

Esercizio 1

(1) Esercizio 1 v1

ESSAY marked out of 10 penalty 0 File picker

Una programma per smartphone é stato ideato per monitorare il tempo che un utente trascorre nell'arco di una giornata (24 ore) su ogni app installata nello smartphone. Il programma salva la durata e l'identificativo numerico dell'app utilizzata in un file chiamato `Log.txt`. Il file `Log.txt` é sempre composto da due colonne: nella prima é riportata la durata dell'utilizzo di una determinata app in millisecondi (un intero), nella seconda l'identificativo numerico dell'app (un intero tra 0 e 8 estremi inclusi). Ogni riga del file, quindi, riporta per quanto tempo é stata usata una certa app in un certo momento della giornata e il suo identificativo. Si assuma che gli identificativi delle app siano corrette, ovvero che abbiano un valore compreso tra 0 e 8 estremi inclusi). L'identificativo e il nome delle app sono qui riportate:

0: WhatsApp
1: ilMeteo
2: Spotify
3: ESSE3
4: GialloZafferano
5: La Gazzetta dello Sport
6: Telegram
7: MyUnitn
8: YouTube

Si scriva un programma C++ che apra il file `Log.txt` e ne analizzi il contenuto. In particolare, il programma deve eseguire le seguenti operazioni (nell'ordine indicato):

- (a) Chiami una funzione `Occorrenze` (che sar  vostra cura definire) che riceva come parametro un nome di file (qui `Log.txt`) e un array `count` di 9 elementi di tipo `int` (in cui ogni elemento é inizializzato a 0). La funzione apra il file, lo legga, e conti quante occorrenze di ogni identificatore ci sono popolandolo l'array `count`. La funzione ritorna il valore massimo `m` contenuto in `count`. Se il file non esiste la funzione termina immediatamente e ritorna -1. Il programma in questo caso stampa un messaggio di errore e termina.
- (b) Stampi a video il contenuto dell'array `count`.
- (c) Allochi dinamicamente una matrice `dati` di 9 righe e `m` colonne e la inizializzi a zero. Ogni riga corrisponde a una web app e ogni colonna a una durata di utilizzo;
- (d) Apra il file `Log.txt` e copi il contenuto nella matrice `dati`. Le durate contenute nella matrice `dati` sono numeri reali che indicato le durate espresse in minuti. Per convertire le durate da millisecondi in minuti si definisca e si usi una funzione `Converti` che riceva come parametro un intero che esprime la durata di utilizzo in millisecondi e ritorni un numero reale che esprime la stessa durata espressa in minuti.
- (e) Stampi a video l'identificatore delle web app usate stretta e te pi  di 5 minuti e strettamente meno di 30 minuti.
- (f) Stampi a video un messaggio di allerta nel caso in cui l'utilizzo complessivo delle web app sia superiore a 60 minuti.

NON SI SCRIVANO ALTRE FUNZIONI RISPETTO A QUELLE RICHIESTE NEL TESTO, PENA ANNULLAMENTO DELL'ESERCIZIO.

Di seguito é riprotato un esempio di esecuzione del programma.

```
computer > ./a.out
3 1 2 1 1 1 2 2 1
App 1
App 3
App 7
Alert!
computer >
```

Note:

- Scaricare il file `esercizio1.cpp`, modificarlo per inserire il codice necessario per rispondere a questo esercizio. **Caricare il file sorgente risultato delle vostre modifiche a soluzione di questo esercizio** nello spazio apposito.
- All'interno di questo programma **non è ammesso** l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream`, `fstream`, `cmath`.
- Si ricorda che, gli esempi di esecuzione sono puramente indicativi, e la soluzione proposta NON deve funzionare solo per l'input fornito, ma deve essere robusta a variazioni compatibili con la specifica riportata in questo testo.
- Si ricorda di inserire solo nuovo codice e di **NON MODIFICARE** il resto del programma (pena annullamento dell'esercizio).
- Si ricorda che la soluzione deve essere implementata in C++ **NON usando `string` e altri elementi della C++ standard template library, anche se il file compila senza cambiare gli header!**

`esercizio1.cpp`

`Log.txt`

Information for graders:

Total of marks: 10