

## Contesto

Si consideri un ventilatore da soffitto come quello rappresentato nella figura sotto a sinistra. Il ventilatore, che integra una lampada, viene controllato tramite il telecomando mostrato nella figura sotto a destra.



Il telecomando dispone di 7 pulsanti con le seguenti funzionalità:

- Spegnimento della ventola (pulsante OFF).
- Accensione/spegnimento della luce (pulsante LIGHT); se la luce accesa, la pressione del pulsante la spegne e viceversa.
- Accensione della ventola a varie velocità: bassa (pulsante (L)ow), media (pulsante M) o alta (pulsante (H)igh).
- Impostazione di un timer per lo spegnimento automatico della ventola dopo 1 ora dalla pressione del pulsante “1h” oppure dopo 3 ore dalla pressione del pulsante “3h”. Se il timer già impostato, la pressione di uno dei due tasti di spegnimento temporizzato fa ripartire il timer da zero. Inoltre, se il pulsante del timer viene premuto quando la ventola è spenta, il suo effetto è nullo.

Il ventilatore è collocato in un sistema di automazione domestica che riceve e registra i comandi impostati tramite il telecomando. I dati vengono registrati in un file di testo, un comando per riga. Il formato della singola riga è il seguente:

timestamp comando

dove

- timestamp rappresenta il *numero di millisecondi* trascorsi dalla mezzanotte del giorno corrente.
- comando è una stringa associata al nome del pulsante nell'insieme { OFF, LIGHT, L, M, H, 1h, 3h }, case-sensitive.

Il file contiene i dati di un singolo, intero giorno di registrazioni. Le registrazioni sono in ordine crescente di timestamp. Si consideri che a mezzanotte, istante di inizio della registrazione, il ventilatore ha la *luce accesa* e la *ventola spenta*.

Un esempio di file contenente tre misurazioni è il seguente:

```
10000 LIGHT
15050 L
20132 1h
21000 OFF
```

## 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 Pulsante maggiormente premuto**

Determinare il pulsante che registra il maggior numero di pressioni nel giorno di registrazione. In caso per più di un pulsante si registri un numero di pressioni pari al massimo, stampare solo quello la cui registrazione compare per prima nel file.

Riportare il nome del pulsante, usando una delle stringhe associate al nome del pulsante, col seguente formato:

```
[MAX-PRESSIONI]  
comando
```

### **2 Tempo medio di accensione della luce**

Calcolare la durata del tempo di accensione della luce nel giorno di registrazione. Arrotondare al secondo e all'intero più vicino per difetto il valore calcolato. Stamparne il valore con il seguente formato:

```
[MEDIA-LUCE]  
MEDIA
```

Se l'ultimo comando di tipo LIGHT presente nel file corrisponde all'accensione della luce, considerare che questa rimanga accesa fino alla mezzanotte esatta del giorno stesso.

### **3 Tempo totale di accensione della ventola a velocità alta**

Calcolare il tempo totale di accensione della ventola a velocità alta (comando H) nel giorno di registrazione. Si consideri di trascurare l'effetto dei timer di spegnimento, ovvero non considerare l'effetto di tali comandi nel funzionamento della ventola. Stamparne il valore con il seguente formato:

```
[TOT-ALTA-VELOCITA]  
DURATA
```

Se dopo l'ultimo comando di tipo H presente nel file non viene più dato un comando che modifichi la velocità di rotazione della ventola, o la spenga, considerare che la ventola rimanga accesa a velocità alta fino alla mezzanotte esatta del giorno stesso.

### **4 Tempo totale di accensione della ventola, indipendentemente dalla velocità**

Calcolare il tempo totale di accensione della ventola nel giorno di registrazione, indipendentemente dalla velocità impostata. Si consideri di trascurare l'effetto dei timer di spegnimento, ovvero non considerare l'effetto di tali comandi nel funzionamento della ventola. Stamparne il valore con il seguente formato:

```
[TOT-ACCENSIONE]  
DURATA
```

Se dopo l'ultimo comando di accensione della ventola non viene più dato un comando di spegnimento, considerare che la ventola rimanga accesa fino alla mezzanotte esatta del giorno stesso.

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

## 5 Ordinamento

Ordinare alfabeticamente (tramite `strcmp`) le misurazioni in senso crescente rispetto al comando. In caso più misurazioni siano presenti per lo stesso comando, ordinare tali misurazioni in senso crescente di timestamp.

Riportare le righe ordinate con lo stesso formato del file di ingresso:

```
[ORDINAMENTO]
timestamp_1 comando_1
...
timestamp_n comando_n
```

Si consideri che non vi possono essere due o più comandi aventi lo stesso timestamp.

### 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)