

Fondamenti di Programmazione  
A.A. 2013/2014  
Appello 11 Settembre 2014 - Fila A

NOME: \_\_\_\_\_

COGNOME: \_\_\_\_\_

MATRICOLA: \_\_\_\_\_

email: \_\_\_\_\_

Dettagli: il presente documento deve essere consegnato al docente, mentre il codice sorgente associato alla prova d'esame deve essere memorizzato nel file

**nome\_cognome.c.**

Consegna dell'elaborato: il file **nome\_cognome.c** deve essere copiato nella directory FONDAMENTI DI PROGRAMMAZIONE, accessibile tramite il collegamento Esami su SUNTO presente nel Desktop di Windows. E' sufficiente trascinare il file nella cartella di consegna.

ATTENZIONE: L'operazione può essere effettuata una sola volta.

Valutazione: ogni funzione ha un punteggio massimo. La valutazione totale è data dalla somma dei punteggi ottenuti in ciascuna funzione. Si ricorda di commentare brevemente il codice.

## PROVA D'ESAME

Scrivere un programma per valutare un esame composto da **nq** domande (con **nq** compreso tra 5 e 50) a scelta multipla (**a, b, c, d**) e notificare le domande sbagliate più frequentemente. Il programma riceve i dati da un file **examdat.txt**. La prima riga del file contiene il numero **nq** di domande da cui è composto l'esame, seguito da uno spazio bianco e da una sequenza (**corr\_answer**) di **nq** caratteri contenente le risposte esatte. Ognuna delle righe successive alla prima contiene un valore intero corrispondente alla matricola dello studente seguito da uno spazio bianco e infine dalle risposte dello studente. Di seguito un breve esempio del file in ingresso:

```
(examdat.txt)
5 dbbac
111 dabac
102 dcbdc
251 dbbac
```

l'elenco dei numeri di matricola verrà poi memorizzato (vedi Esercizio 2) in un array monodimensionale **id\_students** e le risposte in un array bidimensionale **answers** (le risposte relative ad uno stesso studente su una stessa riga).

Con questi dati il programma calcola le percentuali di errori per ciascuna domanda da parte degli studenti memorizzandoli in un array **wrong**.

Infine il programma produce in uscita un file **report.txt** contenente tutte le informazioni precedenti come mostrato nell'esempio successivo:

```
(report.txt)
Exam Report
Question 1 2 3 4 5
Answer    d b b a c
Wrong(%) 0 66 0 33 0
```

In particolare, scrivere le seguenti funzioni C per la realizzazione del programma:

### Esercizio 1: [punti 8]

**main;**

effettua il collaudo del programma, apre il file in lettura e richiama la funzione dell'Esercizio 2 e richiude il file. Chiama la funzione dell'Esercizio 3 per il calcolo delle percentuali di errori per ciascuna domanda da parte degli studenti. Apre il file in scrittura, richiama la funzione dell'Esercizio 4 e richiude il file.

### Esercizio 2:[punti 8]

acquisisce le risposte degli studenti da un file (**examdat.txt**) aperto e memorizza i dati opportunamente:

**fget\_answer;**

legge prende come parametri: il file da cui leggere i dati, in ingresso; il numero di domande, l'array delle risposte corrette **corr\_answer**, l'array **id\_students** e l'array **answers**. Legge i dati dal file e li memorizza nei rispettivi array.

Esercizio 3:[punti 8]

**get\_wrong;**

calcola la percentuale di errore per ciascuna domanda da parte degli studenti, per tutte le domande, e le memorizza in **wrong** (parametro in uscita). Quali parametri in ingresso devono essere passati alla funzione?

Esercizio 4:[punti 8]

**fwrite\_report;**

scrive su un file (**report.txt**) aperto le risposte corrette per ciascuna domanda, la percentuale di errore per ciascuna domanda da parte degli studenti, nel formato mostrato nell'esempio. Quindi prende come parametri in ingresso (in aggiunta al file) l'array **corr\_answer**, e l'array **wrong**.