

UNIVERSITÀ DEGLI STUDI DI PARMA  
Dipartimento di Ingegneria e architettura  
Corso di Laurea in Ingegneria dei Sistemi Informativi  
a.a. 2020-2021

**PROVA PRATICA DI INFORMATICA & LAB. PROGRAMMAZIONE**  
9 aprile 2021

Nome: \_\_\_\_\_ Cognome: \_\_\_\_\_ Matr: \_\_\_\_\_ Postazione \_\_\_\_\_

Scrivere un programma in linguaggio C (chiamare il progetto con la propria <**matricola**>) che abbia il comportamento descritto nel seguito. Il tempo a disposizione è di **120 minuti**. Al termine del tempo gli elaborati salvati su **Z:\** verranno raccolti automaticamente dal sistema di laboratorio. Eventuali documenti sono disponibili in **P:\Bertozzi**, si consiglia di usare **wordpad** per leggere i file di testo.

**Per la realizzazione del programma non utilizzare array o matrici allocate staticamente se non indicato diversamente nel testo. L'uso di variabili globali o numeri magici sarà inoltre sanzionato.**

È fornito un file elenco.txt contenente un numero non precisato di righe che riga per riga elenca i voti degli esami sostenuti dagli studenti separati da uno o più spazi: matricola (max 10 car.), nome (max 63 car.), cognome (max 63 car.), numero degli esami sostenuti (max. 24) e relativi voti. Es:

```
339992 NGONGANG FREDRIK 3 26 27 29
391652 SIGNORINI     UGO  4 18 18 19 20
391652 SIGNORINI     UGO  5 18 18 19 20 22
339992 NGONGANG FREDRIK 5 26 27 29 30 30
391652 SIGNORINI     UGO  6 18 18 19 20 22 27
```

I dati vengono ogni volta aggiunti per cui i nomi possono essere ripetuti.

Il candidato sviluppi i seguenti punti:

1. definire la struttura “studente” atta a memorizzare i dati di una riga e implementare le seguenti funzioni:
2. **int leggi\_studente(FILE \*fp, struct studente \*dest)** che legge da file già opportunamente aperto una riga e la memorizza in dest. La funzione restituisce vero o falso se la lettura è andata o meno a buon fine
3. **int compara\_studenti(const struct studente \*a, const struct studente \*b)** che restituisce 0 se le matricole di a e b sono differenti
4. **struct studente \*leggi\_file(const char \*filename, int \*n)** che legge un file di nome “filename” avendo il formato descritto e restituisce un array allocato dinamicamente di struct studente e ponga in dim il numero di studenti letti da file
5. **int studenti\_diversi(const struct studente \*src, const int dim)** che prende in ingresso l’array di studenti restituito al punto precedente e conta quanti studenti differenti sono memorizzati (usando la funzione compara\_studenti()) e restituisce tale numero
6. **struct studente \*pulisci\_studenti(const struct studente \*src, const int dim, int \*ndim)** che prende in ingresso i dati letti al punto (4) e restituisce un nuovo array sempre

allocato dinamicamente contenente solo i dati piú aggiornati per ciascun studente. Tale procedura sfrutta la funzione al punto (5)

7. **void salva\_studente(FILE \*fp, const struct studente \*src)** che salva, su file già aperto, una riga con i dati dello studente all'indirizzo “src”

Il candidato sviluppi opportuna funzione main() che, sfruttando quanto fatto fino al punto 6 legga il file fornito e crei opportuna struttura con i dati degli studenti, stampi il risultato a video e, usando la funzione al punto (7) salvi su opportuno file i dati aggiornati.