

ES1.

Si realizzino i seguenti sottoprogrammi utilizzando la struttura dati e funzioni fornite nei file lista.c e lista.h.

Scrivere un sottoprogramma estrai() che riceve come parametro una lista dinamica di numeri interi detta l1, e un parametro intero detto dispari. Il sottoprogramma restituisce un puntatore ad una nuova lista l2 contenente gli elementi di l1 in posizione dispari se il parametro dispari vale 1, o gli elementi in posizione pari se il parametro dispari vale 0.

NOTA: con "posizione" si intende l'equivalente dell'indice se stessimo parlando di vettori invece che di liste: il primo elemento della lista ha indice zero (pari), il secondo elemento ha indice 1 (dispari), e così via.

Scrivere un sottoprogramma uguali() che verifica se due liste ricevute come parametro sono uguali, cioè contengono la stessa sequenza di numeri.

Scrivere un sottoprogramma stessoContenuto() che verifica se due liste l1 ed l2 ricevute come parametro contengono gli stessi valori, a prescindere dall'ordine; cioè se una lista è una permutazione dell'altra. Suggerimento: perchè l1 sia permutazione di l2 basta controllare che le due liste abbiano pari lunghezza e che ciascun valore presente in l1 compaia sia in l1 che in l2 lo stesso numero di volte. Si suggerisce inoltre di implementare anche

- 1) un sottoprogramma di ausilio conta() che riceve come parametri una lista ed un valore intero, e conta e restituisce quante volte il valore intero compare nella lista, ed
- 2) un sottoprogramma lunghezza() che restituisce la lunghezza di una lista ricevuta come parametro.

Scrivere un sottoprogramma unisci() che riceve come parametri due liste l1 ed l2. Il sottoprogramma crea e restituisce una nuova lista concatenando il contenuto di l1 con quello di l2.

Scrivere un programma che carica dal file di testo "serie.txt" (il cui nome è stato passato come argomento alla chiamata del programma) una serie di lunghezza indefinita di numeri interi e li memorizza in una lista lf. Il programma, invocando i sottoprogrammi sopra definiti,

- 1) crea una nuova lista lp, contenente i valori in posizione pari della lista lf.
- 2) crea una nuova lista ld, contenente i valori in posizione dispari della lista lf.
- 3) verifica se lp è uguale a ld ed in caso negativo se almeno contengono gli stessi numeri sebbene in ordine diverso. Il programma stampa a video gli esiti dei test.
- 4) genera una nuova lista u in cui appaiono prima tutti gli elementi di lf originariamente in posizione pari e poi gli elementi originariamente in posizione dispari.
- 5) infine, visualizza (a) la lista di partenza, (b) la lista degli elementi in posizione pari, (c) la lista degli elementi in posizione dispari, (d) la nuova lista con gli elementi riordinati come

specificato.

Infine il programma termina deallocando tutta la memoria allocata.

ES2.

Definire un nuovo tipo di dato per una lista concatenata che permetta di memorizzare un polinomio; ciascun nodo della lista memorizzerà un monomio in termini di due interi che rappresentano il grado e il coefficiente di un dato monomio.

Scrivere un programma che chiede all'utente i dati di due polinomi.

L'acquisizione di ciascun polinomio termina quando l'utente inserisce i dati 0 0 (che non andrà aggiunta alla lista) ed inoltre va prestata attenzione al fatto che una lista non può contenere due monomi con stesso grado; il programma deve prevenire tale

inserimento errato nella lista visualizzando anche un apposito messaggio di errore. Si consiglia di organizzare ciascuna lista ordinando i monomi in ordine decrescente del grado.

In seguito il programma esegue il prodotto e la somma tra i due polinomi salvando il risultato in due nuove liste. Attenzione al fatto che il prodotto tra due monomi può generare un monomio di grado già presente nella lista risultato (e quindi ne va soltanto aggiornato il coefficiente) o un nuovo monomio (che va quindi aggiunto alla lista).

Il programma visualizza infine le quattro liste e, prima di terminare, rilascia tutta la memoria allocata.

Si consiglia di strutturare il programma in vari sottoprogrammi.

ES3.

Definire un tipo di dato per una lista dinamica di parole, dove ciascuna parola è lunga al massimo 20 caratteri. Realizzare i sottoprogrammi standard per la gestione di una lista di parole, in particolare per l'inserimento in coda di un nuovo elemento, la visualizzazione del contenuto della lista, il calcolo della lunghezza della lista e la sua distruzione.

Scrivere un programma che riceve come argomenti i nomi di due file di testo, ciascuno contenente una sequenza di lunghezza indefinita di parole di al massimo 20 caratteri. Il programma carica le due sequenze in due liste dinamiche s1 e s2. In seguito, il sottoprogramma verifica se s1 è sottosequenza di s2 o, viceversa se s2 è sottosequenza di s1, e stampa a video i risultati dei test (1 o 0); si noti che se almeno una delle due liste è vuota l'esito è negativo. Prima di terminare il programma libera tutta la memoria allocata. Visualizzare opportuni messaggi nel caso di errore (es: argomenti mancanti, memoria non allocata correttamente, file non aperto, ...). Si consiglia di suddividere opportunamente il programma in sottoprogrammi.