

## Esame 20250616

### Esercizio 1

#### (1) Esercizio 2 v1

ESSAY marked out of 10 penalty 0 File picker

Sei stato assunto da un'azienda che produce smartwatch e sei incaricato di sviluppare una parte del software per gestire tale dispositivo. Uno dei componenti fondamentali del dispositivo è l'accelerometro, un sensore in grado di misurare l'accelerazione lungo i tre assi X, Y, Z. Queste tre accelerazioni combinate insieme attraverso operatori matematici sono utilizzate dallo smartwatch per il conteggio dei passi o altre funzionalità. Le accelerazioni sui tre assi X, Y, Z sono rappresentate con numeri reali nell'intervallo  $[0,1]$  e salvati in un file di testo di nome `Test.txt`. La struttura del file, tranne nel caso in cui il dispositivo produca un errore, è sempre la seguente: ci sono tre colonne di numeri reali, ognuna delle quali è relativa all'accelerazione su uno dei tre assi. La prima colonna è relativa all'asse X, la seconda all'asse Y, la terza all'asse Z. La lunghezza del file è nota a priori: ogni colonna contiene 30 elementi. Se il dispositivo produce un errore, il file `Test.txt` sarà completamente vuoto.

Scrivere un programma C++ che:

- Apre il file `Test.txt`. Se il file non esiste, scriva un messaggio di errore e termini.
- Legga il primo valore contenuto nel file, se esso è EOF scriva un messaggio di errore e termini. Per verificare che un valore sia uguale a EOF si usi la funzione `eof()` invocata sullo stream (esempio: `finput.eof()`). Altrimenti legga gli altri due valori della prima riga e poi tutte le righe successive e li salvi nella matrice `campioni` avente 30 righe e 3 colonne.
- Chiami la funzione `calcola_modulo` che calcola il modulo dei valori dell'accelerazione contenuti nella matrice `campioni`. La funzione `calcola_modulo` prende come parametro la matrice `campioni` e un array di 30 elementi reali e chiamato `modulo_acc`. La funzione non restituisce alcun valore. Per ogni riga  $i$  della matrice `campioni`, il modulo dell'accelerazione viene calcolato usando la seguente formula:  $a_i = \sqrt{x_i^2 + y_i^2 + z_i^2}$ . Dove  $x_i$ ,  $y_i$  e  $z_i$  sono i valori di accelerazione contenuti alla riga  $i$ -esima della matrice `campioni`. Per il calcolo della radice quadrata e della potenza si possono usare le funzioni nella libreria `cmath`.
- Chiami la funzione `normalizza` che prende come parametro due array di numeri reali `modulo_acc` e `modulo_acc_norm` di 30 elementi. La funzione `normalizza` i valori contenuti in `modulo_acc` nell'intervallo  $[0,1]$ , ovvero cerca il valore massimo (`val_max`) nell'array `modulo_acc` e calcola ciascun valore dell'array `modulo_acc_norm` dividendo il corrispondente valore di `modulo_acc` per `val_max`. La funzione non ritorna alcun valore. Nota: si assuma che il valore massimo sia sempre diverso da zero.
- Chiami la funzione `quantizza` che presi come parametri l'array `modulo_acc_norm`, un array di interi `q` con dimensione 30 ed un numero intero `n` passato per valore, converte ogni valore dell'array `modulo_acc` in un numero intero  $q_i$  esprimibile con un numero `n` di bit uguale a 8. Tale operazione tecnicamente ha il nome di quantizzazione e si realizza tramite la formula seguente:  $q_i = \text{round}(a_i \cdot (2^{n-1}))$ . Si usino la funzione `round` la funzione `pow` della libreria `cmath`. La funzione `quantizza` non ritorna alcun valore.
- Stampa a video il contenuto dell'array `q` e termina.

**NON SI SCRIVANO ALTRE FUNZIONI RISPETTO A QUELLE RICHIESTE NEL TESTO.**

Di seguito é riprotato un esempio di esecuzione del programma.

```
computer > ./a.out
114 82 51 100 95 110 81 56 118 87 84 93 128 84 108 54 56 100 107 114 69 93 106 95 100 83
124 80 83 55
computer > rm Test.txt
computer > ./a.out
File non esiste!
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](#)

[Test.txt](#)

*Information for graders:*

## (2) Esercizio 2 v2

ESSAY

marked out of 10

penalty 0

File picker

Sei stato assunto da un'azienda che produce smartwatch e sei incaricato di sviluppare una parte del software per gestire tale dispositivo. Uno dei componenti fondamentali del dispositivo è l'accelerometro, un sensore in grado di misurare l'accelerazione lungo i tre assi X, Y, Z. Queste tre accelerazioni combinate insieme attraverso operatori matematici sono utilizzate dallo smartwatch per il conteggio dei passi o altre funzionalità. Le accelerazioni sui tre assi X, Y, Z sono rappresentate con numeri reali nell'intervallo  $[0,1]$  e salvati in un file di testo di nome `Test.txt`. La struttura del file, tranne nel caso in cui il dispositivo produca un errore, è sempre la seguente: ci sono tre colonne di numeri reali, ognuna delle quali è relativa all'accelerazione su uno dei tre assi. La prima colonna è relativa all'asse X, la seconda all'asse Y, la terza all'asse Z. La lunghezza del file è nota a priori: ogni colonna contiene 40 elementi. Se il dispositivo produce un errore, il file `Test.txt` sarà completamente vuoto.

Scrivere un programma C++ che:

- Apre il file `Test.txt`. Se il file non esiste, scriva un messaggio di errore e termini.
- Legga il primo valore contenuto nel file, se esso è EOF scriva un messaggio di errore e termini. Per verificare che un valore sia uguale a EOF si usi la funzione `eof()` invocata sullo stream (esempio: `fininput.eof()`). Altrimenti legga gli altri due valori della prima riga e poi tutte le righe successive e li salvi nella matrice `campioni` avente 40 righe e 3 colonne.
- Chiami la funzione `calcola_modulo` che calcola il modulo dei valori dell'accelerazione contenuti nella matrice `campioni`. La funzione `calcola_modulo` prende come parametro la matrice `campioni` e un array di 40 elementi reali e chiamato `modulo_acc`. La funzione non restituisce alcun valore. Per ogni riga  $i$  della matrice `campioni`, il modulo dell'accelerazione viene calcolato usando la seguente formula:  $a_i = \sqrt{x_i^2 + y_i^2 + z_i^2}$ . Dove  $x_i$ ,  $y_i$  e  $z_i$  sono i valori di accelerazione contenuti alla riga  $i$ -esima della matrice `campioni`. Per il calcolo della radice quadrata e della potenza si possono usare le funzioni nella libreria `cmath`.
- Chiami la funzione `normalizza` che prende come parametro due array di numeri reali `modulo_acc` e `modulo_acc_norm` di 40 elementi. La funzione `normalizza` i valori contenuti in `modulo_acc` nell'intervallo  $[0,1]$ , ovvero cerca il valore massimo (`val_max`) nell'array `modulo_acc` e calcola ciascun valore dell'array `modulo_acc_norm` dividendo il corrispondente valore di `modulo_acc` per `val_max`. La funzione non ritorna alcun valore. Nota: si assuma che il valore massimo sia sempre diverso da zero.
- Chiami la funzione `quantizza` che presi come parametri l'array `modulo_acc_norm`, un array di interi `q` con dimensione 40 ed un numero intero `n` passato per valore, converte ogni valore dell'array `modulo_acc` in un numero intero  $q_i$  esprimibile con un numero `n` di bit uguale a 16. Tale operazione tecnicamente ha il nome di quantizzazione e si realizza tramite la formula seguente:  $q_i = \text{round}(a_i \cdot (2^n - 1))$ . Si usino la funzione `round` la funzione `pow` della libreria `cmath`. La funzione `quantizza` non ritorna alcun valore.
- Stampa a video il contenuto dell'array `q` e termina.

**NON SI SCRIVANO ALTRE FUNZIONI RISPETTO A QUELLE RICHIESTE NEL TESTO.**

Di seguito é riprotato un esempio di esecuzione del programma.

```
computer > ./a.out
23804 30949 23373 31982 27165 13837 22164 27258 30081 32469 7858 28524 30675 22420 29716
```

```
12847 28681 26064 23688 25041 24295 20080 24920 30800 27225 32768 23076 18641 23153 19468
28536 6484 21404 22757 22766 29281 21658 25853 27296 15606
computer > rm Text.txt
computer > ./a.out
File non esiste!
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`

`Test.txt`

*Information for graders:*

*Total of marks: 20*