

COP 3502- Lab 2 on Dynamic Memory Allocation

In this lab you will be practicing with a dynamic memory allocation coding problem. Then you will be learning how to use the memory leak detector code and check whether your code has any memory leak or not. The steps of using leak detector code in Eustis and repl.it is available in webcourses. For most of your programming assignments in this semester, you will need to use the provided leak detector code. Note that you do not have to submit this code. But, you must try to work on it in the lab with the help of the TA. Doing this will improve your understanding on DMA. Working on this will also better prepare you for the Dynamic Memory Allocation Assignment.

The Coding Problem:

In this problem, you will read a set of students data and their grading information from a file and then process them and then write the requested data to another file.

Problem:

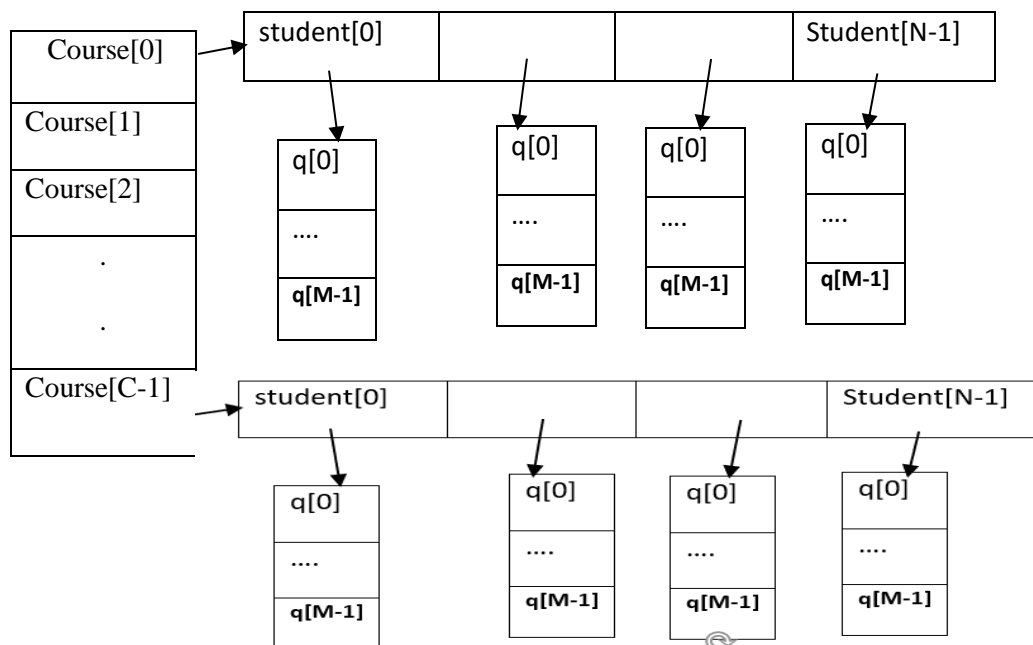
ABC training center offers **C number** of courses. A course has **N number of students**. A student has a last name (single word string with maximum length 50 characters) and a set of course activities, including **one assignment**, **M number of quizzes** and a **final exam**. The total score of a student is calculated as follows:

The total score = sum of scores from M quizzes + score in the assignment + score in the final exam

The student structure should follow the following definition.

```
typedef struct student
{
    char *lname;
    int assignment;
    int finalExam;
    int total;
    int *quizzes;
}student;
```

You can visualize it by the following picture.



- All the inputs are available in a text file in.txt. Take number of Courses **C**, number of students **N** and number of quizzes **M** as the input from the first line of the input file. Then Dynamically allocate memory for C courses with N students for each course. For each student, dynamically allocate memory to store scores of **M** quizzes. Take input for all the scores for quizzes, assignments, and final exams for all the students. Calculate the total scores for each student and store it in the corresponding structure. Do this whole task in a separate method and return the appropriate pointer.

The function header should look like this:

```
student** readCourses(FILE *in, int *C, int *N, int *M);
```

Display the student's details who achieved the highest total score across all the courses. You do not have to consider if a student with same name is available in multiple courses.

- Also write the same details into a file **out.txt**
- Also note that, in the in.txt file, there will be a blank line after each course. However, fscanf will automatically take care of it.
- After writing the result, you should call the release_memory function to free up all the memory.

The function prototype is:

```
void release_memroy(student ** st, int C, int N, int M)
```

- You should also use the memory leak detector code as instructed in the webcourses.

Sample Input: All the inputs, including N and M will come from an input file called input.txt The input file is structured as follows:

Sample in.txt

3 4 2 ///**C N M**

adel 10 12 9 45 // last name, assignment score, scores for M number of quizzes, and final exam score

smile 6 8 9 39

mahmud 10 12 10 15

jose 8 11 7 41

adam 10 12 9 45 //second course

smith 6 8 9 39

muhammad 10 12 10 45

jones 8 11 7 41

adil 10 12 9 45 //third course

samuel 6 8 9 39

miguel 5 12 10 40

jerry 8 11 7 41

Sample out.txt:

Name: muhammad

Assignment: 10

Quizzes: 12 10

Final exam: 45

Total: 77

Course Number: 1