

Informatica - CdL in FISICA

Appello straordinario del 1/6/2012

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

N.B.: Negli esercizi di programmazione, 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). Infine non è consentito l'uso di variabili statiche.

Laddove è utilizzato, il tipo `boolean` è definito da `typedef enum {false, true} boolean;`

ESERCIZIO 1 (3 punti)

Dato il linguaggio simbolico visto a lezione

```
LOAD R1 X | LOAD R2 X | LOAD R1 #C | SUM R1 R2 | SUB R1 R2 |
-----
STORE R1 X | STORE R2 X | READ X | WRITE X | JUMP A |
-----
JUMPZ A | STOP |
-----
```

dove `X` sta per un generico nome simbolico, `A` per un generico indirizzo, e `C` per una generica costante intera tradurre la seguente istruzione:

```
while (x!=y) x = x+1;
```

ESERCIZIO 2 (5 punti)

Si dice che una sequenza di interi non negativi è *bilanciata* se contiene lo stesso numero di pari e di dispari. Si scriva una funzione *iterativa* che legga una sequenza di interi non negativi (senza memorizzarla) che termina quando vengono immessi due numeri uguali e che, supponendo di non considerare l'ultimo come parte della sequenza, restituisca:

- 0 se la sequenza non è bilanciata, né ordinata in modo crescente;
- 1 se la sequenza è bilanciata, e non crescente;
- 2 se la sequenza non è bilanciata, ma è crescente;
- 3 se la sequenza è bilanciata e crescente.

ESERCIZIO 3 (6 punti)

Si definisca una funzione *ricorsiva* che dato un array `vet` di interi e la sua dimensione `dim`, restituisca il più piccolo intero maggiore di tutti gli elementi di `vet`. Se ad esempio l'array contiene : 15 100 0 -20, allora la funzione restituisce 101.

N.B. Ogni chiamata ricorsiva deve risolvere il problema originario per una porzione dell'array.

es416 Si vuole modellare un gioco con le *carte* mediante una lista concatenata. Una carta e' rappresentata dal suo valore (un intero da 1 a 10) e dal suo seme (Quadri,Cuori, Picche e Fiori). Ogni nodo della lista deve rappresentare una carta con in aggiunta l'informazione di quante carte dello stesso seme seguono nella lista.

- 2 punti** Definire i tipi opportuni per rappresentare il gioco.
- 3 punti** Scrivere una funzione *ricorsiva* che data una lista di carte, un valore e il seme di una carta, controlli che tale carta appartenga alla lista *sfruttando al meglio tutte le informazioni contenute nella lista*.
- 5 punti** Scrivere una procedura che data una lista di carte, un valore e il seme di una nuova carta, la inserisca prima della prima carta con lo stesso seme, se esiste, altrimenti la inserisca in coda.
- 6 punti** Scrivere una funzione *ricorsiva* che data una lista di carte, un valore e il seme di una carta, cancelli la prima occorrenza di tale carta, se esiste, nella lista. La funzione deve restituire `true` se la cancellazione e' avvenuta.