

Fondamenti di Informatica

Esercitazione 13

12 dicembre 2024

Liste Concatenate

13.1 Scrivere un sottoprogramma che data un lista di interi e un intero n in input, rimuove l' n -esimo nodo dalla fine della lista. Implementare una soluzione che scorre la lista una sola volta. Se il nodo n -esimo dalla fine non esiste, il sottoprogramma stampa un messaggio di errore e non modifica la lista.
Esempio: data la lista $[1 \rightarrow 2 \rightarrow 3 \rightarrow 4]$ e $n = 2$ la lista risultante e' $[1 \rightarrow 2 \rightarrow 4]$

13.2 (tema d'esame) Scrivere un sottoprogramma che riceve in ingresso una lista per la gestione dei numeri interi e la modifica nel seguente modo: per ogni tripletta di valori, se il valore centrale è strettamente maggiore degli altri due, viene eliminato. Si noti che le triplette non sono sovrapposte e il numero degli elementi nella lista di partenza non è necessariamente multiplo di 3. Nel caso in cui gli ultimi elementi non formino una tripletta, non viene fatta alcuna operazione su questi. Non sono disponibili sottoprogrammi che manipolano liste: nel caso vengano utilizzati, devono essere sviluppati.

Esempio:

lista in ingresso:

1 -> 2 -> 3 -> 1 -> 7 -> 5 -> -4 -> 5 -> 4 -> 1 -> 2 -|

lista al termine:

1 -> 2 -> 3 -> 1 -> 5 -> -4 -> 4 -> 1 -> 2 -|

13.3 Scrivere un programma che inverte una lista concatenata

13.4 Scrivere un programma che data una lista in input, rappresentata come: $L_0 \rightarrow L_1 \rightarrow \dots \rightarrow L_{n-2} \rightarrow L_{n-1}$ la riordina nel seguente ordine $L_0 \rightarrow L_{n-1} \rightarrow L_1 \rightarrow \dots \rightarrow L_{n-2} \rightarrow L_2 \dots$. Per esempio, data la lista $[1, 2, 3, 4]$ il programma la riordina in $[1, 4, 2, 3]$

13.5 Si vuole realizzare un programma "interattivo" che permette di gestire un percorso su uno spazio bidimensionale composto da una lista di località'. Dichiarare un tipo di dato per rappresentare una località' su una cartina in termini di coordinate x e y (due interi) ed un nome (una stringa di al max 30 caratteri).

Scrivere un programma in grado di gestire un numero indefinito di localita'. Al-
l'avvio il programma visualizza un menu' con le seguenti funzionalita' e chiede
all'utente di selezionarne una:

1. Inserire in testa una nuova localita'
2. Inserire in coda una nuova localita'
3. Inserire in una specificata posizione una nuova localita'
4. Cancellare una localita' nella lista
5. Visualizzare l'intero percorso
6. Visualizzare la lunghezza complessiva dell'itinerario
7. Visualizzare la coppia di localita' (anche non consecutive nel percorso) la
cui distanza e' maggiore di qualsiasi altra coppia
8. Ricerca localita' per nome
9. Calcola la cornice intorno al percorso
10. Salvare percorso su file di testo.
11. Caricare percorso da file di testo.
12. Salvare percorso su file binario.
13. Caricare percorso da file binario.
14. Uscita

Implementare le funzionalita' specificate in base alle seguenti specifiche. Inol-
tre realizzare ciascuna funzionalita' mediante una funzione. Ciascuna funzione
riceve come parametro la struttura dati che contiene la lista di localita' ed ese-
gue le elaborazioni richieste dalla data funzionalita'.

1 - L'inserimento in testa: il programma chiede all'utente i dati di una localita'
e la aggiunge in testa al percorso gia' memorizzato. 2 - L'inserimento in coda:
il programma chiede all'utente i dati di una localita' e la aggiunge in fondo al
percorso gia' memorizzato.

3 - L'inserimento in mezzo: il programma chiede all'utente prima l'indice in
cui inserire la nuova localita' e poi i dati della nuova localita'; il programma
inserisce la nuova localita' nell'indice specificato. Nel caso l'utente inserisca un
indice non valido il programma visualizza un messaggio di errore e non procede
con la richiesta dei dati e l'inserimento.

4 - Il programma chiede all'utente l'indice della localita' da cancellare. Se l'in-
dice e' valido rimuove i dati relativi altrimenti visualizza un messaggio di errore.

5 - Il programma stampa a video il percorso inserito dall'utente

6 - Il programma calcola la distanza del percorso. La distanza tra due locali-
ta' consecutive nell'elenco e' calcolata mediante il teorema di Pitagora. Per il

calcolo della radice quadrata utilizzare la funzione `sqrt()` presente nella libreria `math.h`.

7 - Il programma identifica e visualizza la coppia di località (anche non consecutive nel percorso) la cui distanza è massima. Il calcolo è eseguibile solo se ci sono almeno due località inserite. Se ci fossero più coppie a pari distanza massima, il programma ne stampa una sola (VARIANTE: stampare tutte le coppie).

8 - Ricerca di una località dato il nome. La funzionalità chiede all'utente il nome (una stringa di al massimo 30 caratteri) di una località e stampa a video tutte le località con quel dato nome. Si ricordi che per confrontare le stringhe si può usare la funzione `strcmp()` contenuta in `string.h` che riceve come parametri due stringhe e restituisce 0 se sono uguali.

9 - Calcola cornice del percorso. La funzionalità calcola la cornice rettangolare sulla cartina che racchiude tutto il percorso e ne visualizza i due vertici in basso a destra ed in alto a sinistra. Visualizzare un apposito messaggio di errore nel caso il percorso sia vuoto.

10 - Salvare percorso su file di testo. Il nome del file (al massimo 30 caratteri) è chiesto all'utente.

11 - Caricare percorso da file di testo. Il nome del file (al massimo 30 caratteri) è chiesto all'utente. Per semplicità si assuma che il file non contenga più di 50 punti.

12 - Salvare percorso su file binario. Il nome del file (al massimo 30 caratteri) è chiesto all'utente.

13 - Caricare percorso da file binario. Il nome del file (al massimo 30 caratteri) è chiesto all'utente.

Segnalare eventuali errori di apertura del file. Modificare opportunamente il programma per gestire le nuove opzioni del menù. Se l'utente inserisce un valore sbagliato, il programma visualizza un messaggio di errore e poi nuovamente il menu'.