

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

COMPITO B – Esercizi di Programmazione (23 punti)

Esercizio B1 (7 punti): Stringhe

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

(6pt) L'applicazione contiene una funzione **cancellaDueNumericiUguali** che riceve come parametro una stringa e la modifica rimuovendo ogni sequenza di esattamente due caratteri numerici uguali consecutivi. Ad esempio, se la stringa ricevuta come parametro è "22cde555ee12a13367bb5", la stringa deve essere modificata in "cde555ee12a167bb5".

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

Esercizio B2 (6 punti): Ricorsione

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

(4pt) L'applicazione contiene una funzione **ricorsiva verificaSequenza** che verifica, all'interno di un array di interi ricevuto come parametro, se per ogni tripla di elementi adiacenti la somma del primo e del secondo elemento della tripla è uguale al terzo. Ad esempio, la funzione deve restituire **1** se l'array parametro è [6, 4, 10, 14, 24], in quanto $6+4=10$, $4+10=14$ e $10+14=24$. Se l'array parametro è [7, 5, 12, 4], la funzione deve invece restituire **0**, in quanto $5+12 \neq 4$.

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

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

(1pt) In un commento che precede la funzione **verificaSequenzaSucc** descrivere la specifica della funzione **verificaSequenzaSucc** (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 per ogni tripla di elementi adiacenti la somma del primo e del secondo elemento della tripla è uguale al terzo.

Esercizio B3 (10 punti): Liste

Realizzare un'applicazione **Esami** per gestire un elenco di esami.

(1pt) Definire due strutture, una per rappresentare un elemento della lista ed una per rappresentare un esame che fa parte dell'elenco. In particolare, l'applicazione deve gestire ciascun esame come una struttura con quattro campi. Il primo campo è una stringa che indica il nome dell'esame; il secondo campo è una stringa che indica il nome del professore titolare del relativo insegnamento; il terzo campo è un intero che indica il numero di CFU dell'esame; il quarto campo è un carattere che indica se l'esame è a scelta oppure obbligatorio.

(1.5pt) Definire una funzione che visualizza l'elenco, stampando per ciascun esame il nome, il professore titolare del relativo insegnamento, il numero di CFU e la frase "a scelta" oppure la frase "obbligatorio", in base al fatto che l'esame sia a scelta oppure obbligatorio.

(2pt) Definire una funzione che inserisce un esame in testa all'elenco.

(2.5pt) Definire una funzione che cancella l'ultimo esame 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.