

PROGRAMMAZIONE 2: SPERIMENTAZIONI

Appello del 24 luglio 2020

Turno 1 (mattino)

Tempo a disposizione: un'ora e trenta minuti

Esercizio 1 di 2

Scrivere un programma che ha come argomento un numero intero (che deve essere compreso tra 1 e 20 inclusi) e crea una lista con tanti nodi in base al numero passato come argomento. Ogni nodo della lista deve contenere un valore casuale (random) compreso tra 0 e 100 (estremi inclusi).

Dopo aver creato la lista tramite una funzione denominata **lista_crea**, scrivere le seguenti funzioni:

- 1) **lista_visualizza**: visualizza la lista in modo *ricorsivo*;
- 2) **lista_pari**: crea una seconda lista contenente solamente i nodi che possiedono valori pari nella prima lista; visualizzare la lista ottenuta tramite la funzione sviluppata nel punto 1); la funzione deve essere *iterativa*;
- 3) **lista_pari_file**: crea un file chiamato "pari.txt" contenente i valori dei nodi nella seconda lista separati da una virgola; la funzione deve essere *iterativa*.

Richiamare ed eventualmente visualizzare i risultati delle funzioni 1), 2), 3) nel main del programma.

Esempio

./a.out 5

1) lista 1: 1 -> 20 -> 3 -> 40 -> 5

2) lista 2: 20 -> 40

3) dati.txt conterrà
20,40

Organizzazione programma

Il programma deve essere organizzato su tre file:

- 1) main.c
- 2) liste.c
- 3) liste.h

Tutti i file devono essere salvati all'interno di una cartella chiamata esercizio_1; la cartella deve essere consegnata in modalità compressa (esercizio_1.zip).

Documentazione del programma

Nel file .h è necessario documentare le funzioni nella forma:

```
/**  
 * Descrizione esaustiva della funzione  
 * @param: ...  
 * @param: ...  
 * @return: ...  
 * @see: ... (se applicabile, altrimenti scrivere un trattino -)  
 */
```

Esercizio 2 di 2

Scrivere un programma che ha come argomento una stringa e crea una lista con tanti nodi quanti sono i caratteri presenti nella stringa.

Dopo aver creato la lista tramite una funzione denominata **lista_crea**, scrivere le seguenti funzioni:

- 1) **lista_visualizza**: visualizza la lista *in modo ricorsivo*;
- 2) **lista_maiuscole**: se è presente tra i nodi della lista un carattere minuscolo da 'a' (codice ASCII 97) a 'z' (codice ASCII 122) lo cancella dalla lista; visualizzare quindi la lista ottenuta tramite la funzione sviluppata nel punto 1); la funzione deve essere *iterativa*;
- 3) **lista_numeri**: restituisce la quantità di numeri da '0' (codice ASCII 48) a '9' (codice ASCII 57) presenti nella lista *in modo ricorsivo*.

Richiamare ed eventualmente visualizzare i risultati delle funzioni 1), 2), 3) nel main del programma.

Esempio

./a.out Anna79!

1) lista 1: A -> n -> n -> a -> 6 -> 2 -> !

2) lista 1 (aggiornata): A -> 6 -> 2 -> !

3) numeri presenti: 2

Organizzazione programma

Il programma deve essere organizzato su tre file:

- 1) main.c
- 2) liste.c
- 3) liste.h

Il programma deve contenere un makefile per essere compilato tramite la funzionalità make.

Tutti i file devono essere salvati all'interno di una cartella chiamata esercizio_2; la cartella deve essere consegnata in modalità compressa (esercizio_2.zip).

Documentazione del programma

Nel file .h è necessario documentare le funzioni nella forma:

```
/**
 * Descrizione esaustiva della funzione
 * @param: ...
 * @param: ...
 * @return: ...
 * @see: ... (se applicabile, altrimenti scrivere un trattino -)
 */
```