

# INFORMATICA - CdL in FISICA

## PROVA SCRITTA DEL 21/07/2014

Scrivere **in stampatello** COGNOME, NOME e MATRICOLA su ogni foglio consegnato

**N.B.:** In tutti gli esercizi viene valutata anche la leggibilità del codice proposto. Inoltre, non è consentito l'uso di istruzioni che alterino il normale flusso dell'esecuzione (come, ad esempio, `continue`, `break` e istruzioni di `return` all'interno di cicli che ne provochino l'uscita forzata).

Se necessario, si utilizzi il tipo `boolean` definito da `typedef enum {false, true} boolean;`

### ESERCIZIO 1 (5 punti)

Si vuole leggere una sequenza di interi nella quale gli interi positivi sono ordinati in maniera strettamente crescente e gli interi negativi sono ordinati in maniera strettamente decrescente. La sequenza termina quando si immette un intero che viola la precedente condizione e che non è considerato far parte della sequenza. Si scriva un programma che legge una sequenza di interi e stampa il numero di interi positivi della sequenza.

Se ad esempio la sequenza digitata dall'utente è: `"-3 1 -5 2 4 -7 7 -5"`  
un possibile output del programma è:

Lunghezza seq. pos : 4.

Mentre se la sequenza digitata dall'utente è: `"-3 1 -5 2 4 -7 8 -9 -10 6"`  
un possibile output del programma è:

Lunghezza seq. pos : 4.

Infine, se la sequenza digitata dall'utente è: `"-3 -1 "` un possibile output del programma è:

Lunghezza seq. pos : 0.

### ESERCIZIO 2 (6 punti)

Sia  $\mathcal{A}$  la successione di interi definita come segue:

$$\mathcal{A}_n = \begin{cases} g(0) & \text{se } n = 0 \\ g(1) & \text{se } n = 1 \\ \mathcal{A}_{n-1} + 2 * g(n-1) & \text{se } n > 1 \end{cases}$$

dove  $g$  è una funzione a valori interi.

È data inoltre una funzione in C con prototipo

```
int G(int *)
```

che, dato un puntatore a intero, calcola il valore di  $g$  sull'intero puntato dal suo argomento. Senza utilizzare la ricorsione, si definisca una funzione con prototipo

```
int A(int)
```

che calcola il valore del termine  $n$ -esimo di  $\mathcal{A}$ , dove  $n$  è il valore del suo argomento.

### ESERCIZIO 3 (5 punti)

Scrivere una funzione **ricorsiva** in C, che dato un array e la sua dimensione, calcoli il numero di elementi adiacenti i cui valori siano ordinati in ordine strettamente decrescente. Se l'array ha dimensione 1 la funzione deve restituire 0.

Ad esempio se l'array contenesse : 2 8 7 5 8 10 5 6 la funzione dovrebbe restituire 5.

### ESERCIZIO 3 (15 punti)

Si vuole rappresentare un multi-insieme di caratteri alfabetici sia minuscoli che maiuscoli tramite una lista. Ogni elemento della lista deve quindi contenere un carattere e il numero di occorrenze ( $> 0$ ) di tale carattere nel multi-insieme. La lista è mantenuta ordinata nel seguente modo: tutti i caratteri minuscoli precedono quelli maiuscoli e poi sia i caratteri minuscoli che quelli maiuscoli sono ordinati in ordine alfabetico. Inoltre, per facilitare la gestione della lista, ad ogni carattere minuscolo è associata un'informazione booleana che indica se il corrispondente carattere maiuscolo è presente nella lista. Analogamente, ad ogni carattere maiuscolo è associata un'informazione booleana che indica se il corrispondente carattere minuscolo è presente nella lista.

Ad esempio, il multi-insieme  $\{(d', 8), (A', 5), (u', 1), (t', 4), (D', 4), (U', 6)\}$  è rappresentato dalla seguente lista:

$[d', 8, True] - > [t', 4, False] - > [u', 1, True] - > [A', 5, False] - > [D', 4, True] - > [U', 6, True]$

(i) (1 punti) Si definiscano i tipi di dato necessari per implementare in C la rappresentazione indicata.

Si definiscano poi le seguenti operazioni, *in modo da mantenere le proprietà della lista e sfruttare tutte le caratteristiche della lista in modo da non fare operazioni inutili*:

- (ii) (4 punti) Scrivere una funzione che data una lista e un carattere controlli se esiste almeno un'occorrenza di tale carattere nella lista. Sfruttare tutte le caratteristiche della lista in modo da fermare appena possibile la ricerca.
- (iii) (5 punti) Scrivere una procedura *ricorsiva* che data una lista e un *carattere maiuscolo* inserisca un'occorrenza di tale carattere nella lista in modo da mantenerne le proprietà'.
- (iv) (5 punti) Scrivere una procedura che data una lista e un carattere *carattere minuscolo* cancelli un'occorrenza di tale carattere dalla lista in modo da mantenerne le proprietà'.