

## 12 febbraio – Fondamenti di Informatica – Seconda Parte – 135 minuti

### COMPITO A – Esercizi di Programmazione (23 punti)

#### Esercizio A1 (7 punti): Stringhe

Realizzare un'applicazione **ViaParoleDaDue** definita come segue.

**(6pt)** L'applicazione contiene una funzione **viaParoleDaDue** che riceve come parametro una stringa e la modifica rimuovendo ogni sequenza di esattamente due caratteri alfabetici consecutivi. Ad esempio, se la stringa ricevuta come parametro è "aB12cde56EE78a67bb", la stringa deve essere modificata in "12cde5678a67".

**(1pt)** L'applicazione contiene una funzione **main** che chiede all'utente di inserire una stringa da tastiera ed utilizzando la funzione *fgets* memorizza la stringa introdotta dall'utente all'interno di un array di 50 caratteri. Dopo aver rimosso il carattere '\n' dalla stringa, la funzione **main** invoca la funzione **viaParoleDaDue** fornendole come parametro la stringa letta; la funzione **main** stampa quindi la stringa modificata.

#### Esercizio A2 (6 punti): Ricorsione

Realizzare un'applicazione **Quadrati** definita come segue.

**(4pt)** L'applicazione contiene una funzione **ricorsiva quadratoPrecSucc** che verifica, all'interno di un array di interi ricevuto come parametro, se esiste un elemento che è il quadrato dell'elemento precedente o dell'elemento successivo. Ad esempio, la funzione deve restituire **1** se l'array parametro è [7, 4, 2, -3], in quanto **4** è il quadrato del suo successivo **2**. La funzione deve restituire **1** anche se l'array parametro è [-3, 9, 2, 3], in quanto **9** è il quadrato del suo precedente **-3**. Se l'array parametro è [7, 5, 12, 4], la funzione deve invece restituire **0**, in quanto nessun numero è il quadrato del precedente o del successivo.

La funzione **quadratoPrecSucc** ha due parametri: l'array e la sua lunghezza. Nel caso in cui si desideri utilizzare un terzo parametro per realizzare la ricorsione, deve essere definita un'ulteriore funzione **quadratoPrecSuccRic** che ha tre parametri e che realizza la ricorsione. In tal caso la funzione **quadratoPrecSucc** invoca la funzione **quadratoPrecSuccRic** (fornendole opportuni parametri) per calcolare il risultato.

**(0.5pt)** In un commento che precede la funzione **quadratoPrecSucc** descrivere il tipo di problema (accumulazione/conteggio/verifica esistenziale/verifica universale/minimo-massimo/ricerca) che è risolto dall'algoritmo implementato dalla funzione **quadratoPrecSucc**.

**(1pt)** In un commento che precede la funzione **quadratoPrecSucc** descrivere la specifica della funzione **quadratoPrecSucc** (espressa come Input-Precondizione-Output-Postcondizione).

**(0.5pt)** L'applicazione contiene una funzione **main** che gestisce l'interazione con l'utente. La funzione chiede all'utente la lunghezza della sequenza che vuole inserire. Dopo aver letto e memorizzato tale sequenza in un array, viene stampato un messaggio che indica se esiste un elemento nella sequenza che è il quadrato dell'elemento precedente o dell'elemento successivo oppure no.

#### Esercizio A3 (10 punti): Liste

Realizzare un'applicazione **Libri** per gestire un elenco di libri.

**(1pt)** Definire due strutture, una per rappresentare un elemento della lista ed una per rappresentare un libro che fa parte dell'elenco. In particolare, l'applicazione deve gestire ciascun libro come una struttura con quattro campi. Il primo campo è una stringa che indica il nome del libro; il secondo campo è una stringa che indica il nome dell'autore; il terzo campo è un intero che indica il numero di

pagine del libro; il quarto campo è un carattere che indica se il libro è una prima edizione oppure una ristampa.

**(1.5pt)** Definire una funzione che visualizza l'elenco, stampando per ciascun libro il nome, l'autore, il numero di pagine e la frase "prima edizione" oppure la frase "ristampa", in base al fatto che il libro sia una prima edizione oppure una ristampa.

**(2.5pt)** Definire una funzione che inserisce un libro in coda all'elenco.

**(2pt)** Definire una funzione che cancella il primo libro dall'elenco.

**(3pt)** Definire una funzione main che crea la lista e gestisce l'interazione con l'utente. All'avvio dell'esecuzione l'applicazione deve inizializzare l'elenco con i valori letti da un file; al termine dell'esecuzione l'applicazione deve salvare i dati della lista nello stesso file. La scelta fra l'utilizzo di un file binario oppure di un file testuale è lasciata allo studente.