12 febbraio – Fondamenti di Informatica – Seconda Parte – 135 minuti
COMPITO C – Esercizi di Programmazione (23 punti)

Esercizio C1 (7 punti): Stringhe

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

(6pt) L'applicazione contiene una funzione ***nienteNumeriDueCifre*** che riceve come parametro una stringa e la modifica rimuovendo ogni sequenza di esattamente due caratteri numerici consecutivi. Ad esempio, se la stringa ricevuta come parametro è **"12ab123fg67xyh2iu09"**, la stringa deve essere modificata in **"ab123fgxyh2iu"**.

(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 *nienteNumeriDueCifre* fornendole come parametro la stringa letta; la funzione *main* stampa quindi la stringa modificata.

Esercizio C2 (6 punti): Ricorsione

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

(4pt) L'applicazione contiene una funzione **ricorsiva** ***cuboPrecSucc*** che verifica, all'interno di un array di interi ricevuto come parametro, se esiste un elemento che è il cubo dell'elemento precedente o dell'elemento successivo. Ad esempio, la funzione deve restituire **1** se l'array parametro è **[7, 2, 8, -3]**, in quanto **8** è il cubo del suo precedente **2.** La funzione deve restituire **1** anche se l'array parametro è **[-27, -3, 2, 3]**, in quanto **-27** è il cubo del suo successivo **-3**. Se l'array parametro è **[7, 5, 12, 4]**, la funzione deve invece restituire **0**, in quanto nessun numero è il cubo del precedente o del successivo.

La funzione ***cuboPrecSucc*** 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 ***cuboPrecSuccRic*** che ha tre parametri e che realizza la ricorsione. In tal caso la funzione ***cuboPrecSucc*** invoca la funzione ***cuboPrecSuccRic*** (fornendole opportuni parametri) per calcolare il risultato.

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

(1pt) In un commento che precede la funzione ***cuboPrecSucc*** descrivere la specifica della funzione ***cuboPrecSucc*** (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 cubo dell'elemento precedente o dell'elemento successivo oppure no.

Esercizio C3 (10 punti): Liste

Realizzare un'applicazione ***Pasta*** per gestire un elenco di pacchi di pasta.

(1pt) Definire due strutture, una per rappresentare un elemento della lista ed una per rappresentare un pacco di pasta che fa parte dell'elenco. In particolare, l'applicazione deve gestire ciascun pacco di pasta come una struttura con quattro campi. Il primo campo è una stringa che indica il tipo di pasta (ad esempio, "fusilli"); il secondo campo è una stringa che indica l'azienda che la produce; il terzo

campo è un intero che indica il tempo di cottura; il quarto campo è un carattere che indica se la pasta è integrale oppure raffinata.

(1.5pt) Definire una funzione che visualizza l'elenco, stampando per ciascun pacco di pasta il tipo, l'azienda, il tempo di cottura e la frase "integrale" oppure la frase "raffinata", in base al fatto che la pasta sia integrale oppure raffinata.

(2.5pt) Definire una funzione che inserisce un pacco di pasta in coda all'elenco.

(2pt) Definire una funzione che cancella il primo pacco di pasta 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.