

# Informatica – Prova di laboratorio, 9 febbraio 2024

## TEMPO D'ESAMI

La sessione d'esame di gennaio-febbraio del corso di Laurea (CdL) in Fisica dell'Università di Watopia si è appena conclusa: è il momento di aggiornare la carriera degli studenti.

I voti conseguiti dagli studenti che hanno sostenuto esami nella sessione di gennaio-febbraio, incluse le eventuali insufficienze (ovvero i voti  $< 18$ ), sono stati tutti registrati in un file `votiGF.dat`. In ciascuna riga di questo file sono stati registrati, in ordine: il codice dell'esame, il numero di crediti "dell'esame", la matricola dello studente, e il voto conseguito. Tutti questi valori sono interi (non negativi).

La segreteria studenti dell'Università di Watopia ha registrato i dati degli studenti iscritti al CdL in Fisica nel file `studenti.dat`. I dati sono aggiornati all'ultima sessione d'esame precedente quella di gennaio-febbraio. Su ciascuna riga del file sono indicati: la matricola dello studente, il numero di crediti complessivo degli esami sostenuti, la media pesata (vedi sotto) dei voti conseguiti e il numero di esami "falliti", ovvero quante volte uno studente ha sostenuto un esame e ha ottenuto un voto insufficiente. Tutti questi valori sono interi non negativi ad esclusione della media (pesata) che è un valore razionale (in singola precisione).

Il calcolo della media pesata di  $n$ -esami superati si ottiene a partire dalla  $n$ -upla di voti  $(v_1, v_2, \dots, v_n)$  e dalla corrispondente  $n$ -upla di crediti  $(c_1, c_2, \dots, c_n)$  usando la relazione:

$$m_n = \frac{\sum_{i=1}^n c_i \cdot v_i}{\sum_{i=1}^n c_i} = \frac{\sum_{i=1}^n c_i \cdot v_i}{C_n}$$

dove  $v_i \geq 18$  è il voto conseguito nell'esame  $i$ -esimo e  $c_i$  è il numero di crediti "dell'esame"  $i$ -esimo. Se uno studente non ha esami in carriera, la sua media pesata è definita uguale a zero. Conoscendo la media pesata attuale  $m_n$  e il numero di crediti complessivo attuale  $C_n = \sum_{i=1}^n c_i$  è quindi possibile ottenere la media aggiornata all' $n + 1$ -esimo voto  $v_{n+1} \geq 18$  per un esame da  $c_{n+1}$  crediti come

$$m_{n+1} = \frac{c_{n+1} \cdot v_{n+1} + m_n \cdot C_n}{C_n + c_{n+1}}.$$

L'Università di Watopia non chiede tasse di iscrizione, ma concede agli studenti un numero massimo di "fallimenti" nella loro carriera: uno studente che ottenga  $d = 5$  insufficienze viene espulso dal CdL. Per un errore della segreteria, alcuni studenti che erano stati espulsi dopo la precedente sessione non sono stati eliminati dal file `studenti.dat`. Comunque a nessuno di questi studenti è stato consentito di sostenere esami nella sessione di gennaio-febbraio, quindi nel file `votiGF.dat` non compaiono risultati riferiti a questi studenti.

In questa prova d'esame ci occuperemo dell'implementazione del codice che consenta di rettificare l'errore commesso dalla segreteria e aggiornare la carriera degli studenti dell'Università di Watopia. I file di dati `votiGF.dat` e `studenti.dat` sopra menzionati sono contenuti nella cartella

`/home/comune/20240209_Dati`

che risiede sulla macchina `tolab.fisica.unimi.it`

Specifiche del progetto, leggete attentamente  $\Rightarrow$

## SPECIFICHE DEL PROGETTO

Definite le strutture

```
struct voto {
    int cod; //codice esame
    int cre; //numero crediti esame
    int matr; //matricola studente
    int val; //voto conseguito
};

struct studente {
    int matr; //matricola studente
    int cre; //numero crediti conseguiti
    float wm; //media pesata
    int nfails; //numero esami falliti
    bool is_out; //true: espulso
};
```

1. Caricare i voti dal file `votiGF.dat` in un array di `voto` allocato dinamicamente. Stampare a video il numero di voti caricati, la frazione di voti sufficienti sul totale e la frazione di voti insufficienti sul totale. NOTA: sono due numeri tra zero e uno. Stampare a video la descrizione (codice, numero crediti e voto) dei primi 3 e degli ultimi 3 voti caricati.
2. Caricare i dati degli studenti dal file `studenti.dat` in un array di `studente` allocato dinamicamente. Assegnare `true` al campo `is_out` se il numero di esami “falliti” supera il limite consentito, ovvero se  $n\_fails \geq d = 5$ , `false` altrimenti. Stampare a video il numero di studenti registrati su file, il numero di studenti NON espulsi e il numero di studenti espulsi. Stampare a video la descrizione di tutti e soli gli studenti espulsi, stampando la stringa “fuori” in corrispondenza del campo `is_out` se lo studente è espulso (`is_out` vale `true`) e la stringa “dentro” altrimenti.
3. Eliminare (scremare) dall’array di studenti creato al punto precedente tutti gli studenti espulsi. Ordinare quindi l’array di `studente` “ripulito” dagli studenti espulsi in ordine di matricola crescente. Stampare a video la descrizione dei primi 3 e gli ultimi 3 elementi dell’array ordinato.
4. Usando l’array dei voti caricato al Punto 1, aggiornare le schede degli studenti come segue. In particolare: per ogni voto:
  - (i) individuare la scheda dello studente che ha conseguito il voto all’interno dell’array di `studente` ottenuto al Punto 3.
  - (ii) se il voto è sufficiente ( $\geq 18$ ), aggiornare il numero di crediti e la media ponderata dello studente usando la relazione (2) alla pagina precedente;
  - (iii) se il voto è insufficiente ( $< 18$ ) incrementare il numero di esami falliti e aggiornare, se necessario, il campo `is_out`.
5. Terminata la fase di aggiornamento, determinare il numero di studenti espulsi a seguito dei risultati della sessione di gennaio-febbraio. Stampare a video la descrizione di tutti e soli gli studenti espulsi.
6. Registrare sul file `studenti_GF.dat` le schede aggiornate di tutti e soli gli studenti NON espulsi al termine della sessione di gennaio-febbraio, in ordine di matricola crescente (come al Punto 3). Le schede devono essere registrate nello stesso formato del file `studenti.dat` usato al Punto 2, ovvero registrando una sola scheda per riga, senza alcuna intestazione o didascalia, così che il file possa essere usato dal vostro programma per aggiornamenti futuri.

Importanti avvertenze alla pagina successiva  $\Rightarrow$

## ISTRUZIONI PER LA CONSEGNA DEL PROGETTO

Il vostro software deve essere predisposto in una cartella denominata `cognome_matricola` che deve essere copiata in `/home/comune/20240209_Risultati` sulla macchina `tolab.fisica.unimi.it`

Nella cartella `cognome_matricola` devono essere inclusi:

- un `makefile` che tramite i comandi `make compila` e `make esegui` consenta rispettivamente di compilare e di eseguire il programma,
- i due file `studenti.dat` e `votiGF.dat` dei dati di input del progetto,
- il file `studenti_GF.dat` prodotto dal programma,
- tutti e soli i file `.C`, `.cpp`, `.h`, `.hpp` utili alla soluzione del problema.

## VALUTAZIONE DEL PROGETTO.

La valutazione terrà conto sia della qualità dei risultati sia della struttura e dell'organizzazione del codice; per chiarire, sono graditi uso di funzioni e compilazione separata, mentre non è gradito un `main` onnicomprensivo. **I progetti che non compilano o che entrano in loop dopo il lancio verranno immediatamente classificati come insufficienti.**