

12BHD INFORMATICA, A.A. 2017/2018

Esercitazione di Laboratorio 8

Obiettivi dell'esercitazione

- Scrivere programmi che leggano e manipolino caratteri, stringhe e matrici

Contenuti tecnici

- Uso dello specificatore di formato `%s`
 - Uso delle funzionalità contenute in `string.h` e `ctype.h`
 - Uso delle funzioni per la formattazione delle stringhe
 - Uso delle matrici
-

Da risolvere preferibilmente in laboratorio

Esercizio 1. Si scriva un programma che:

- Definisca un vettore di caratteri e acquisisca una stringa al suo interno
- Analizzi tale stringa rispondendo alle seguenti domande
 - Quanto è lunga la stringa?
 - Quanti caratteri sono alfabetici e quanti numerici?

Approfondimento: acquisita una seconda stringa, stabilire se quest'ultima è inclusa nella prima (ad esempio: "importante" include "porta")

Esercizio 2. Si scriva un programma C che:

- Acquisisca una stringa di massimo N caratteri (con N valore costante)
- Ne manipoli il contenuto
 - Trasformando tutte le lettere minuscole in maiuscole
 - Rimpiazzando tutti i caratteri non alfanumerici con il carattere '_'
 - Sostituendo i caratteri numerici con il carattere '*'
- Scandisca la stringa manipolata per contare quante parole sono presenti al suo interno, considerando una o più occorrenze del carattere '_' come separatore tra parole.

Approfondimento: l'ordine in cui vengono eseguite le manipolazioni influenza il risultato? Verificare la risposta scrivendo due versioni del programma che manipolino la stringa in modi differenti.

Esercizio 3. Si scriva un programma che acquisisca 2 stringhe corrispondenti a 2 orari nel formato `hh:mm`. Il programma deve:

- Controllare le stringhe, segnalando i casi in cui il formato non sia rispettato (ad esempio 10,30 non è valido)
- Stabilire se l'orario contenuto nella prima stringa è precedente a quello contenuto nella seconda stringa
- In caso affermativo, tradurre i 2 orari in valori interi corrispondenti all'orario espresso come distanza in minuti da 00:00 e calcolarne la differenza
- Convertire il risultato (sarà un numero intero positivo) in una stringa così composta "<intervallo calcolato>_minuti" e la stampi a video.

- Esercizio 4. Si scriva un programma che acquisisca utilizzando la funzione *gets* una stringa composta da un massimo di 5 parole separate da spazi, per un totale di massimo 60 caratteri. Il programma deve
- Stabilire quante sono le parole contenute effettivamente nella stringa
 - Calcolare la media della lunghezza delle parole
 - Produrre una statistica sulla lunghezza delle parole

Esempio:

Se la stringa inserita è “questa stringa contiene cinque parole”
allora visualizzerà a video

- La stringa contiene 5 parole
- La lunghezza media delle parole è 6,6 caratteri
- La stringa contiene
 - o 3 parole da 6 caratteri
 - o 1 parola da 7 caratteri
 - o 1 parola da 8 caratteri

Invece, se la stringa inserita è “questa stringa è corta”
allora visualizzerà a video

- La stringa contiene 4 parole
- La lunghezza media delle parole è 4,75 caratteri
- La stringa contiene
 - o 1 parola da 1 carattere
 - o 1 parola da 5 caratteri
 - o 1 parola da 6 caratteri
 - o 1 parola da 7 caratteri

- Esercizio 5. Si scriva un programma che acquisisca 3 stringhe, ciascuna contenente il nome di un prodotto e il relativo prezzo separati da spazio. Una volta memorizzate queste informazioni in opportune variabili, il programma riceve un’ulteriore stringa contenente un nome di prodotto e un valore intero corrispondente a una quantità.

Il programma deve:

- Stabilire se il prodotto inserito per ultimo corrisponde ad uno dei prodotti memorizzati in precedenza
- In caso negativo, richiedere un altro inserimento di nome prodotto e quantità
- In caso positivo, calcolare e visualizzare il costo totale moltiplicando quantità e prezzo del prodotto in questione

- Esercizio 6. ¹Si scriva un programma C che acquisisca sequenze di caratteri da tastiera conclusa da un ritorno a capo. Il programma deve continuare ad acquisire

¹ Questo esercizio sarà svolto in modo multimediale e inserito sul Portale, tra il materiale comune, nelle settimane successive.

sequenze fino alla ricezione di un EOF (*Ctrl-z*). Il programma deve quindi stampare le sequenze caratteri acquisiti

- a. Sostituendo ad ogni sequenza "ch" il carattere 'k'
- b. Sostituendo le doppie con una sola ripetizione del carattere.

Esempio: rischio → riskio

 cammello → camelo

Esercizio 7. Si scriva un programma C che:

- legga da tastiera (per righe o per colonne, a scelta) una matrice quadrata di dimensione uguale a 5 righe e 5 colonne
- rintracci se tale matrice contiene delle sequenze di elementi adiacenti uguali a zero di lunghezza uguale o maggiore di 3
- visualizzi l'indice di riga in cui tali sequenze si presentano.

Esempio.

Sia la matrice la seguente:

```
0 0 0 4 5
1 2 0 4 5
1 0 0 4 0
1 2 3 4 5
1 0 0 0 0
```

La sequenza di valori "0 0 0" compare nella prima e nell'ultima riga e quindi occorre riportare una indicazione del tipo:

La sequenza compare nella riga 1

La sequenza compare nella riga 5

Si osservi che la riga 3 non contiene la sequenza indicata in quanto i tre zeri non si trovano in posizioni contigue.

FACOLTATIVO

Si effettui lo stesso controllo anche lungo le colonne.

Nell'esempio precedente occorre visualizzare, oltre ai messaggi già indicati, anche il seguente:

La sequenza compare nella colonna 3

FACOLTATIVO

Si generalizzi l'esercizio precedente per gestire sequenze di lunghezza variabile e contenenti un valore variabile.

Ovvero si leggano da tastiera due valori che specificano la lunghezza e il valore contenuto nelle sequenze da ricercare.

Nel caso sia:

Lunghezza della sequenza = 3

Valore nella sequenza = 0

occorre risolvere l'esercizio originale.
Se invece si introduce

Lunghezza della sequenza = 4
Valore nella sequenza = 1

occorre ricercare delle sequenze di 4 valori adiacenti uguali a 1,
ovvero "1 1 1 1".