

CORSO DI FONDAMENTI DI INFORMATICA

Prof. Salvatore Andolina - A.A. 2023/2024

Laboratorio di Programmazione in C – Laboratorio 3

Esercizio 1 – Esami

Scrivere un programma che definisca la struttura di un esame universitario e ne permetta l'inserimento dei dati da riga di comando.

Gli esami vengono rappresentati con:

- nome corso (di tipo stringa)
- cognome docente (di tipo stringa)
- numero di CFU (di tipo intero)
- voto (di tipo intero)

Per rappresentare in blocco i dati dell'esame si usi una struttura dati di tipo `struct`. Scrivere un `main()` che permetta l'inserimento da linea di comando di un elemento di tipo esame e che stampi poi a schermo il contenuto.

Suggerimento: per acquisire una stringa contenente spazi bianchi utilizzare la funzione `fgets()` specificando un buffer opportunamente grande (se si utilizza la funzione `scanf()` l'acquisizione della stringa termina al primo carattere spazio bianco).

Esercizio 2 – Esami 2.0

Modificare il programma precedente per permettere all'utente di inserire un numero prefissato di esami (es. 5), stamparne i dati e calcolare la media pesata dei voti.

Esercizio 3 – Copia dispari

Scrivere un programma che, data una matrice `Mat1` di interi $N \times N$, con N costante opportunamente definita, copia i soli elementi dispari in una nuova matrice `Mat2` della stessa dimensione, senza lasciare posizioni vuote intermedie.

Esempio con $N = 3$:

<u>Mat1</u>	<u>Mat2</u>
1 2 3	1 3 5
4 5 6	7 9 0
7 8 9	0 0 0

Successivamente, stampare a video la matrice `Mat2`, visualizzando gli elementi per righe.

Esercizio 4 – Aritmetica dei puntatori

In questo esercizio dovreste dimostrare le relazioni esistenti tra puntatori e array, utilizzando l'aritmetica dei puntatori per scorrere un array.

L'esercizio è da risolvere un passo dopo l'altro, secondo questa sequenza:

Passo 1 - dichiarare due variabili: un array di interi e un puntatore a intero a cui assegnare il nome dell'array. Dimostrare l'equivalenza tra l'indirizzo dell'array, l'indirizzo del primo elemento dell'array e il contenuto del puntatore stampandoli;

Passo 2 - mostrare come dereferenziando il puntatore si ottiene il valore del primo elemento dell'array;

Passo 3 - stampare l'array non accedendo al suo contenuto tramite gli indici, ma incrementando il valore del puntatore all'array o direttamente il nome dell'array avvalendosi dell'aritmetica dei puntatori;

Esercizio 5 - Fusione di array ordinati con Bubblesort

Scrivere un programma che:

- chiede due volte all'utente di inserire N valori interi, e li salva in due array di dimensione N;
- ordina quindi ciascun array utilizzando il metodo Bubble Sort;
- effettua la fusione dei due array, in modo che l'array risultante mantenga l'ordinamento;
- stampa il vettore dopo la fusione.

Valutare inoltre la complessità dell'algoritmo di ordinamento contando il numero totale di confronti e il numero totale di swap eseguiti e stampare questi valori.

L'algoritmo Bubble Sort confronta ogni coppia di elementi adiacenti del vettore. Se due elementi sono nell'ordine sbagliato, essi vengono invertiti (swap). Come risultato, ad ogni passo l'elemento più grande non ancora ordinato si sposta verso la posizione più alta dell'array. Al passo p, gli ultimi p-1 elementi dell'array saranno nella loro posizione definitiva (non è necessario controllarli di nuovo). L'algoritmo continua a scorrere tutta la lista di elementi finché non vengono più eseguiti scambi, situazione che indica che la lista è ordinata (*qui un'interpretazione visiva dell'algoritmo: <https://youtu.be/lv3vgjM8Pv4?t=51>*).