

Elaborato Assembly

Laboratorio Architettura degli Elaboratori

Corso di laurea in Informatica

Michele Cipriani (VR471337) - Alex Gaiga (VR471343)
Tommaso Vilotto (VR471487)

A.A. 2021/2022

Indice

<u>Descrizione progetto e specifiche</u>	3
<u>Principali variabili usate</u>	5
<u>Flusso del Programma</u>	6
<u>Descrizioni scelte progettuali effettuate</u>	7

Descrizione progetto e specifiche

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: `,,, ...` 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

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: `,,,` 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: ,, Le soglie per i dati monitorati sono così definite:

- **Giri Motore**

- o LOW: rpm \leq 5000
- o MEDIUM: $5000 < \text{rpm} \leq 10000$
- o HIGH: rpm > 10000

- **Temperatura**

- o LOW: temp \leq 90
- o MEDIUM: $90 < \text{temp} \leq 110$
- o HIGH: temp > 110

- **Velocità**

- o LOW: speed \leq 100
- o MEDIUM: $100 < \text{speed} \leq 250$
- o 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
5001,100,10,7

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.

Principali variabili usate

file **telemetry.s**:

pilot_0 ... 19_str: stringa contenente il nome di ogni pilota dal 0 al 19

pilot_0 ... 19_str_len: long che contiene la lunghezza della stringa contenente il nome del pilota

invalid_pilot_str: stringa che contiene il messaggio di errore in caso il pilota del file di input sia invalido

rpmMax: long che contiene gli rpm massimi tra i dati inseriti

tempMax: long che contiene la temperatura massima tra i dati inseriti

velMax: long che contiene la velocità massima tra i dati inseriti

velocitaMedia: long che contiene la media tra tutte le velocità

checkRigheTrovate: long che contiene il numero di righe corrispondenti con l'id del pilota inserito nel file di input

velocita: long che contiene la velocità che viene aggiornato ad ogni riga che cicliamo che ha il record dell'id interessato

rpm: long che contiene la velocità che viene aggiornato ad ogni riga che cicliamo ha il record dell'id interessato

temperatura: long che contiene la velocità che viene aggiornato ad ogni riga che ha il record dell'id interessato

tempo1: stringa contenente il valore del tempo del record che si sta andando a leggere, viene aggiornata ad ogni riga

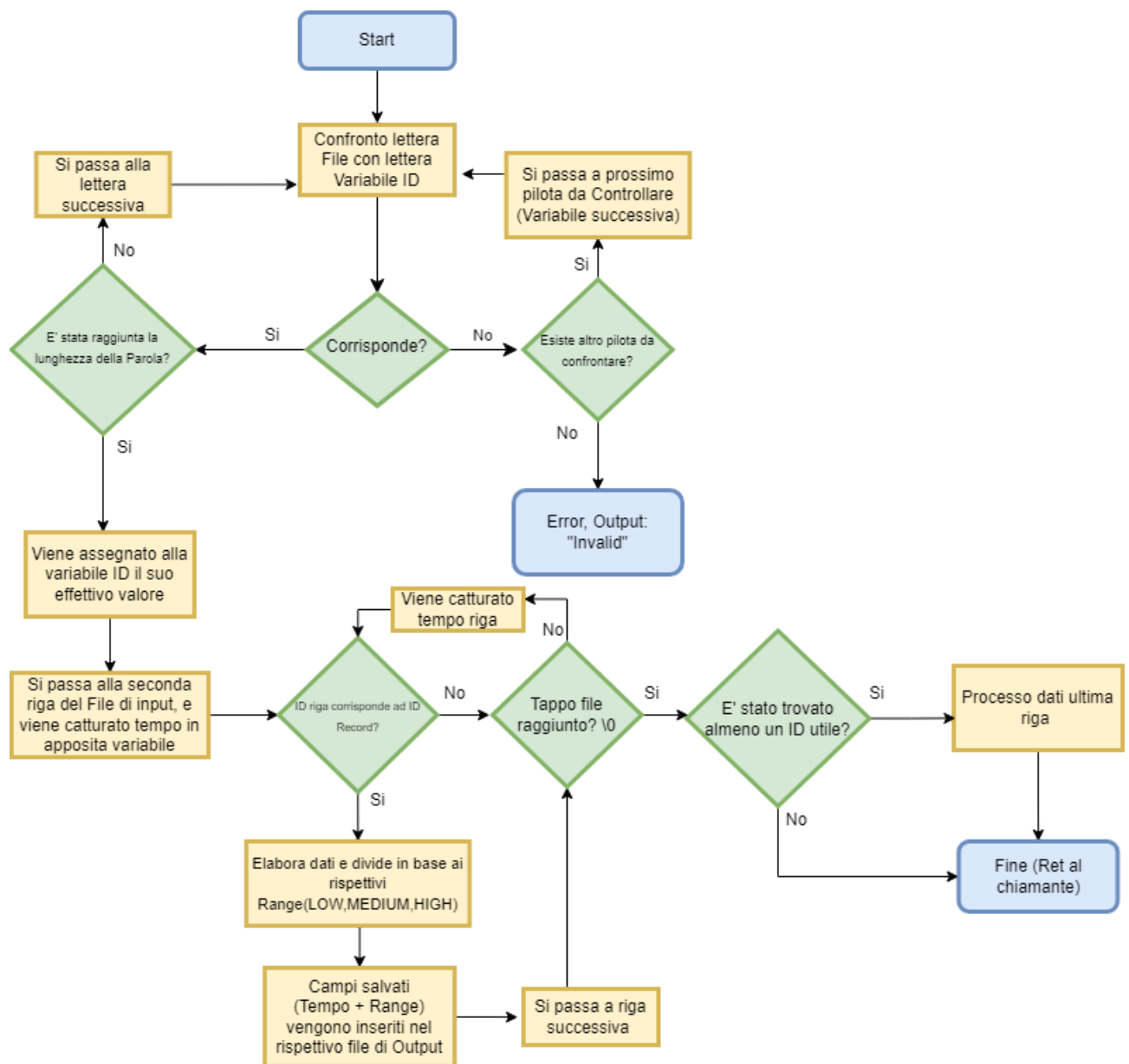
tempo1_len: long che contiene lunghezza della stringa tempo1

tempCOUNTER: long contenente valori di dati temporanei da decodificare

megaTEMP: stringa contenente valori di dati temporanei da decodificare, usata a braccetto con tempCOUNTER, viene aggiornata ad ogni riga che ha il record dell'id interessato

megaTEMP_len: long che contiene lunghezza della stringa megaTemp

Flusso del Programma



Descrizione scelte progettuali effettuate

Cercando di aumentare l'efficienza e le performance del programma è stato scelto di andare a prelevare e decodificare i valori dei record interessati solo dopo aver controllato l'ID. Quindi qualora venisse identificato un ID non interessato l'intera riga verrebbe scartata, evitando la relativa decodifica e il conseguente salvataggio.

Inoltre abbiamo aggiunto 4 funzioni per alleggerire le righe del programma ed aumentare la leggibilità:

- stampaHIGH
- stampaMEDIUM
- stampaLOW
- stampaVIRGOLA

per la stampa nel file rispettivamente di "HIGH", "MEDIUM", "LOW" e ",".