Fondamenti di Informatica 29/01/2019

Contesto

Un file contiene un elenco di parole. Ciascuna parola è composta al massimo da 20 caratteri. Sono presenti soltanto caratteri alfabetici minuscoli. Ogni riga del file contiene esattamente due parole.
Un esempio di contenuto del file il seguente:

```
informatica fondamenti
aiuola abcd
qwertytrewq senzasignificato
altra riga
ultima riga
```

ATTENZIONE: le parole non sono necessariamente di senso compiuto.

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 specificati di seguito. 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 finale dovrà produrre la stampa di risultati esattamente col formato specificato nei vari punti. In particolare, *non aggiungere all'output del testo non richiesto*.
Eventuali righe di output aggiuntive che si vogliono generare in fase di debug, ma che si vogliono escludere dai test, possono essere stampate includendo in prima posizione il carattere $\#$.

Il buon funzionamento del programma può essere verificato col comando

```
./pvcheck ./a.out
```

dove `a.out` è il nome del file eseguibile.

RICHIESTE

1 Numero massimo di vocali

Determinare il massimo numero di vocali M contenute in una singola parloa. Le vocali sono le lettere 'a', 'e', 'i', 'o', 'u'.
Stampare il valore `M` col seguente formato:

```
[MAX-VOCALI]
M
```

2 Palindromi

Una parola si dice *palindroma* se rimane invariata quando viene letta in senso inverso. Ad esempio, sono parole palindrome "RADAR" e "OSSO".
Determinare il numero di parole N che sono palindrome tra tutte quelle presenti nel file. Stampare il valore di N

```
[PALINDROMI]
N
```

3 Filtro dei caratteri

Si considerino le prime 4 parole e le ultime 4 parole presenti nel file.
Ogni parola deve essere stampata eliminando i caratteri 'a', 'b', 'c', 'd', ed 'e'. Se una parola è composta soltanto da lettere da eliminare, non venga stampata.
Le parole devono essere stampate nell'ordine in cui compaiono nel file, una per riga. Se il file contiene meno di 4 righe, si stampino tutte le parole.
Esempio di stampa (usando i dati di esempio):

```
[FILTRO]
informti
fonmnti
iuol
ltr
rig
ultim
rig
```

4 Duplicati

Determinare se esiste almeno una parola che compare nel file due o più volte.
Stampare, con il formato riportato sotto, `SI` oppure `NO` a seconda che sia stato trovato un duplicato o meno. Il seguente esempio stampa `NO`:

```
[DUPLICATI]
NO
```

5 Ordinamento

Stampare tutte le parole lette ordinandole in senso crescente. Stampare una parola per riga. Si stampino anche le parole eventualmente duplicate.
Stampare l'elenco ordinato col seguente formato (vengono usati i dati di esempio):

```
[ORDINAMENTO]
abcd
aiuola
altra
fondamenti
informatica
qwertytrewq
riga
riga
senzasignificato
ultima
```

..:: CONTINUA SULL'ALTRO LATO ::..

Esempio di output

```
[MAX-VOCALI]
7

[PALINDROMI]
1

[FILTRO]
informti
fonmnti
iuol
ltr
rig
ultim
rig

[DUPLICATI]
SI

[ORDINAMENTO]
abcd
aiuola
altra
fondamenti
informatica
qwertytrewq
riga
riga
senzasignificato
ultima
```

Note

- salvare il proprio programma nella directory di lavoro
- assegnare il nome del file in base al proprio cognome, chiamandolo **cognome.c** (es. **facchinetti.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)