CORSO DI FONDAMENTI DI INFORMATICA

Prof. Maristella Matera - A.A. 2019 / 2020

Laboratorio di Programmazione in C – Laboratorio 3

Esercizio 1 – Copia dispari

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

Implementare un `main()` che, dopo aver richiamato la funzione implementata, stampa a video la nuova matrice, visualizzando gli elementi per righe.

Esercizio 2 – Copia dispari v. 2.0

Modificare il programma dell'Esercizio 1 in modo che la matrice `Mat1` di interi `NxN`, con `N` costante opportunamente definita, **venga letta da un file di testo chiamato 'matrice.txt' nel formato:**

```
1,2,3
4,5,6
7,8,9
```

Il programma, prima di procedere con la copia, **verifica che la matrice definita nel file abbia le dimensioni richieste.** Se la matrice non è di dimensioni NxN allora deve comparire un messaggio di errore a video e il programma viene terminato.

Come nell'Esercizio 1, la funzione copia i soli elementi dispari in una nuova matrice `Mat2` della stessa dimensione, senza lasciare posizioni vuote intermedie.

Implementare un `main()` che, dopo aver richiamato la funzione implementata, **scriva in un file di testo 'matrice_dispari.txt' la nuova matrice Mat2,** rispettando il formato riportato sopra.

Esercizio 3 – Sudoku

Si realizzi un programma che verifichi la corretta soluzione della griglia di un Sudoku.

Un Sudoku è una griglia 9x9 che rispetta queste caratteristiche:

- tutti i valori devono essere singoli numeri tra 1 e 9
- su ciascuna delle 9 **righe** non devono esserci valori ripetuti
- su ciascuna delle 9 **colonne** non devono esserci valori ripetuti
- in ciascuno dei 9 **blocchi 3x3** che compongono la griglia non devono esserci valori ripetuti

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

Livello 1 – Utilizzare il file `sudoku_template.c` fornito assieme a questo testo come punto di partenza: il file contiene la dichiarazione di due matrici Sudoku da verificare (una

corretta e l'altra sbagliata). Il programma deve verificare la validità di queste due matrici e stampare a video il risultato del controllo. In questo livello il programma va scritto **senza** utilizzare le funzioni (quindi completando solo il `main()` già esistente).

Livello 2 – Modificare il programma già scritto usando le funzioni per dividere il problema assegnato in sottoproblemi.

Livello 3 – Modificare il programma già scritto in modo che la matrice da verificare venga acquisita numero per numero, e che venga eseguito il controllo di validità già in fase di inserimento.

Livello 4 – Modificare il programma già scritto in modo che la matrice da verificare venga letta dal file di testo `sudoku.txt` fornito assieme a questo testo.

Esercizio 4 – Aritmetica dei puntatori

Esercizio che dimostra le relazioni esistenti tra puntatori e array e come si può utilizzare la aritmetica dei puntatori per scorrere un array:

- 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;
- mostrare come dereferenziando il puntatore si ottiene il valore del primo elemento dell'array;
- stampare infine 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;
- eseguire la stampa dell'array tramite una funzione che riceve un puntatore a un array di interi.

Esercizio 5 – Allocazione dinamica (interi)

Scrivere un programma che riceve dall'utente un numero N di interi non definito a priori, li memorizza in un array allocato dinamicamente e infine stampa la somma e la media dei numeri inseriti *(N viene inserito dall'utente prima di inserire i valori).*

Esercizio 6 – Copia di stringhe

Scrivere un programma che dichiari due stringhe di massimo 15 caratteri e ne inizializzi una. Scrivere una funzione che copia il contenuto della stringa inizializzata nella stringa non inizializzata usando l'aritmetica dei puntatori.

Esercizio 7 – Allocazione dinamica (struct)

Definire una struct che rappresenti uno studente. Ogni studente è composto da: nome (stringa), numero di matricola (intero) e percentuale di esami sostenuti e passati (float).

Scrivere un programma che riceve dall'utente un numero N di studenti non definito a priori, li memorizza in un array allocato dinamicamente e infine stampa tutti gli studenti inseriti *(N viene inserito dall'utente prima di inserire i valori).*

Esercizio 8 - 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. N è specificato dall'utente, quindi gli array devono essere allocati dinamicamente;
- 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.

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, a 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 finché non vengono più eseguiti scambi, situazione che indica che la lista è ordinata.

Implementare l'algoritmo scomponendo il problema in sottoproblemi mediante l'uso di funzioni. Creare una funzione per eseguire l'ordinamento bubblesort di un vettore, una funzione per eseguire lo swap di due elementi, e una funzione per fondere due array in un unico vettore.

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.