Laboratorio di Programmazione & Informatica 1

Raccolta di esercizi su programmazione in C e gestione liste in C

- Scrivere un programma C che:
 1. Legge una sequenza di interi positivi $x_1, x_2, \dots, x_n$ inseriti da tastiera e li memorizza in una lista. L'immissione termina quando l'utente immette il valore 0.
 2. Partiziona la lista in 2 liste $\langle x_1, x_2, \dots, x_i \rangle, \langle x_{i+1}, x_{i+2}, \dots, x_n \rangle$ tali che la somma degli elementi della prima sotto-lista sia minore o uguale a $1/2$ del totale di tutti gli elementi e che tale somma sia massimale (l'aggiunta di un elemento ulteriore alla sotto-lista fa superare la soglia).
 3. Stampa le sottoliste generate.

I punti 1., 2. e 3. vanno realizzati con delle funzioni distinte.

- Sia dato un grafo diretto G e sia v un vertice con grado entrante 1 (cioè v ha un solo predecessore). Possiamo *disconnettere* v dal grafo eliminando tutti gli archi entranti in e uscenti da v e collegando con un arco il vertice predecessore di v con ogni vertice successore di v . Scrivere una funzione che prende come parametri un grafo come matrice delle adiacenze e un vertice, verifica se il vertice ha grado entrante 1 e in caso affermativo lo disconnette dal grafo modificando opportunamente la matrice.
- Sia P una lista non vuota che contiene ordinatamente i primi k numeri primi ($k > 0$).
 1. Scrivere una funzione C che riceve come parametri la lista e un intero n e restituisce l' n -simo numero primo. Se $n < k$, si limita a ritornare il contenuto dell' n -esimo nodo di P . Altrimenti, oltre a ritornare l' n -esimo numero primo, modifica P aggiungendovi in coda altri $n - k$ numeri primi. Possono essere utilizzate anche funzioni ausiliarie.
 2. Si discutano eventuali variazioni che possano migliorare il tempo di esecuzione dell'algoritmo proposto o mostrare che quello proposto è il migliore possibile.
- Sia L una lista concatenata contenente valori interi positivi. Siano m ed M rispettivamente i valori minimo e massimo presenti in L e sia $r = (m + M)/2$. Scrivere una funzione C che data L costruisce una nuova lista S che è una sottolista di L (cioè i valori in S appartengono ad L e si trovano nello stesso ordine) che contiene tutti i valori $\geq r$.
- Una sequenza s_1 è immersa in s_2 se gli elementi di s_1 occorrono ordinatamente (non necessariamente in modo consecutivo) in s_2 . Ad esempio, la sequenza di interi $\langle 1, 4, 7 \rangle$ è immersa nella sequenza $\langle 2, 1, 3, 4, 6, 7, 9 \rangle$, ma non nella sequenza $\langle 7, 2, 1, 4, 3 \rangle$ (i valori 1, 4, 7 non sono presenti nel giusto ordine).
 1. Si scriva una funzione C che, date due liste di interi L e M , restituisca 1 se la sequenza di interi rappresentata dalla lista L è immersa nella sequenza rappresentata dalla lista M .
 2. Opzionale (+3 punti) Si scriva la stessa funzione in versione ricorsiva.

- Sia dato un grafo G con vertici numerati da 0 a $N - 1$, definito tramite una matrice delle adiacenze e sia L una lista di interi distinti compresi tra 0 ed $N - 1$ definita tramite il puntatore al primo elemento. La lista descrive un sottoinsieme di vertici di G . Sia G' il sottografo di G indotto dai vertici in L (cioé gli archi di G' sono gli archi di G incidenti sui vertici di G'). Scrivere una funzione in C che prende come parametri il grafo e la lista e verifica se il sottografo indotto G' é completo cioé (ogni vertice é collegato con un arco ad ognuno degli altri vertici). Analizzare il tempo di esecuzione dell'algoritmo proposto.

Suggerimento (non é necessario seguirlo!): per ogni vertice nella lista, scorrere tutta la lista controllando la presenza degli archi corrispondenti nella matrice di G .

- Scrivere un programma C che:
 1. Legge una sequenza di interi $x_1, x_2, \dots, x_n$ inseriti da tastiera, $1 \leq x_i \leq 30$ e li memorizza in una lista L . L'immissione termina quando l'utente immette un valore non valido ossia < 1 o > 30 .
 2. Stampa tutta la lista.
 3. Analizza la lista e stampa sullo schermo il numero totale dei valori immessi nella lista e il numero dei valori *distinti* immessi nella lista. (Non si richiede di modificare la lista)
 4. Stampa la sequenza dei valori distinti in un ordine qualsiasi, non necessariamente quello di inserimento.

I punti 1,2,3, 4 vanno realizzati con funzioni.

Suggerimento (non é necessario seguirlo!): utilizzare un vettore di appoggio.

- Scrivere un programma C che legge una sequenza di n interi positivi $x_1, x_2, \dots, x_n$ inseriti da tastiera e li inserisce in una struttura dati tipo lista inserendo sempre il nuovo elemento alla fine della lista corrente. Il numero n non é stabilito a-priori, ma l'immissione termina quando l'utente inserisce il valore 0. Se l'utente inserisce un valore negativo dovrà comparire un messaggio del tipo valore non consentito. Alla fine dell'immissione (dopo il valore 0 che non va inserito nella lista) verranno stampate le seguenti informazioni:

1. numero di elementi nella lista
2. valore piú piccolo inserito
3. valore piú grande inserito
- 4 tutti gli elementi inseriti nella lista (nell'ordine in cui sono stati inseriti)

n.b. E' possibile eseguire le operazioni di conta e calcolo min e max nella fase di inserimento.

- Usando l'istruzione **struct** definire un tipo di nodo-lista che contiene due campi **int** e un campo puntatore. I due campi **int** sono da interpretare come i vertici estremi di un arco di un grafo. Una lista di tali nodi rappresenterà quindi un insieme di archi (e quindi un grafo).

Sia G un grafo di n vertici corrispondenti ai valori $\{0, 1, \dots, n - 1\}$. Definire una funzione che prende in input una lista di nodi definiti come sopra e restituisce una rappresentazione del grafo corrispondente tramite una matrice delle adiacenze. Si assuma che la lista di archi contenga solo valori tra 0 e $n - 1$.

Scrivere poi una funzione che (usando la matrice delle adiacenze) determina se nel grafo esiste un vertice da cui parte un arco verso ogni altro vertice del grafo.