CORSO di LAUREA in **INFORMATICA**
Corso di
PROGRAMMAZIONE I e LABORATORIO PROGRAMMAZIONE I
(12 CFU)
A.A. 2013-14

Docenti: Angelo Ciaramella e Giulio Giunta

Cognome: Limite
Nome: Gennaro
Matricola: 0124/891

PROGETTO D'ESAME DI LABORATORIO

1. Algoritmi per la cifratura/decifratura di un messaggio.

Sviluppare una coppia di algoritmi, implementati come function, per crittografare e decrittografare un messaggio. L'algoritmo si basa sulla cosiddetta cifratura polialfabetica, che consiste nel trasformare il messaggio in un testo di lunghezza maggiore o uguale a quella del messaggio, detto il "testo cifrato", utilizzando una matrice di caratteri (prefissata), detta "matrice di sostituzione". Il messaggio da crittografare viene dapprima partizionato in coppie di lettere adiacenti; se in tale partizionamento accade che una coppia è formata dalla stessa lettera, allora si inserisce una X tra le due. Per esempio, il messaggio è LET US MEET AT NOON viene partizionato in LE TU SM EX ET AT NO ON. Si è inserito una X tra le due E, ma non tra le due O, che si trovano in coppie diverse. Si consideri la seguente matrice di sostituzione:

8	J	E	Q	D	N	5	O
P	U	3	A	R	F	L	W
4	V	C	2	T	M	B	I
K	7	Z	S	G	X	H	Y

Ogni coppia di lettere viene crittografata nel seguente modo:

a. se le lettere sono nella stessa riga della matrice di sostituzione, le due lettere da inserire nel testo cifrato saranno le lettere immediatamente a destra nella stessa riga. Ogni riga è considerata circolare. Per esempio, la coppia TI viene crittografata come M4.
b. se le lettere sono nella stessa colonna della matrice di sostituzione, le due lettere da inserire nel testo cifrato saranno le lettere immediatamente sotto nella stessa colonna. Ogni colonna è considerata circolare. Per esempio, la coppia RG viene crittografata come TD.
c. se le lettere appaiono in differenti righe e colonne della matrice di sostituzione, ognuna delle due lettere sarà crittografata con la lettera nella stessa riga ma nella colonna dell'altra lettera.. Per esempio, la coppia LE viene crittografata come 35.

Il messaggio LET US MEET AT NOON viene quindi crittografato in 35VRX2NZDCR25885.

Il main legge da tastiera il messaggio da crittografare (l'equivalente di LET US MEET AT NOON nell'esempio), chiama la function di cifratura (passando come parametro il messaggio e la matrice di sostituzione), che restituisce il teso cifrato, visualizza il testo cifrato, chiama la function di decifratura, passando come parametro il testo cifrato e la matrice di sostituzione, visualizza il messaggio decifrato, che deve coincidere con il messaggio di partenza. Usare solo lettere maiuscole. Usare le stringhe per rappresentare il messaggio e il testo crittografato e decrittografato Fare una versione alternativa del main, in cui la matrice di sostituzione è una permutazione casuale della matrice precedente, usando la function **`rand()`**, il cui prototipo è in **`<stdlib.h>`**, per generare gli interi casuali per lo scambio a coppie di elementi della matrice. Si ricorda che, se **`numero_casuale`** è dichiarata di tipo **`int`**, allora la chiamata **`numero_casuale=rand()%(n+1);`** genera un numero casuale intero (distribuzione uniforme) nell'insieme (0,1,2,..n). Usare sempre la **`srand()`** per rendere automatica la scelta iniziale del *seed* della sequenza di numeri casuali. Nella Relazione si deve riportare l'analisi della complessità di tempo dell'algoritmo (operazione dominante: confronto)

1. Elenco telefonico

   Consideriamo un elenco telefonico composto da 30 utenti. Ogni utente è identificato da un cognome, nome, domicilio e numero telefonico.
   Si supponga che gli utenti inizialmente sono presenti in ordine casuale.
   Permettere al programma di
   - Ordinare l'elenco in base al cognome (usare un algoritmo di ordinamento per inserimento).
   - Dato un cognome, un nome, visualizzare il numero corrispondente.

   Effettuare almeno un test per ognuna delle opzioni richieste dall'utente.

ATTENZIONE – LEGGERE ATTENTAMENTE

La prova d'esame di laboratorio richiede il progetto degli algoritmi e la loro implementazione come programmi C.
Tutti i programmi devono contenere

- un insieme di commenti iniziali che spiega brevemente le finalità del programma;
- un insieme di commenti all'inizio di ogni function che spiega le finalità della `function` e il significato dei parametri di input output (*specifiche* della `function`);
- commenti esplicativi dei principali blocchi di istruzioni;

e devono essere corredati da

- un insieme di almeno **3 esecuzioni** per testare il programma con diversi dati di input.

Lo studente deve consegnare al docente una UNICA relazione organizzata come documento multimediale. In particolare deve essere inviata per e-mail al docente una UNICA cartella (zippata) denominata `Relazione_Cognome_Nome.zip`.

La cartella deve contenere:

- **un file `index.html` che è il documento multimediale;**
- **una cartella `images` che contiene le immagini del documento multimediale;**
- **una cartella `C` contenente i file sorgente del progetto (`.c, .h`);**
- **il testo della prova inviata dal docente in formato `.pdf` ;**
- **altre cartelle eventualmente generate per il documento.**

La relazione deve contenere necessariamente almeno

- **il testo della prova inviata dal docente;**
- **il testo dei programmi C (sorgente);**
- **l'output e la descrizione dei test di esecuzione.**

I test devono essere almeno tre per ogni programma, devono essere salvati come "*print screen*" e come figure nel documento multimediale. Devono essere corredati da una descrizione per l'interpretazione dei risultati del test.

La relazione deve riportare chiaramente il nome e cognome dell'allievo e la sua matricola.

La relazione deve essere inviata al docente per e-mail (`angelo.ciaramella@uniparthenope.it`) entro la data di scadenza della prenotazione on-line dell'esame e deve essere inviata esclusivamente dall'indirizzo e-mail personale dello studente (`nome.cognome@studenti.uniparthenope.it`).

IL NOME DELLA CARTELLA CHE CONTIENE LA RELAZIONE DEVE ESSERE `Relazione_cognomeallievo_nomeallievo.zip`

NON SARANNO ESAMINATI PROGETTI DIFFORMI DA QUANTO PRECISATO.