Esercizi 21-11-23

Riccardo Gennaro

November 2023

Esercizio 1 (Archivio studenti)

Sviluppa un programma per gestire un archivio di studenti utilizzando una struttura `Student` che descrive nome, cognome e id, e una struttura `Archivio` che contiene gli studenti ordinati per cognome e id. Per mantenere due ordinamenti, la struttura `Archivio` usa due array. Si richiedono funzioni per l'aggiunta e la stampa degli studenti. Non è richiesto l'input dall'utente, potete usare il file `01-esercizio_main.c|.cpp` per il testing.

Esercizio 2 (Inverti array ricorsivo)

Completare il programma definito nel file `02-esercizio.cpp` con la dichiarazione e la definizione della funzione ricorsiva `reverseArray`, che, preso in input un array di caratteri ed un intero che rappresenti la sua lunghezza, inverta l'array di caratteri passato come parametro; inoltre, il codice deve sostituire ogni vocale minuscola ('a', 'e', 'i', 'o', 'u') con un punto esclamativo ('!'). L'array in ingresso corrisponde ai caratteri della stringa passata come primo argomento da riga di comando all'atto dell'invocazione del programma. Per esempio, se l'eseguibile è `a.out`, dato il comando:

```
./a.out aIUolA
```

l'output del programma dovrà essere:

```
Array invertito: Al!UI!
```

NOTA 1: La funzione `reverseArray` deve essere ricorsiva ed al suo interno non ci possono essere cicli o chiamate a funzioni contenenti cicli. Si può fare uso di eventuali funzioni ricorsive ausiliarie.

NOTA 2: È vietato modificare in alcun modo il codice della funzione `main`.

NOTA 3: Si assuma che la dimensione massima di una parola sia `DIMMAX` caratteri.

NOTA 4: All'interno di questo programma non è ammesso l'utilizzo di variabili globali o di tipo static e di funzioni di libreria, comprese quelle della libreria `cstring` o `string.h`.

Esercizio 3 (Panico)

Scrivere un programma che prenda come argomento del main:

- Il nome di un file di testo in input, contenente una sequenza di parole separate da spazi, di soli caratteri 0...9,a...z (10 cifre e 26 lettere minuscole).
- Il nome di un file di testo da usare per l'output.

Il programma chiede all'utente un numero di al massimo 7 cifre da usare come **chiave di criptazione** del file letto. Richiedere continuamente all'utente la chiave se il numero immesso ha più di 7 cifre. Il programma quindi legge il contenuto del file di input parola per parola **troncando al quarto carattere**, e.g. abcde diventa abcd.

Decodifica la parola nella sua rappresentazione decimale, effettuando una trasformazione dalla base 36 alla base 10. La base 36 usa le cifre 0...9 e 26 caratteri minuscoli a...z, che assumono valori compresi tra 0 e 35, ad esempio: $036 \rightarrow 0$, $936 \rightarrow 9$, $a36 \rightarrow 10$, $z36 \rightarrow 35$. Data una parola/numero in base 36 sulla destra abbiamo la cifra/carattere meno significativo e poi man mano potenze di 36 crescenti. Ad esempio, $abc36 \rightarrow 10 \times (36^2) + 11 \times (36^1) + 12 \times (36^0) = 13368$, altri esempi sono $1z36 \rightarrow 71$, $2036 \rightarrow 72$, $2136 \rightarrow 73$.

La chiave intera precedentemente acquisita viene quindi sommata ad ogni parola decodificata per criptarla.

Una volta criptata, la parola va ricodificata nella sua rappresentazione originale in base 36, quindi cifre 0...9 e lettere minuscole a...z, e stampata nel file di output.

Ad esempio, se la chiave inserita vale 1, a36 diventa b36, mentre 1z36 (=71) diventa 2036 (=72).

Per fare la ricodifica, effettuare una trasformazione dalla base 10 alla base 36 per codificare le parole, ad esempio $13368 \rightarrow abc36$. Qui un possibile procedimento: (1) Il primo carattere/cifra a destra si ottiene facendo il modulo 36 del numero, e.g. $13368\%36 = 12$. (2) Quindi si rappresenta il valore ottenuto con il relativo carattere/cifra, e.g. $12 \rightarrow c36$. (3) Poi, sottratto il modulo si divide per 36, e.g. $(13368 - 12)/36 = 371$. (4) Si ripete da (1) fino ad avere resto 0. (5) Al termine riordinare le cifre dalla più significativa a sinistra alla meno significativa a destra. Ogni cifra può quindi essere usata per trasformarla nel carattere corrispondente per la base 36 (secondo la codifica specificata con 0,...,9,a,...,z).

Supponiamo che il primo file `input.txt` contenga

```
sole mare luna mondo
```

Eseguendo il programma `./esercizio1.out input.txt output.txt` verrà chiesta all'utente la chiave per criptare il testo. Inserendo **36** il programma dovrà produrre un file chiamato `output.txt` che conterrà i seguenti valori:

```
some mase luoa mood
```

Note:

- Si assuma che il file di input includa solo cifre 0...9 e caratteri a...z. Si noti che 36 è dato dal conteggio delle cifre 0...9 e dei caratteri a...z, dove quindi 'z' - 'a' = 25.
- È ammesso l'uso della libreria `cstring`, e.g. `strlen`.
- È consentito l'utilizzo della libreria `cmath`, e.g. la funzione `pow`.
- Non è consentito l'utilizzo di altre funzioni di libreria "particolari" diverse da quelle specificate sopra o da quelle standard necessarie a risolvere l'esercizio.
- Si noti che le parole in input devono essere troncate per evitare problemi di overflow, lo stesso vale per la chiave numerica chiesta all'utente che deve soddisfare la lunghezza specificata nel testo.
- Si ricorda che, l'esempio di esecuzione è puramente indicativo, e la soluzione proposta NON deve funzionare solo per l'input fornito, ma deve essere robusta a variazioni compatibili con la specifica riportata in questo testo.