

06AZN - Fondamenti di Informatica (GES, LOP, ORG)
Esercitazione di laboratorio n. 8 (1/12/09)

La presente esercitazione presuppone conoscenza delle seguenti parti del linguaggio C: main, return, commenti, variabili, identificatori, tipi scalari, assegnazione, costanti letterali, printf (base), scanf (deprecata), operazioni aritmetiche, if...else, operatori relazionali, operatori Booleani, getchar, putchar, ciclo while, libreria ctype.h, const, #define, exit, programmazione a stati, gets, sscanf, stringhe, traduzione di algoritmi matematici; switch, break, #ifdef, math.h, cast, printf (completa), ciclo for; argc, argv; definizione di funzioni; file sequenziali, fopen, fclose, fgetc, fprintf; string.h

Esercizio 1

Scrivere un programma in linguaggio C che legga da un file (il cui nome è passato come primo parametro da riga di comando) una lista di numeri interi (due per riga) e stampi in output il numero ordinale delle righe ove compare il numero N passato come secondo parametro da riga di comando. Inoltre il programma deve stampare quante volte è presente in totale N nel file. Ad esempio, se il programma (chiamato trovanum.c) venisse attivato nel seguente modo:

```
trovanum valori.txt 7
```

e se il file valori.txt contenesse le seguenti righe:

```
7 10  
11 21  
27 18  
7 7  
7 172
```

il programma dovrebbe fornire il seguente output:

```
1  
4  
5  
il numero 7 e' presente 4 volte in 3 righe
```

Esercizio 2

Scrivere un programma in linguaggio C che riceva come primo parametro da riga di comando il nome di un file contenente una lista di parole (una parola per riga) e come secondo parametro da riga di comando un numero intero N. Il programma deve stampare in output la N-esima parola in maiuscolo. Ad esempio, se il programma (chiamato paromaiu.c) venisse attivato nel seguente modo:

```
paromaiu parole.txt 3
```

e se il file parole.txt contenesse le seguenti righe:

```
Albero  
Foresta  
bosco  
Piante  
radici
```

il programma dovrebbe fornire il seguente output:

BOSCO

Una volta risolto l'esercizio, provare a scrivere una funzione `str_toupper` che renda una stringa maiuscola.

Esercizio 3

Scrivere un programma in linguaggio C che legga dal file `parole.txt` una lista di parole (una per riga) e scriva la parola più lunga in un secondo file il cui nome è passato come primo parametro da riga di comando. Ad esempio, se il programma (chiamato `parolunga.c`) venisse attivato nel seguente modo:

```
parolunga piulunga.txt
```

e se il file `parole.txt` contenesse le seguenti righe:

```
Torino  
Ancona  
Capracotta  
Roma  
Morteron
```

il programma dovrebbe scrivere nel file `piulunga.txt`:

```
Capracotta
```

Esercizio 4

Scrivere un programma in linguaggio C che legga in ogni riga (finché ce ne sono) una parola e ne restituisca la sua lunghezza. Ad esempio se il programma riceve il seguente input:

```
Sicilia  
Calabria  
Basilicata  
Puglia  
Marche
```

il programma dovrebbe generare il seguente output:

```
7  
8  
10  
6  
6
```

Esercizio 5

Scrivere un programma in linguaggio C che legga in input righe (finché ce ne sono) contenenti *nome* e *cognome* separati da uno o più spazi. Il programma deve scrivere in output la lunghezza del cognome presente su ogni riga e presentare alla fine la coppia *nomecognome* avente lunghezza massima del cognome. Ad esempio se il programma ricevesse il seguente input:

```
Riccardo Rossi  
Maria Benedetto  
Mario Ferrero  
Elena Giordano
```

il programma dovrebbe generare il seguente output:

```
5  
9  
7  
8  
Il cognome piu' lungo e' di Maria Benedetto e conta 9 caratteri.
```

Esercizio 6

Scrivere un programma in linguaggio C che riceva come primo parametro da riga di comando il nome di una regione italiana e legga il file `dati.txt` contenente una per riga tutte le regioni italiane. Il programma deve stampare in output il numero della riga della regione richiesta. Ad esempio, se il programma (chiamato `regioni.c`) venisse attivato nel seguente modo:

```
regioni Sicilia
```

e se il file `dati.txt` contenesse le seguenti righe:

```
Lombardia  
Campania  
Lazio  
Sicilia  
Veneto  
...
```

il programma dovrebbe generare il seguente output:

```
Sicilia : 4
```

Esercizio 7

Scrivere un programma in linguaggio C che presenti in output il numero di abitanti della città il cui nome è passato come primo parametro da riga di comando, sapendo che è disponibile il file `citta.txt` contenente in ogni riga il nome di una città e il numero dei suoi abitanti. Si richiede che il confronto tra i nomi presenti nel file ed il nome passato sulla riga di comando ignori le differenze tra maiuscole e minuscole (ossia sia *case-insensitive*). Ad esempio, se il programma (chiamato `abitanti.c`) venisse attivato nel seguente modo:

06AZN - Fondamenti di Informatica (GES, LOP, ORG)
Esercitazione di laboratorio n. 8 (1/12/09)

abitanti VENEZIA

e se il file `citta.txt` contenesse le seguenti righe:

```
Roma 2546804
Milano 1256211
Napoli 1004500
Torino 865263
Venezia 271073
Monza 120204
```

il programma dovrebbe generare il seguente output:

```
La citta' di Venezia ha una popolazione di 271073 abitanti
```

Esercizio 8

Scrivere un programma in linguaggio C che riceva come primo parametro da riga di comando un numero di riga `N` del file `PARC.txt` (contenente due parole per riga) e dica se la parola passata come secondo parametro da riga di comando è la parola composta che si ottiene unendo le due parole della `N`-esima riga con un trattino.

Ad esempio, se il programma (chiamato `par_comp.c`) venisse attivato nel seguente modo:

```
par_comp 4 high-level
```

e se il file `PARC.txt` contenesse le seguenti righe:

```
policy based
high level
low level
medium level
off line
quasi integrabile
```

il programma dovrebbe generare il seguente output:

```
La parola medium-level e' diversa da high-level
```

Esercizio 9

Scrivere un programma in linguaggio C riceva il nome di un file passato come primo parametro da riga di comando contenente in ogni riga una parola. Il programma deve trovare e stampare in output le righe contenenti il carattere passato come secondo parametro da riga di comando. Ad esempio, se il programma (chiamato `trovacar.c`) venisse attivato nel seguente modo:

```
trovacar citta.txt z
```

e se il file `citta.txt` contenesse le seguenti righe:

Torino
Ancona
Firenze
Capracotta
Roma
Morteronone
Venezia

il programma dovrebbe generare il seguente output:

Firenze
Venezia

Esercizio 10

Scrivere un programma in linguaggio C riceva il nome di un file passato come primo parametro da riga di comando. Il programma deve trovare e stampare in output le righe contenenti la sottostringa passata come secondo parametro da riga di comando. Ad esempio, se il programma (chiamato `trovastr.c`) venisse attivato nel seguente modo:

```
trovastr dati.txt luogo
```

e se il file `dati.txt` contenesse le seguenti righe:

```
Roma: capitale d'Italia e capoluogo del Lazio, ha piu'  
di due milionicinquecentomila abitanti. Sorge sulle rive  
del fiume Tevere, a sud della confluenza con l'Aniene,  
a nordovest del vulcano laziale. Roma e' il luogo ideale per  
chi vuole trascorre una vacanza.
```

il programma dovrebbe generare il seguente output:

```
Roma: capitale d'Italia e capoluogo del Lazio, ha piu'  
a nordovest del vulcano laziale. Roma e' il luogo ideale per
```

Esercizio 11

Realizzare un programma in linguaggio C che riceva come primo parametro da riga di comando il nome di un file contenente il percorso fatto da un'auto, descritto come una sequenza di tratti rettilinei di cui sono fornite le coordinate rispetto ad un'origine arbitraria. Ogni riga del file contiene le coordinate XY (numeri reali) che indicano la distanza in chilometri dall'origine del punto di arrivo di questa tratta; La prima riga del file indica le coordinate di partenza. Non è nota a priori la lunghezza del file. Il programma deve calcolare:

- la lunghezza totale del percorso (espressa in km, con la precisione delle centinaia di metri);
- il tratto più corto (espresso in km, con la precisione delle centinaia di metri).

Esempio. Supponiamo che il programma si chiami `viaggio` e che il file `dativiaggio.txt` contenga i seguenti dati:

06AZN - Fondamenti di Informatica (GES, LOP, ORG)
Esercitazione di laboratorio n. 8 (1/12/09)

100 190
90 200
90 280
80 300

Nel caso che il programma sia attivato nel seguente modo:

viaggio dativiaggio.txt

il programma dovrebbe generare il seguente output:

distanza totale percorsa = 116.5 km
tratta piu' corta = 14.1 km, da (100,190) a (90,200)