo, massimo e la media calcolati.

Attenzione! L'esercizio verrà annullato se sarà presente nel codice una parentesi quadra

ESERCIZIO 3

Immagina di dover monitorare le temperature giornaliere di una città per analizzare le variazioni stagionali. Ogni giorno, inserisci la temperatura media rilevata. Tuttavia, non sai esattamente quanti giorni continuerà questa raccolta dati, poiché dipenderà dalle condizioni e dalle esigenze di monitoraggio. Il programma deve essere in grado di adattarsi se si decide di aggiungere più valori, ad esempio continuando la raccolta oltre un periodo prestabilito. Alla fine, il programma calcola la temperatura media su tutti i giorni monitorati.

Scrivi un programma in C che:

1. Chieda all'utente quanti giorni di temperature intende inizialmente registrare.
2. Chieda all'utente di riempire l'array con le temperature senza preoccuparsi di quante ne inserirà
3. Calcoli la temperatura media dei giorni e la stampi a schermo.

ESERCIZIO 4

Scrivi un programma in C che implementi una semplice gestione di ricette per un ricettario:

1. Definisci una struct Ricetta con i campi nome (nome della ricetta), tempoPreparazione (tempo di preparazione in minuti), tipoPortata (a per antipasto, p per primo, s per secondo e d per dolce)
2. Chiedi all'utente quante n ricette vuole inserire e crea un array di n struct Ricetta.
3. Salva i dati inseriti in un file di testo chiamato "ricettario.txt".
4. Alla fine, stampa le ricette leggendo dal file.

Attenzione! Le ricette vecchie devono sempre rimanere una volta inserite

Punteggi:

- | | |
|-----------------------------------|----------|
| • Esercizio 1 | 10 punti |
| • Esercizio 2 | 15 punti |
| • Esercizio 3 | 20 punti |
| • Esercizio 4 | 35 punti |
| • Interfacce utente | 10 punti |
| • Correttezza e ordine del codice | 10 punti |

Consegnare tutti i progetti in un file zip (cognome_nome_TPSIT_verificaC.zip)