Q1)

Define a structure named `Student` with the following fields:

- `rollNo` (int)
- `name` (string)
- `marks` (float)

Create an array of 5 students. Prompt the user to enter details for each student and then display all the entered student records.

Expected Output:

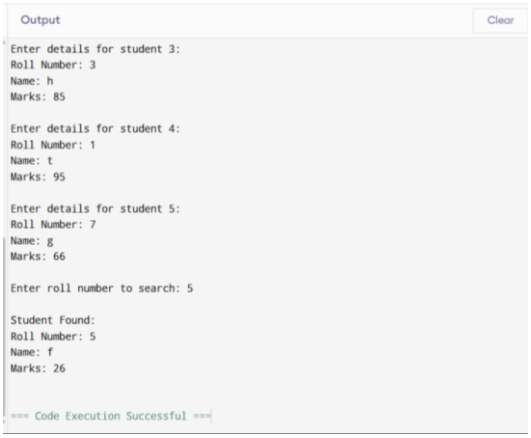
Q2)

Modify the previous program so that it uses a `pointer to a structure` to display student details instead of directly accessing the array.

Q3)

Extend the student record program to allow the user to enter a roll number and search whether a student with that roll number exists. If found, display the student's details; otherwise, show a "*not found*" message.

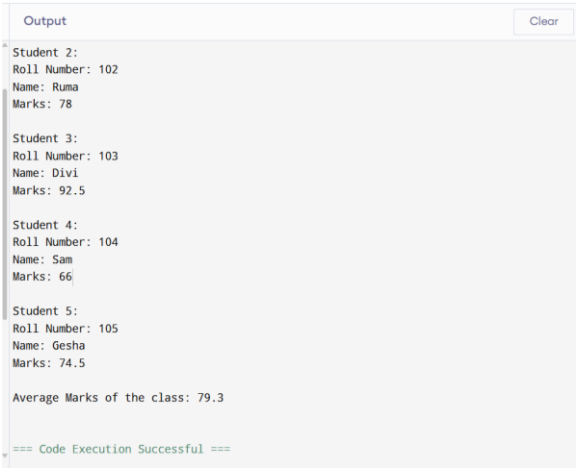
Expected Output:



Q4)

Create a program that defines a structure `Student` with fields `rollNo`, `name`, and `marks`. Read details for 5 students and then calculate and display the average marks of the class.

Expected Output:



Q5)

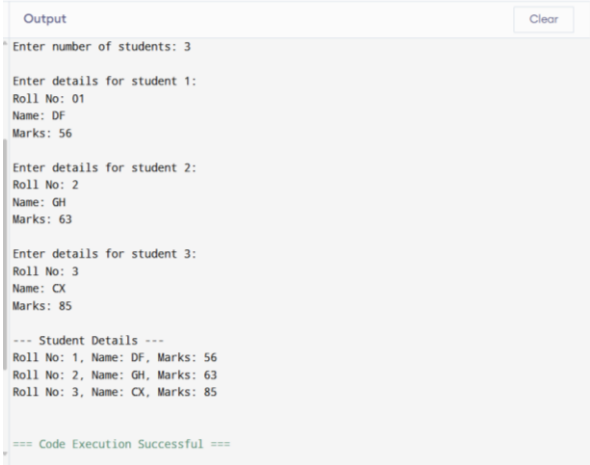
Create a structure `Student` with fields `rollNo`, `name`, and `marks`. Read data for 5 students and sort them in descending order of their marks.

Expected Output:

Q6)

Write a program using `new` and `delete` to dynamically allocate memory for an array of students based on user input (i.e., the user enters how many students they want to input).

Expected Output:



Q7)

Create a structure for `Student` with fields `rollNo`, `name`, and `marks`.

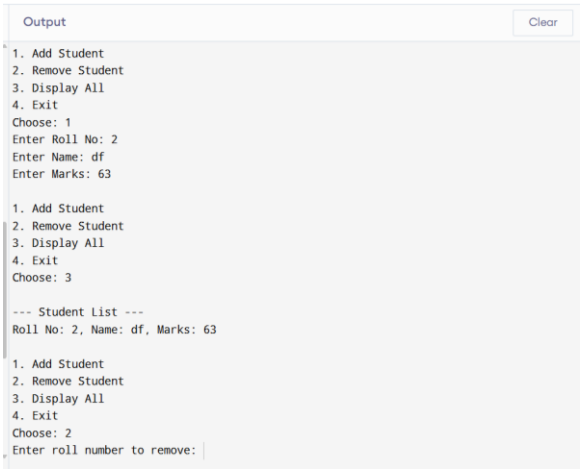
- Accept details for 3 students
- Write them to a file
- Read the file and display the student records.

Q8)

Using dynamic memory, allow the user to add and remove student records.

- Allow addition of new students
- Remove a student by roll number
- Show the updated list.

Expected Output:



Q9)

Write a program where student details are passed to a function by reference. The function should update the student's marks by adding 5 bonus marks.

Q10)

Demonstrate the difference between:

- Array of Structures
- Structure of Arrays
  By storing marks of students in both ways and printing them.

Expected Output: