

# Laboratorio di Architettura degli Elaboratori

## Elaborato Assembly

A.A. 2021/2022

### Descrizione:

Si descriva un programma che simuli il sistema di telemetria del videogame F1.

Il sistema fornisce in input i dati grezzi di giri motore (rpm), temperatura motore e velocità di tutti i piloti presenti in gara per ogni istante di tempo.

Ogni campo è diviso da una virgola usata come separatore.

Ogni riga del file di input è così composta:

<tempo>,<id\_pilota>,<velocità>,<rpm>,<temperatura>

...

Id pilota rappresenta un valore numerico che identifica univocamente un pilota.

L'associazione tra id e nome del pilota è il seguente:

ID	Nome
0	Pierre Gasly
1	Charles Leclerc
2	Max Verstappen
3	Lando Norris
4	Sebastian Vettel
5	Daniel Ricciardo
6	Lance Stroll
7	Carlos Sainz
8	Antonio Giovinazzi
9	Kevin Magnussen
10	Alexander Albon
11	Nicholas Latifi
12	Lewis Hamilton
13	Romain Grosjean
14	George Russell
15	Sergio Perez
16	Daniil Kvyat
17	Kimi Raikkonen
18	Esteban Ocon
19	Valtteri Bottas

Nella prima riga del file di input è presente il nome di un pilota che si vuole monitorare

## **Obiettivo:**

Si scriva un programma in assembly che restituisca i dati relativi al solo pilota indicato nella prima riga del file, in base a delle soglie indicate.

Vengono definite tre soglie per tutti i dati monitorati: LOW,MEDIUM,HIGH.

Il file di output dovrà riportare queste soglie per tutti gli istanti di tempo in cui il pilota è monitorato.

Le righe del file di output saranno strutturate nel seguente modo e ordine:

<tempo>,<livello rpm>,<livello temperatura>,<livello velocità>

Inoltre, viene richiesto di aggiungere alla fine del file di output una riga aggiuntiva che contenga, nel seguente ordine: il numero di giri massimi rilevati, la temperatura massima rilevata, la velocità di picco e infine la velocità media.

La struttura dell'ultima riga sarà quindi la seguente:

<rpm max>,<temp max>,<velocità max>,<velocità media>

Le soglie per i dati monitorati sono così definite:

- **Giri Motore**
  - LOW: rpm  $\leq$  5000
  - MEDIUM:  $5000 < \text{rpm} \leq 10000$
  - HIGH: rpm  $> 10000$
- **Temperatura**
  - LOW: temp  $\leq$  90
  - MEDIUM:  $90 < \text{temp} \leq 110$
  - HIGH: temp  $> 110$
- **Velocità**
  - LOW: speed  $\leq$  100
  - MEDIUM:  $100 < \text{speed} \leq 250$
  - HIGH: speed  $> 250$

## Esempio:

Prendiamo come esempio un file di input dove ci sono solo 3 piloti, due campioni di tempo e dove viene richiesto di monitorare Charles Leclerc.

### File di input :

```
Charles Leclerc
0.01023,0,0,3505,90
0.01023,1,5,4305,89
0.01023,2,0,3505,90
0.02042,0,0,3507,90
0.02042,1,10,5001,100
0.02042,2,0,3507,90
```

### File di output:

```
0.01023, LOW, LOW, LOW
0.02042, MEDIUM, MEDIUM, LOW
4307, 100, 10, 7
```

L'id di Charles Leclerc è 1 (vedi tabella). Il sistema deve filtrare le righe andando a considerare solo quelle in cui l'id è quello di Charles Leclerc.

## Requisiti & Vincoli:

Le istanze che verranno usate non contengono errori nei campi.

La velocità media viene calcolata come quoziente intero della divisione.

L'ordine dei campi del file di output DEVE seguire esattamente le specifiche.

Se il nome del pilota inserito non è valido il programma deve restituire la stringa scritta esattamente nel seguente modo: `Invalid`

Il programma deve essere lanciato da riga di comando con due stringhe come parametri, la prima stringa identifica il nome del file .txt da usare come input, la seconda quello da usare come output:

```
$ ./telemetry input.txt output.txt
```

Il programma deve leggere il contenuto di `input.txt` e restituire il risultato della espressione in `output.txt`.

Il file `input.txt` contiene il nome pilota nella prima riga e tutti i campioni nelle righe successive.

Il file `output.txt` dovrà contenere solamente le righe del pilota selezionato e infine la riga con i valori massimi e la media della velocità.

Assieme al presente documento sono forniti i seguenti files:

- `main.c`: Contenente il sorgente C che chiama la funzione `assembly` e che non può essere editato. La modifica di tale file esclude automaticamente la valutazione dell'elaborato.
- `Istanze.tar.gz`: Contiene le istanze di test e i relativi risultati.
  - `in_X.txt`: istanza di test in input al programma. In fase di valutazione l'elaborato verrà testato con diverse istanze di diversa complessità.
  - `out_X.txt` da utilizzare per verifica del corretto funzionamento del programma con l'espressione `in_X.txt`. Sarà sufficiente usare il comando  

```
$ diff <vostro_output>.txt out_X.txt
```

per vedere eventuali differenze tra i due files (quello generato dal programma e quello corretto).

Durante l'esame orale è possibile che venga richiesto di operare delle modifiche al codice sul momento.

### **Materiale da consegnare:**

Il codice e la relazione vanno compressi in un unico file tarball (`.tar.gz`) denominato `VRXXXXXX_VRXXXXXX.tar.gz`, come precedentemente specificato nel progetto SIS.

Devono essere presenti:

- Sorgenti dell'intero progetto:
  - a. Il file `assembly` principale si dovrà chiamare `telemetry.s`
  - b. Il file C, contenente il `main` e la chiamata `assembly`, dovrà chiamarsi `main.c`
- Relazione in formato pdf denominata `Relazione.pdf`, che affronti nel dettaglio almeno i seguenti punti:
  - a. le variabili utilizzate e il loro scopo;
  - b. le modalità di passaggio/restituzione dei valori delle funzioni create;

- c. il diagramma di flusso o lo pseudo-codice ad alto livello del codice prodotto;
- d. la descrizione delle scelte progettuali effettuate.

Il pacchetto deve contenere un'unica cartella denominata asm/ la cui struttura dovrà essere la seguente:

- asm/
  - src/
    - telemetry.s ed eventuali altri file di altre funzioni assembly
    - main.c non editato
  - obj/
    - cartella di supporto per la creazione dei file oggetto (vuota)
  - bin/
    - cartella dove verrà creato l'eseguibile (vuota)
  - Makefile
    - File Makefile per la catena di compilazione. File oggetto generati in obj/ ed eseguibile generato in bin/
  - Relazione.pdf
    - File della relazione

### **Per gli studenti che consegnano entrambi gli elaborati:**

Verrà aperta una sezione su moodle per la consegna completa (sis+asm).

Il nome del file rimane invariato (VRXXXXXX\_VRXXXXXX.tar.gz).

La struttura del file deve essere la seguente:

- sis/
  - Contiene tutte le sottocartelle ed i files come definito da progetto SIS
- asm/
  - Contiene tutte le sottocartelle ed i files come definito da progetto ASM

### **Modalità di consegna:**

Tutto il materiale va consegnato elettronicamente tramite procedura guidata sul sito Moodle del corso. Indicativamente 15 giorni prima della data di consegna sarà attivata una apposita sezione denominata "consegna\_ASM\_mmmaaaa" (mmm=mese, aaaa=anno); accedendo a quella pagina sarà possibile effettuare l'upload del materiale. La consegna del materiale comporta automaticamente l'iscrizione all'appello orale.

Si ricorda che è possibile effettuare più sottomissioni, ma ogni nuova sottomissione cancella quella precedente. Ogni gruppo deve consegnare una sola volta il materiale, ovvero un solo membro del gruppo deve effettuare la sottomissione!

### **Note importanti:**

1. È possibile effettuare più sottomissioni, ma ogni nuova sottomissione cancella quella precedente.
2. Un solo membro del gruppo deve effettuare la sottomissione.
3. Tutti i componenti del gruppo devono essere iscritti alla pagina Moodle del corso.
4. Non si accettano progetti consegnati via mail e/o dopo la scadenza.
5. I progetti che non soddisfano tutti i requisiti sopraelencati e che non rispondono correttamente alle istanze di test contenute in [istanze.tar.gz](#) non verranno ammessi all'orale e non verranno valutati.
6. **Tutti i progetti verranno testati automaticamente. Solo i progetti che supereranno i test saranno ammessi alla discussione orale.**
7. I progetti non ammessi potranno essere visionati e discussi al termine della sessione su richiesta degli studenti.