

Pre-appello compitino - 9 Gennaio 2017

A.A. 2016/2017

(Tempo a disposizione - 2h)

Dato lo scheletro di codice qui riportato, lo si completi aggiungendo l'implementazione delle funzioni richieste.

bozza.c

```
#include <stdio.h>
#include <stdlib.h>

// funzioni da implementare
float** read_grades(int n_students, int n_courses);
float** student_maximums(float** grades, int n_students, int
    n_courses);
float* course_averages(float** grades, int n_students, int
    n_courses);

// funzione di stampa voti medi con header (matricola, voto)
void print_students(float **M, int rows, int columns) {
    int i, j;
    printf("%15s%15s\n", "Matricola", "Voto");
    for (i=0; i<rows; i++) {
        float *row = M[i];
        for (j=0; j<columns-1; j++)
            printf("%15.2f ", row[j]);
        printf("%15.2f\n", row[columns-1]);
    }
}

// funzione di stampa voti, con il nome del corso nel header
void print_courses(float *grades, char** course_names, int size){
    int i;
    for (i=0; i<size; i++) {
        printf("%15s ", course_names[i]);
    }
    printf("\n");
    for (i=0; i<size; i++) {
        printf("%15.2f ", grades[i]);
    }
    printf("\n");
}
```

```
}
```

```
int main() {
    //numero di studenti
    int n_students;

    //numero di corsi
    const int n_courses=5;

    //nomi dei corsi - gia definiti
    char* course_names[]={ "Mathematics", "Physics", "Programming", "
        Chemistry", "English" };

    //voti degli studenti - una matrice dove
    //ogni riga rappresenta uno studente
    //la prima colonna rapresenta la matricola dello studente
    //e il resto delle colonne rapresentano i corsi
    float** student_grades;

    //voti massimi degli studenti - una matrice con 2 colonne e
    n_students righe
    //la prima colonna contiene la matricola, la seconda il voto
    massimo
    //ogni riga corrisponde ad uno studente
    float** student_max;

    //voti medi per ogni corso
    float* avg_grades;

    // leggo il numero di studenti (n_students>0)
    scanf("%d", &n_students);
    if(n_students<=0) return 0;

    // leggo le matricole e i voti degli studenti
    student_grades=read_grades(n_students, n_courses);

    // calcolo i voti massimi degli studenti
    student_max = student_maximums(student_grades, n_students,
        n_courses);

    // stampo i voti medi
    printf("Voti massimi:\n");
    print_students(student_max, n_students, 2);

    // calcolo il voto medio per ogni corso
    avg_grades = course_averages(student_grades, n_students,
        n_courses);

    // stampo voti medi per ogni corso
    printf("Voti medi per corso:\n");
    print_courses(avg_grades, course_names, n_courses);
}
```

```

    return 0;
}

```

Le funzioni da implementare devono rispettare le seguenti specifiche:

- **read_grades**: legge le matricole e i voti di **n_students** studenti. Per ogni studente, viene prima letto il numero di matricola, seguito da una sequenza di **n_courses** voti reali, uno per ogni corso. Per semplicità, si assuma che anche la matricola sia un numero reale. I valori **n_students** e **n_courses** vengono passati come parametri alla funzione. I dati letti vanno inseriti in una matrice di numeri reali dove ogni riga corrisponde ad uno studente: il primo elemento di ciascuna riga contiene la matricola dello studente; gli elementi successivi contengono i voti nell'ordine di inserimento. La matrice *deve* essere memorizzata come illustrato in Fig. 1. La funzione restituisce questa matrice.

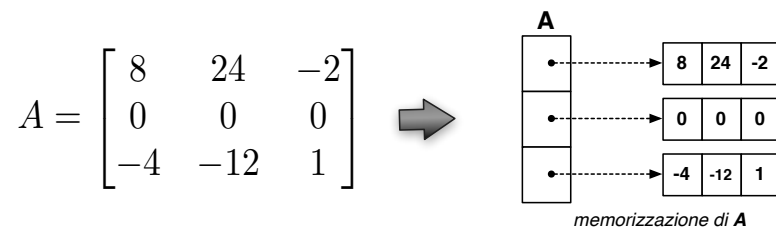


Figura 1: Memory layout di una matrice.

- **student_maximums**: data la matrice di voti, **grades**, la funzione calcola il massimo dei voti per ogni studente. Il risultato viene memorizzato in una matrice **max** con **n_students** righe e 2 colonne: il primo elemento di ciascuna riga contiene il numero di matricola dello studente (sempre in ordine di inserimento), mentre il secondo elemento contiene il voto massimo corrispondente. Per esempio se i voti per 5 corsi sono (N.B. la prima colonna contiene le matricole):

$$\mathbf{grades} = \begin{bmatrix} 1111.0 & 27.5 & 20.0 & 18.2 & 24.0 & 30.0 \\ 2222.0 & 26.5 & 30 & 27 & 28 & 30 \\ 3333.0 & 18.5 & 21 & 24.5 & 27.5 & 23 \end{bmatrix}$$

i voti massimi saranno:

$$\mathbf{max} = \begin{bmatrix} 1111.0 & 30.00 \\ 2222.0 & 30.00 \\ 3333.0 & 27.50 \end{bmatrix}$$

La funzione deve restituire la matrice ottenuta, **max**. Anche per la matrice **max** è obbligatorio assumere il layout indicato in Figura 1.

- **course_averages**: data la matrice di voti, **grades**, la funzione restituisce un array **avg** di dimensione **n_courses** dove ogni elemento rappresenta il voto massimo ottenuto dagli studenti per quel corso. Notate che l'elemento *i*-esimo dell'array **avg** conterrà il voto medio del corso memorizzato nella colonna **i+1** della matrice **grades** (dal momento che la prima colonna di tale matrice contiene i numeri di matricola). La funzione prende come parametro la matrice **grades** , il numero di studenti **n_students** ed il numero di corsi **n_courses** e restituisce un puntatore a **avg**.

Esempio

Input	Output
3	Voti massimi:
1111	Matricola Voto
27.5	1111.00 30.00
20	2222.00 30.00
18.2	3333.00 27.50
24	Voti medi per corso:
30	Mathematics Physics Programming Chemistry English
2222	24.17 23.67 23.23 26.50 27.67
26.5	
30	
27	
28	
30	
3333	
18.5	
21	
24.5	
27.5	
23	