

Università di Catania
Dipartimento di Matematica e Informatica
Corso di Studio in Informatica, A.A. 2024-2025
Compito di Programmazione 1 e Laboratorio F-N
22 settembre 2025

Descrizione del programma

Si scriva un programma in linguaggio C che esegua le seguenti operazioni:

- A. [5 punti] Il programma deve accettare tre parametri da riga di comando:
- a. `input_filename`: il nome di un file di input (es. `dati.tsv`). Il programma deve verificare che l'estensione del file sia `.tsv`.
 - b. `output_filename`: il nome di un file di output (es. `report.txt`). Il programma deve verificare che l'estensione del file sia `.txt`.
 - c. `tipo_sensore`: un singolo carattere che identifica il tipo di sensore da analizzare. Il programma deve verificare che il carattere sia una delle seguenti lettere maiuscole: 'T' (Temperatura), 'U' (Umidità), o 'P' (Pressione).

Se i parametri forniti non rispettano i requisiti, il programma dovrà stampare un messaggio di errore sullo standard error e terminare con un codice di uscita appropriato.

Stampare a schermo i parametri acquisiti correttamente.

- B. [10 punti] Il file di input (`.tsv`) contiene dati di rilevazioni di sensori. Ogni riga rappresenta una rilevazione ed è formattata come segue, con i campi separati da un carattere di tabulazione (`\t`):

```
ID_Sensore (int) <tab> Timestamp (long int) <tab> Valore (double) <tab> Tipo (char)
```

Il programma deve leggere tutte le righe dal file di input e memorizzare ciascuna rilevazione in una struct del tipo definito di seguito:

```
typedef struct {
    int id_sensore;
    long int timestamp;
    double valore;
    char tipo;
    int anomalia; // Flag per marcare le anomalie (0 o 1)
} Rilevazione;
```

Tali strutture devono essere inserite in una lista concatenata singolarmente, inserendo ogni nuovo elemento in testa alla lista.

Si stampi a schermo il contenuto della lista dopo aver completato l'inserimento.

- C. [8 punti] Scorrere la lista creata al punto B e calcolare il valore medio (`media_valori`) di tutte le rilevazioni il cui campo `Tipo` corrisponde a quello fornito come parametro (`tipo_sensore`).

Si stampi a schermo la media calcolata.

- D. [2 punti] Scorrere nuovamente la lista e "marcare" (impostando a 1 l'apposito campo `anomalia` nella struct) ogni rilevazione dello stesso tipo (`tipo_sensore`) il

cui Valore sia strettamente maggiore del 150% della media calcolata (ovvero, Valore > media_valori * 1.5).

Si stampi a schermo l'intera lista, evidenziando le rilevazioni che sono state marcate come anomale.

- E. [5 punti] Scorrere la lista un'ultima volta e salvare nel file di output specificato **solo** le rilevazioni che sono state marcate come anomale nel punto precedente. Il formato di ogni riga nel file di output dovrà essere:

ID_Sensore, Timestamp, Valore, Tipo

Gestire opportunamente i casi in cui i file non possono essere aperti correttamente, stampando un messaggio di errore sullo standard error e terminando l'esecuzione.

Specifiche

Il programma dovrà essere articolato in un unico file sorgente, e contenere almeno le seguenti funzioni con opportuni parametri formali:

- **decodeParameters**: riceve argc e argv, valida i parametri e li restituisce tramite una opportuna struct.
- **readDataAndCreateList**: riceve il nome del file di input e restituisce la testa della lista.
- **calculateAverage**: riceve la testa della lista e il tipo di sensore, e restituisce la media.
- **markAnomalies**: riceve la testa della lista, la media e il tipo di sensore, e imposta il flag per ogni nodo che rappresenta un'anomalia.
- **writeAnomaliesToFile**: salva su file solo i nodi della lista marcati come anomali.

Note

- **Durata della prova:** 120 minuti
- **È VIETATO** usare variabili globali.
- **Si inseriscano i file sorgenti** direttamente nella propria home directory .
- **Accesso alla documentazione** disponibile tramite il browser al link: <https://devdocs.io/c/>.
- **È possibile ritirarsi inserendo la scritta “RITIRATO” come prima riga di ogni file sorgente.**

Output di controllo

Si consideri il seguente file dati_sensori.tsv:

101	1667222400	220.5	T
205	1667222415	605.1	U
102	1667222430	35.8	T
301	1667222445	1012.5	P
101	1667222460	203.1	T
205	1667222475	64.8	U
102	1667222490	22.9	T
301	1667222505	101.1	P
101	1667222520	8.5	T

Eseguendo il programma con il comando:

```
./soluzione dati_sensori.tsv report_anomalie.txt T
```

Il programma stamperà:

```
===== PUNTO A: PARAMETRI =====
File di input: dati_sensori.tsv
File di output: report_anomalie.txt
Tipo di sensore da analizzare: T

===== PUNTO B: LISTA CREATA =====
[ID: 101, TS: 1667222520, Val: 8.50, Tipo: T] ->
[ID: 301, TS: 1667222505, Val: 101.10, Tipo: P] ->
[ID: 102, TS: 1667222490, Val: 22.90, Tipo: T] ->
[ID: 205, TS: 1667222475, Val: 64.80, Tipo: U] ->
[ID: 101, TS: 1667222460, Val: 203.10, Tipo: T] ->
[ID: 301, TS: 1667222445, Val: 1012.50, Tipo: P] ->
[ID: 102, TS: 1667222430, Val: 35.80, Tipo: T] ->
[ID: 205, TS: 1667222415, Val: 605.10, Tipo: U] ->
[ID: 101, TS: 1667222400, Val: 220.50, Tipo: T] ->
NULL
```

```
===== PUNTO C: CALCOLO MEDIA =====
Media dei valori per il sensore di tipo 'T': 98.16
```

```
===== PUNTO D: MARCatura ANOMALIE =====
Lista con anomalie evidenziate:
[ID: 101, TS: 1667222520, Val: 8.50, Tipo: T] ->
[ID: 301, TS: 1667222505, Val: 101.10, Tipo: P] ->
[ID: 102, TS: 1667222490, Val: 22.90, Tipo: T] ->
[ID: 205, TS: 1667222475, Val: 64.80, Tipo: U] ->
[ID: 101, TS: 1667222460, Val: 203.10, Tipo: T] **ANOMALIA** ->
[ID: 301, TS: 1667222445, Val: 1012.50, Tipo: P] ->
[ID: 102, TS: 1667222430, Val: 35.80, Tipo: T] ->
[ID: 205, TS: 1667222415, Val: 605.10, Tipo: U] ->
[ID: 101, TS: 1667222400, Val: 220.50, Tipo: T] **ANOMALIA** ->
NULL
```

```
===== PUNTO E: SCRITTURA FILE =====
File report_anomalie.txt scritto correttamente.
```

Contenuto del file report_anomalie.txt:

```
101,1667222460,203.10,T
101,1667222400,220.50,T
```