Fondamenti di Informatica - Ingegneria delle Telecomunicazioni – Prof. Maristella Matera

Appello del 4 Luglio 2014

Il tempo massimo a disposizione per svolgere la prova è di **2 ore**. Non è permessa la consultazione di alcun materiale didattico ed è vietato utilizzare calcolatrici, telefoni, PC.

Il voto minimo per superare la prova è 18.

Esercizio 1 – Stringhe (8 punti).
Si scriva una funzione che, date due stringhe s1 e s2 (ognuna lunga max 20 caratteri), verifichi se s2 è contenuta almeno una volta in s1. La funzione restituisce la posizione in s1 in cui ha inizio la stringa s2 (se contenuta in s1), oppure -1 se s2 non è contenuta in s1.
N.B.: Per la soluzione dell'esercizio, **non deve essere utilizzata la funzione di libreria `strstr`**.

Esercizio 2 – File e Liste Dinamiche (12 punti).
Sia dato un file contenente una lista di punti di interesse (PI) da visitare in un itinerario turistico per la città di Milano. Il file è strutturato in modo che ogni riga contenga il nome di una città (una stringa di 20 caratteri) e un valore intero indicante la posizione del PI nell'itinerario. Per es., la riga "**`Duomo 10`**" indica che il duomo è la decima località prevista nell'itinerario definito nel file.
N.B.: I dati contenuti nel file non sono ordinati. Per esempio, la quinta destinazione potrebbe essere memorizzata prima della prima destinazione. Il numero di destinazioni memorizzate nel file non è noto a priori.

A partire da tale file, si vuole costruire una lista dinamica, **`ListaPI`**, che memorizza i PI estratti dal file **ordinati in base alla loro posizione nell'itinerario**.

1. Si definiscano **opportuni tipi e variabili** necessari per la costruzione della lista.
2. Si scriva quindi la **funzione** … **`CreaListaPI(…)`**, che riceve come parametro il nome del file (una stringa) e, a partire dai dati contenuti nel file, costruisce la lista dinamica, facendo attenzione, durante l'inserimento dei nodi, a ottenere l'ordinamento delle destinazioni. La funzione restituisce il puntatore alla lista così costruita.

Esercizio 3 – Ricorsione (6 punti).
Scrivere in C una **funzione ricorsiva** che riceve come parametri un intero **n** e restituisce il numero di cifre necessario per rappresentare il numero in base 10.
Per esempio, se n=230, la funzione deve restituire il valore 3.

Esercizio 4 – Array (6 punti).
Sia dato un array di n ≥ 2 elementi interi. Si scriva un programma in C, completo di opportune dichiarazioni di variabili e funzioni, che **per rogni elemento dell'array** verifica se il suo valore è inferiore alla somma di tutti gli elementi ad esso precedenti.
Per esempio, nell'array 8 1 9 2 la condizione descritta sopra è verificata per gli elementi di indice 1 e 3, mentre non è verificata per i terzo e banalmente non può essere verificata per l'ultimo.

Il programma deve leggere da tastiera i valori dell'array (max 20), quindi effettuare la verifica per tutte le posizioni dell'array e stampare un opportuno messaggio per evidenziare se i valori nelle varie posizioni sono inferiori alla somma degli elementi precedenti.