17 febbraio – Fondamenti di Informatica – Seconda Parte – 135 minuti Esercizi di Programmazione (24 punti)

Esercizio 1 (6 punti): Stringhe

Realizzare un'applicazione XInRilievo definita come segue.

(5 pt) L'applicazione contiene una funzione xInRilievo che riceve come parametro una stringa e la modifica così che nella stringa compaiano solo le 'X' (ics maiuscole) della stringa di partenza, dove ogni due sequenze di 'X' sono separate da un singolo spazio ''. Ad esempio, se s = "ABCXX1223XxHGXXX", la stringa deve essere modificata in "XXXX", mentre se s = "XXXX", la stringa deve essere modificata in "XXXX".

(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 *xlnRilievo* fornendole come parametro la stringa letta; la funzione *main* stampa quindi un messaggio che mostra all'utente la stringa modificata.

Esercizio 2 (4 punti): Ricorsione

Realizzare un'applicazione Esattamente Uno Pari Grande definita come segue.

(4pt) L'applicazione contiene una funzione ricorsiva esattamente Uno Pari Grande che verifica, all'interno di un array di interi ricevuto come parametro, se ogni tripla di interi adiacenti nell'array contiene almeno due interi maggiori o uguali a 10, di cui esattamente uno pari. Ad esempio, [12, 16, 15] è un'istanza positiva del problema, in quanto l'unica tripla [12, 16, 15] contiene due interi maggiori o uguali a 10 (12 e 15) di cui esattamente uno (12) pari. Invece [12, 16, 7] è un'istanza negativa del problema, in quanto la tripla [12, 16, 7] non contiene due interi che siano entrambi maggiori di 10 e dei quali esattamente uno sia pari.

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

(0 pt) L'applicazione contiene una funzione *main* che gestisce l'interazione con l'utente. La funzione main è già stata implementata in un file *ricorsione.c* il cui codice può essere scaricato e copiato da moodle.

Esercizio 3 (14 punti): Liste

Realizzare un'applicazione **Computer** per gestire i computer di un laboratorio. L'utente dell'applicazione è il gestore del laboratorio.

(1pt) Definire due strutture, per rappresentare un elemento della lista ed un computer. In particolare, l'applicazione deve gestire ciascun computer come una struttura con tre campi. Il primo campo è una stringa che indica il modello del computer (ad esempio "HP 225W1EA"). Il secondo campo è un codice identificativo intero che identifica univocamente il computer all'interno del laboratorio (ad esempio 13). Il terzo campo è un array di 12 interi (ciascuno ad una singola cifra) che indica l'indirizzo IP del computer (ad esempio 1 5 1 0 2 4 1 5 8 1 4 2).

(2pt) L'applicazione contiene una funzione visualizzaLab che permette di visualizzare la lista di computer del laboratorio. Per ciascuno computer, devono essere visualizzati il modello, il codice identificativo e l'indirizzo IP; quest'ultimo va visualizzato inserendo un punto dopo ciascuna tripla di

cifre (ad esempio, per il computer di cui sopra, la visualizzazione potrebbe dire "Il computer è di modello HP 225W1EA, ha codice identificativo 125 ed indirizzo IP 151.024.158.142").

- (2pt) L'applicazione contiene una funzione esisteComputer che, oltre alla lista, prende come parametro un intero e restituisce 1 se esiste un computer nella lista che ha quel codice identificativo, 0 altrimenti.
- (2.5pt) L'applicazione contiene una funzione cancella Computer che cancella il primo computer dalla lista.
- (6pt) L'applicazione contiene una funzione inserisci Computer che inserisce un nuovo computer in coda alla lista. Il modello del computer ed il codice IP devono essere inseriti dall'utente, mentre il codice identificativo deve essere il più piccolo intero positivo che non è già il codice identificativo di un altro computer della lista.
- **(0.5pt)** L'applicazione contiene una funzione *main* che, dopo aver creato una lista di computer inizialmente vuota, ripetutamente chiede all'utente che operazione vuole svolgere fra quelle sopra elencate, oppure se vuole terminare l'applicazione. La funzione main è già stata parzialmente implementata in un file *liste.c* il cui codice può essere scaricato e copiato da moodle.