Laboratorio di Programmazione 1

Scomporre i programmi

Esercizio su array

Diciamo che un array a[] è "quasi ordinato" se tutti i valori negativi (se esistono) precedono i valori maggiori o uguali di zero (se esistono); si noti che i valori negativi possono comparire in ordine qualsiasi, come pure i valori maggiori o uguali di zero. Ad esempio l'array a[] = {-1, -7, 8, 10, 8, 1} è quasi ordinato, come pure l'array a[] = {-9, -7, -1, -3} e a[] = {13, 9, 0, 21, 33},

mentre l'array a[] = {3, -1, 4, 5} (compare un valore >= 0 prima di uno negativo). Scrivere la funzione int quasi_ordinato(int a[], int n) che, dato in input un array a[] non vuoto e la sua lunghezza n, ritorna 1 se e solo se a[] è quasi ordinato, 0 altrimenti.

Esercizio su array

Realizzare il sottoprogramma somme_prefisse che accetta come parametro un array di interi a[] di lunghezza N (e qualsiasi altro parametro ritenuto necessario); si garantisce che si avrà sempre N >= 1.

La funzione deve restituire un NUOVO array b[], sempre di lunghezza N, contenente le "somme prefisse" di a[].

Specificamente, per ogni j (j=0,…,N-1) i valori di b[j]

si ottengono dalla somma degli elementi a[0] +…+ a[j].

Ad esempio, se a[] = {1,2,1,2,1}, si otterrà b[] = {1,3,4,6,7}.

Si scrivano inoltre il prototipo del sottoprogramma,

la chiamata nel main, e la dichiarazione di tutte le variabili ritenute utili.

(Bonus di 1 punto per la definizione ricorsiva)

Esercizio su LISTE

Dato il seguente tipo

```
struct list {
    int val;
    struct list *next;
};
```

Definire la funzione RICORSIVA

int liste_uguali(struct list * L1, struct list * L2) seguente: la funzione deve restituire 1 se e solo se le liste hanno lo stesso numero di nodi, e l'i-esimo nodo della lista L1 contiene lo stesso valore (campo val) dell'i-esimo nodo della lista L2.

DA TEMA ESAME

Esercizio 1 [11 punti] File ESA 18062020 B 1.c
Completare tutte le funzioni dichiarate ma non definite nel programma ESA_18062020 B 1.c per la gestione di una lista concatenata di numeri interi, es.

3 -> 6 -> 10 -> 2 -> 8

In particolare:

- la funzione left rotate() modifica la lista, effettuando una rotazione a sinistra (LEFT) per cui il primo valore viene posto in fondo alla lista. La lista dell'esempio diventa:

  6 -> 10 -> 2 -> 8 -> 3

- la funzione right rotate() modifica la lista, effettuando una rotazione a destra (RIGHT) per cui l'ultimo valore della lista viene messo davanti a tutti gli altri. La lista dell'esempio diventa:

  8 -> 3 -> 6 -> 10 -> 2

Se la lista è vuota o contiene un solo elemento, i sottoprogrammi precedenti non modificano la lista ricevuta in ingresso.
La funzione main() produce il seguente output:

3

3 -> 6 -> 10 -> 2 -> 8

6 -> 10 -> 2 -> 8 -> 3

3 -> 6 -> 10 -> 2 -> 8

8 -> 3 -> 6 -> 10 -> 2

Da Tema ESAME

Completare il file ESA 18062020 B 2.c in modo che il programma apra in lettura un file di testo “testoB.txt” contenente parole (stringhe) ciascuna di al massimo 15 caratteri solo minuscoli (il file è sicuramente non vuoto). In particolare, scrivere:

- la funzione trova, che cerca nel file tutte le parole con almeno 3 vocali, di cui almeno una ’a’, e le stampa. Per contare il numero di vocali la funzione deve richiamare la funzione ricorsiva conta. Si possono utilizzare le funzioni della libreria string.h.
- la funzione ricorsiva conta, che riceve come parametro una stringa e un puntatore ad intero e conta quante vocali minuscole contiene la stringa e di queste quante sono le ’a’.

  Ad esempio, se il file “testoB.txt” contiene il testo: bisogna studiare passo passo e adagio per ottenere un voto alto

Il programma visualizzerà il testo: bisogna studiare adagio