
PROGRAMMAZIONE I

A.A. 2019 - 2020
CORSO DI LAUREA IN INFORMATICA
DOCENTI: G. DE MARCO R. ZIZZA

PROVA DI LABORATORIO - 18 FEBBRAIO 2020

La prova richiede la generazione di due file: uno per l'esercizio 1 e un altro per l'esercizio 2. I due file dovranno essere denominati `cognome_nome_1.c` e `cognome_nome_2.c`.

File del secondo esercizio:

`www.di-srv.unisa.it/professori/demarco/laboratorio/stelle.txt`

URL di consegna:

`www.di-srv.unisa.it/professori/demarco/laboratorio/esame.php`

Alt Gr + 3 \Rightarrow #

Alt Gr + 7 \Rightarrow {

Alt Gr + 0 \Rightarrow }

Alt Gr + è \Rightarrow [

Alt Gr + * \Rightarrow]

Esercizio 1 Si scriva un programma che, preso un intero x e una stringa s (composta da lettere minuscole e cifre numeriche) dalla linea di comando, inverta la posizione delle lettere contenute in s se $x = 1$, inverta invece la posizione delle cifre numeriche se $x=2$. Nel caso s contenga caratteri non corrispondenti né a lettere minuscole né a cifre numeriche, il programma lascia invariata la stringa.

Ad esempio su $x = 1$, $s = \text{ciccio123bello45}$ il programma dovrà stampare la stringa ollebo123iccic45 ; su $x = 2$, $s = \text{ciccio123bello45}$ stamperà invece la stringa ciccio543bello21 ; se $s = \text{ciccio123.bello45}$, stamperà ciccio123.bello45 .

Il programma dovrà usare una funzione `void invert (int x, char *s)` che prenda in input un intero x e una stringa s e modifichi s (senza l'ausilio di stringhe di appoggio) opportunamente nella stringa di output secondo le regole stabilite sopra.

Attenzione: Il programma può usare solo le librerie `<stdio.h>` e `<stdlib.h>`.

. ~ .

Esercizio 2 Scrivere un programma *stelle* che permetta di esaminare la lista delle stelle più brillanti osservabili a occhio nudo memorizzata in un file `stelle.txt`. Il programma può essere usato in tre modi.

1) Prende in input da linea di comando il nome di una stella. Esaminando il file, stamperà il nome della stella, la sua magnitudine apparente e la distanza dalla terra (ci si attenga al formato indicato nell'esempio 2).

2) Prende in input da linea di comando un numero indicante una distanza. Esaminando il file, stamperà, l'elenco delle stelle che sono ad una distanza minore o uguale a quella indicata; per ciascuna di esse, stamperà anche la magnitudine (vedi l'esempio 3 per il formato dell'output).

3) Prende in input da linea di comando il carattere m seguito da due numeri a e b indicanti una distanza. Esaminando il file, stamperà, tra tutte le stelle la cui distanza è compresa tra a e b , quella che presenta la minima magnitudine e quella con la massima magnitudine. (vedi l'esempio 4 per il formato dell'output).

Attenzione: Per la stampa dell'output, ci si attenga strettamente alle indicazioni date sopra e illustrate nei seguenti esempi, evitando di introdurre frasi non richieste, come: "le stelle ... sono ..., la magnitudine ..., ecc. **Soluzioni che ignorino tali indicazioni, rischiano di essere penalizzate o di non essere prese in considerazione!**

Il file `stelle.txt` contiene l'elenco delle stelle ed ha il seguente formato. Ogni riga corrisponde ad una stella e presenta nell'ordine: la magnitudine, il suo nome, la distanza dalla Terra (espressa in anni luce). Si noti che il numero di spazi fra gli elementi è indefinito.

```
-26.73    Sole    0.000016
-1.47     Sirio   8.6
-0.62     Canopo  310
-0.27     Alfa Centauri  4.4
-0.04     Arturo  37
0.03      Vega    25
0.08      Capella  42
0.12      Rigel   770
0.34      Procione  11
0.50      Achernar 140
...
```

Esempio 1.

```
>./a.out
Uso: ./a.out <stella>
Uso: ./a.out <distanza>
Uso: ./a.out m <distanza>
```

Esempio 2.

```
> ./a.out 'Alfa Centauri'
Alfa Centauri, magnitudine: -0.27 distanza: 4.4 anni luce
```

Esempio 3.

```
> ./a.out 50
-26.73 Sole
-1.47 Sirio
-0.27 Alfa Centauri
-0.04 Arturo
0.03 Vega
0.08 Capella
0.34 Procione
0.77 Altair
1.15 Polluce
1.16 Fomalhaut
```

Esempio 4.

```
> ./a.out m 50 100
0.98 Aldebaran (65.00)
1.63 Gacrux (88.00)
```

. ~ .