

Laboratorio Informatica A

Problema 1

Sia data la seguente definizione di lista `typedef struct nodo_t {`

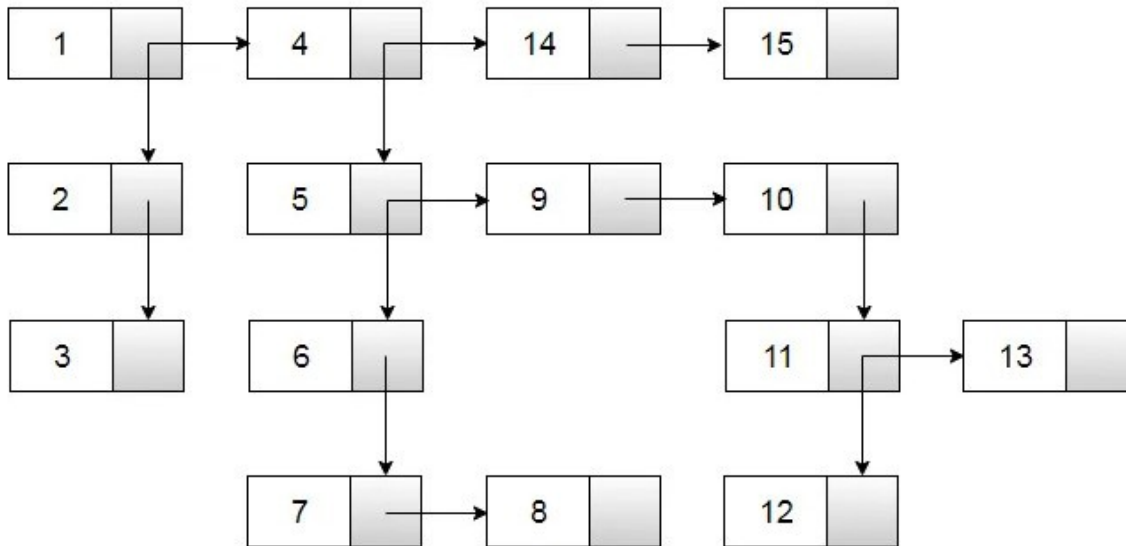
```
int e;  
struct nodo_t *next; struct nodo_t *down;  
} nodo;
```

dove attraverso il puntatore `down` si contempla la possibilità di annidare liste una dentro l'altra a qualsiasi livello di profondità. Si richiede di scrivere una funzione `flatten` che data in ingresso una lista della tipologia sopra specificata ne restituisce la sua versione "appiattita".

Per la versione "appiattita" si utilizzi la seguente struttura dati: `typedef struct nodo_f_t{`

```
int e;  
struct nodo_f_t *next; }nodo_f;
```

Esempio:



1->2->3->4->5->6->7->8->9->10->11->12->13->14->15

Problema 2

Dato un punto nel piano cartesiano, individuato dalle coordinate intere x , y e da una lettera (ad esempio, $A(5,2)$), si chiede di definire un'opportuna struttura dati ed il relativo tipo `punto_t` che lo rappresenti. Si chiede poi di definire una nuova struttura dati che permetta di rappresentare un poligono come una lista di punti. Scrivere quindi un programma, che facendo uso della rappresentazione di poligono introdotta: 1) acquisisca i vertici di un poligono, 2) calcoli il perimetro del poligono, 3) stampi a video il valore del perimetro calcolato.

Problema 3

LightBook è un servizio online simile a Facebook. Per rappresentare i concetti di persona ed amicizia è stata definita una opportuna struttura dati e un array di persone

```
typedef struct _persona {
    char nome[10], cognome[10]; // nome e cognome della persona
    struct _persona * amici[MAX_AMICI]; // array contenente gli indirizzi delle persone nella lista utenti che sono amici di questa persona
    int num_amici; // numero di amici contenuti nell'array amici
    struct _persona * next;
} persona;
persona * lista;
```

Ciò che viene richiesto è di implementare una funzione inserisciPersone per aggiungere utenti a LightBook, una funzione aggiungiAmico per aggiungere un amico nell'array amici di una persona e una funzione ricorsiva calcolaPersoneRaggiungibili che, data una persona iniziale, permette di stampare a video la lista di tutte le persone che puo' raggiungere quella persona passando attraverso la catena di amicizie. Si implementi un opportuno main per testare le funzioni sviluppate.

Problema 4

Costruite un programma per gestire una lista di film. Per immagazzinare i film dovete usare la seguente struttura:

```
typedef struct _movie {
    char title[MAX_LEN];
    char type[MAX_LEN]; // Ad esempio: horror, action . . .
    int year;           // L'anno in cui è stato girato il film
} movie;
```

Per immagazzinare i film che vengono inseriti dall'utente dovete usare una lista. In questo caso, il dato contenuto in ogni nodo della lista non sarà un semplice char o un int, ma una struttura di tipo movie. Nella lista i nuovi film devono essere inseriti in modo ordinato a seconda dell'anno in cui è stato girato il film (prima i film più vecchi poi quelli più nuovi).

Per questo progetto implementate quattro funzioni che devono essere richiamate opportunamente nel main:

- lista add(lista listaFilm) //chiede i dati per un nuovo film, lo aggiunge alla lista listaFilm nella posizione corretta, e restituisce la lista aggiornata.
- void print(lista listaFilm) //stampa la lista dei film.
- lista search(lista listaFilm, char * title) //cerca un film nella lista in base al titolo e restituisce il nodo della lista che contiene tale film.
- lista remove(lista listaFilm, lista film) //rimuove un film dalla lista e restituisce la lista modificata.