Laboratorio 8

Nella propria home directory creare una sottodirectory chiamata es08,
in cui metteremo tutti i file C di oggi.

Note

Quando dovete usare o ritornare dei valori booleani, usate la seguente definizione:
typedef enum {false, true} boolean;
Inoltre quando un esercizio vi chiede di scrivere un "predicato", dovete scrivere
una funzione che ritorni un valore di tipo boolean.

Ricordate che oltre alla funzione o procedura indicata dall'esercizio, si richiede
che scriviate anche un main che usi tale funzione e ne dimostri *ogni* funzionalita'.

In alcuni degli esercizi che seguono (quando richiesto) si utilizzi la seguente
definizione di tipo "lista di interi":

```
typedef struct El {
  int info;
  struct El *next;
} ElementoListaInt;
typedef ElementoListaInt* ListaDiInteri;
```

Inoltre, dove c'e' bisogno di allocazione dinamica della memoria, dovete
assicurarvi di deallocare SEMPRE tutta la memoria allocata: un esercizio
funzionante ma che non deallochi tutta la memoria allocata sara' considerato
scorretto.

Per verificare che il programma 'progesempio' deallochi tutta la memoria, usare
il comando valgrind, che invoca il programma passato e controlla che la memoria
sia tutta libera all'uscita:

```
  valgrind ./progesempio
```

Ricordate infine che dovete includere la libreria stdlib.h per la gestione della
memoria dinamica.

Esercizi strutture:
1) Definire un nuovo tipo di dato capace di rappresentare una data.
Scrivere poi delle opportune funzioni/procedure che:
- ricevuta una data la aggiorni al giorno successivo (ignorando gli anni bisestili);
- ricevute due date verifichino che la prima preceda la seconda;
- ricevute due date, ritornino la differenza in anni fra la prima e la seconda.

2) Definire un nuovo tipo di dato capace di rappresentare i dipendenti di una ditta: di tali dipendenti interessa il codice numerico identificativo unico, la data di assunzione, il grado raggiunto (un intero) e lo stipendio.
Usare il tipo di dato definito nell'esercizio precedente per tutte le
date.
Scrivere le seguenti funzioni/procedure:

- calcolaAnzianita che, dato un dipendente, calcoli il numero di anni trascorsi dalla data di assunzione al 2102;
- aggiornaStipendio che, dato un dipendente, aumenti del 1% il suo stipendio per ogni anno di anzianita' accumulato (attenzione l'interesse da calcolare e' un interesse composto).

- superiore, che dati due dipendenti ritorni 1,0 o -1 se il primo e' rispettivamente un superiore, un pari grado o un sottoposto del secondo.Notare che a parita' di grado, il superiore e' il dipendente conl'anzianita' maggiore. Se e' pari anche l'anzianita', i dipendenti
sono pari grado.

Scrivere quindi un programma che richieda all'utente di inserire i dati di due impiegati, utilizzi le due procedure di cui sopra e ne stampi l'esito.

3) Scrivere un nuovo tipo di dato coppia di interi.
Inizializzare tre istanze di coppie con i primi tre numeri naturali e i loro doppi.

4)Allocare dinamicamente tre istanze del tipo di dato coppia appena definito, inizializzarle con i valori dei primi tre numeri primi e dei loro quadrati e quindi dellaocarle.

5)Scrivere un programma che crei una lista di 3 interi e li inizializzi ai primi tre naturali.

6)Scrivere un programma che crei dinamicamente una lista di 3 interi e li inizializzi ai primi tre naturali.Il programma deve deallocare correttamente la lista prima di uscire,verificare con valgrind che questo sia avvenuto (se correttamente installato sul vostro pc).

7) Scrivere un programma che crei dinamicamente una lista di 3 interi e li inizializzi con valori chiesti all'utente. Il programma deve deallocare correttamente la lista prima di uscire, verificare con valgrind che questo sia avvenuto (se correttamente installato sul vostro pc).

8) Scrivere un programma che chieda all'utente un numero N e crei una lista con N elementi, inizializzata con i primi N naturali
Il programma deve deallocare correttamente la lista prima di uscire,verificare con valgrind che questo sia avvenuto (se correttamente installato sul vostro pc).

9) Scrivere un programma che crei una lista di interi come nell'esercizio 8 e
la stampi a video in questo modo:
1 -> 2 -> 3 -> 4 -> //
Ricordarsi di deallocare la lista prima di uscire dal main (e verificare con
valgrind che questo sia avvenuto, vedi note).

10) Scrivere un programma che, creata una lista di interi come nell'esercizio
precedente, calcoli e stampi la sua lunghezza percorrendo la lista dall'inizio
alla fine.
Ricordarsi di deallocare la lista prima di uscire dal main (e verificare con
valgrind che questo sia avvenuto, vedi note).

11) L'ordinamento parziale di coppie e' cosi' definito: una coppia A e' minore di una coppia B se il primo elemento di A e' minore

del primo elemento di B; se i
primi elementi sono uguali, A e' minore di B se il secondo elemento di A e'
minore del secondo elemento di B; se entrambi gli elementi sono uguali, le due
coppie sono uguali.
Scrivere una funzione che riceva due coppie (come definite precedentemente) e
ritorni 1, 0 o -1 a seconda che la prima sia, rispettivamente, maggiore, uguale
o minore della seconda.

Esercizi su ordinamenti e ricerca
La ricerca binaria e' un metodo per cercare un elemento in un array ordinato con un numero diconfronti al massimo log(N) dove N e' il numero di elementi dell'array.
L'algoritmo e' simile al metodo usato per trovare una parola sul dizionario:
sapendo che il vocabolario e' ordinato alfabeticamente, l'idea e' quella di iniziare la ricerca non dal primo elemento, ma da quello centrale, cioe' a meta' del dizionario. Si confronta questo elemento con
quello cercato:
- se corrisponde, la ricerca termina indicando che l'elemento e' stato trovato;
- se e' inferiore, la ricerca viene ripetuta sugli elementi precedenti (ovvero sulla prima meta' del
dizionario), ignorando quelli successivi;
- se invece e' superiore, la ricerca viene ripetuta sugli elementi successivi (ovvero sulla seconda meta'
del dizionario), ignorando quelli precedenti.
Quando la dimensione del sotto-array raggiunge 0, la ricerca termina indicando che il valore non e'
stato trovato.

12) Scrivere una funzione ricorsiva che riceva un
array ordinato, chieda all'utente un valore e ricerchi il valore nell'array usando l'algoritmo di ricerca binaria spiegato in alto.
Usare una variabile globale per contare il numero di chiamate allafunzione effettuate e verificare la logaritmicita' con il numero di elementi dell'array.

13) Scrivere una funzione ricorsiva che riceva un intero e stampi ricorsivamente un numero corrispondente di punti '.'

14)Scrivere una funzione ricorsiva che riceva un array di interi e ritorni il valore calcolato sommando tutti gli elementi pari e sottraendo tutti gli elementi dispari.

15) Scrivere una funzione ricorsiva che calcoli il coefficiente binomiale (n,m) sfruttando le seguenti relazioni (derivate dal triangolo di Tartaglia):
(n,m) = (n-1, m-1) + (n-1, m)
(n,n) = (n,1) = 1
con 1<=n e 1<=m<=n
5) Scrivere una funzione ricorsiva che ricevuto un array di caratteri, restituisca il carattere piu' piccolo in ordine alfabetico.

16) Sulla base dell'esercizio 1, riuscite a far stampare ricorsivamente punti '.' alternati a '_'? E punti '.' alternati a '_'e a '='?
Qual'e' il metodo generale?
Esempio di interazione col programma
Inserire un numero: 7
7 -> ._._._.
Inserire un numero: 9
9 -> ._.=._.=.

17) Scrivere una funzione merge che presi in input due array ordinati
li fonda in un unico array ordinato. Tale funzione avrà tra i parametri tre array, due sono quelli da fondere ed il terzo conterrà
il risultato. Implementare la versione iterativa e ricorsiva.