

Problema

Un file di testo contiene una serie di stringhe di testo. Ciascuna stringa è separata da un carattere di andata a capo. Il numero di righe non è noto a priori, mentre la lunghezza massima di ciascuna stringa è pari a 50 caratteri. Il file contiene almeno una linea.

NOTA: la funzione consigliata per la lettura delle stringhe è `fgets`. Nell'elaborazione della stringa letta con `fgets` si tenga presente che essa include anche il carattere di andata a capo.

Informazioni sul programma richiesto

Si scriva un programma in linguaggio C in grado di elaborare un file avente il formato descritto, al fine di restituire i risultati indicati nei punti seguenti. Il programma deve poter essere invocato da linea di comando. Un esempio di invocazione è la seguente:

```
./a.out nome_input_file
```

dove

- `a.out` è il nome del programma eseguibile da invocare;
- `nome_input_file` è il nome del file di dati da elaborare.

IMPORTANTE: il programma dovrà produrre la stampa di risultati esattamente col formato specificato nei vari punti. In particolare, *non aggiungere all'output del testo non richiesto*.

NOTA: Il buon funzionamento del programma può essere verificato col comando

```
./pvcheck esame.test ./a.out esame.dat
```

1 Numero di vocali

Per ciascuna stringa, conteggiare il numero di vocali e stamparne il valore. Stampare a video i valori col seguente formato:

```
[VOCALI]
numero_vocali_stringa_1
numero_vocali_stringa_2
...
```

NOTA: non si ometta il “separatore di risultati”, in questo caso la stringa "[VOCALI]", prima dell'elenco dei valori calcolati.

2 Numero minimo e massimo di vocali

Indentificare e stampare le stringhe contenenti il minimo e il massimo numero di vocali. In caso più di una stringa contenga un numero di vocali pari al minimo, stampare l'ultima presente nel file. Lo stesso vale per la stringa contenente il massimo numero di vocali.

Stampare a video i valori col seguente formato:

```
[MINMAX-VOCALI]
stringa_con_minimo_numero_di_vocali
stringa_con_massimo_numero_di_vocali
```

3 Inversione caratteri

Per ciascuna stringa, invertire l'ordine dei caratteri e stampare la stringa invertita. Per esempio, invertendo la stringa “Mario Rossi” si ottiene “issoR oiraM”.

Stampare a video i valori col seguente formato:

```
[INVERSIONE-CARATTERI]
stringa_1_invertita
stringa_2_invertita
...
```

4 Conteggio parole

Conteggiare il numero di parole presenti in ciascuna stringa, e stamparne il valore. Si considerano *parole* le sequenze di caratteri separati da uno o più spazi. Per esempio, le stringhe “Bel tempo.”, “A, B, 1, 2, fine.” e “Molto lieto, grazie.”, contengono rispettivamente 2, 5 e 3 parole.

```
[PAROLE]
numero_parole_stringa_1
numero_parole_stringa_2
...
```

5 Marcatura carattere

Per ciascuna stringa di input, sostituire ogni carattere corrispondente al primo carattere della stringa con un carattere '.' (punto). Per esempio, la stringa “io sono la stringa da sostituRe” inizia con il carattere 'i' (minuscolo), quindi tutti i caratteri i vanno sostituiti col punto. In questo caso il risultato è la stringa “.o sono la str.nga da sost.tuRe”. Si noti che la i maiuscola non è stata sostituita.

Stampare le stringhe così generate col seguente formato:

```
[MARCATURA]
stringa_1_marcata
stringa_2_marcata
...
```

Note

- salvare il proprio programma nella directory di lavoro
- assegnare il nome del file in base al proprio cognome, chiamandolo **cognome.c**
- il primo commento del programma deve riportare **nome e cognome e numero di matricola**
- vengono valutati positivamente aspetti quali la leggibilità del programma, una buona formattazione del sorgente, l'uso appropriato dei commenti, modularità e generalità del codice
- è possibile far uso di manuali, testi, appunti e dispense, ma non di eserciziari (raccolte di esercizi risolti)