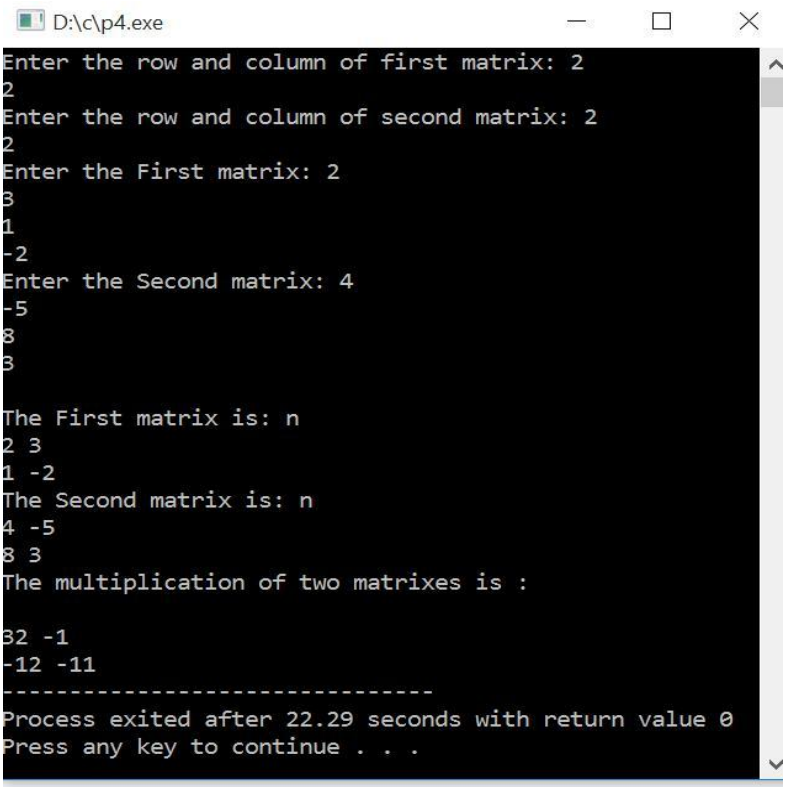
Practice Problems:

Question 1:

Write a program in C to multiply two matrix using recursion.



Question 2:

Write a recursive implementation of the factorial function. Recall that n! = 1 × 2 × … × n, with the special case that 0! = 1.

Question 3:

Write a recursive function that, given two strings, returns whether the first string is a subsequence of the second. For example, given har and cathartic, you should return true, but given bat and table, you should return false.

Question 4:

Write a recursive function that checks whether a string is a palindrome (a palindrome is a string that's the same when reads forwards and backwards.)

Question 5:

Consider an array given below:

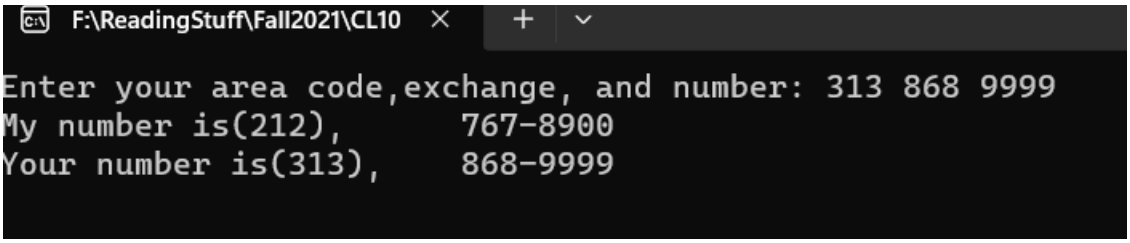
A [10] = {5, 4, 3, 2, 1, 6, 10, 9, 7, 8}

Your task is to write a recursive function that searches for a given element from that array.

Question 6:

A phone number, such as (212) 767-8900, can be thought of as having three parts: e.g., the area code (212), the exchange (767), and the number (8900). Write a program that uses a structure to store these three parts of a phone number separately. Call the structure phone.

Sample output:



Create two structure variables of type phone. Initialize one, and have the user input a number for the other one. Then display both numbers.

The interchange might look like this:

Enter area code: 415

Enter exchange: 555

Enter number: 1212

Then display like below:

My number is (212) 767-8900

Your number is (415) 555-1212

Question 07:

Define a structure to store the following student data:

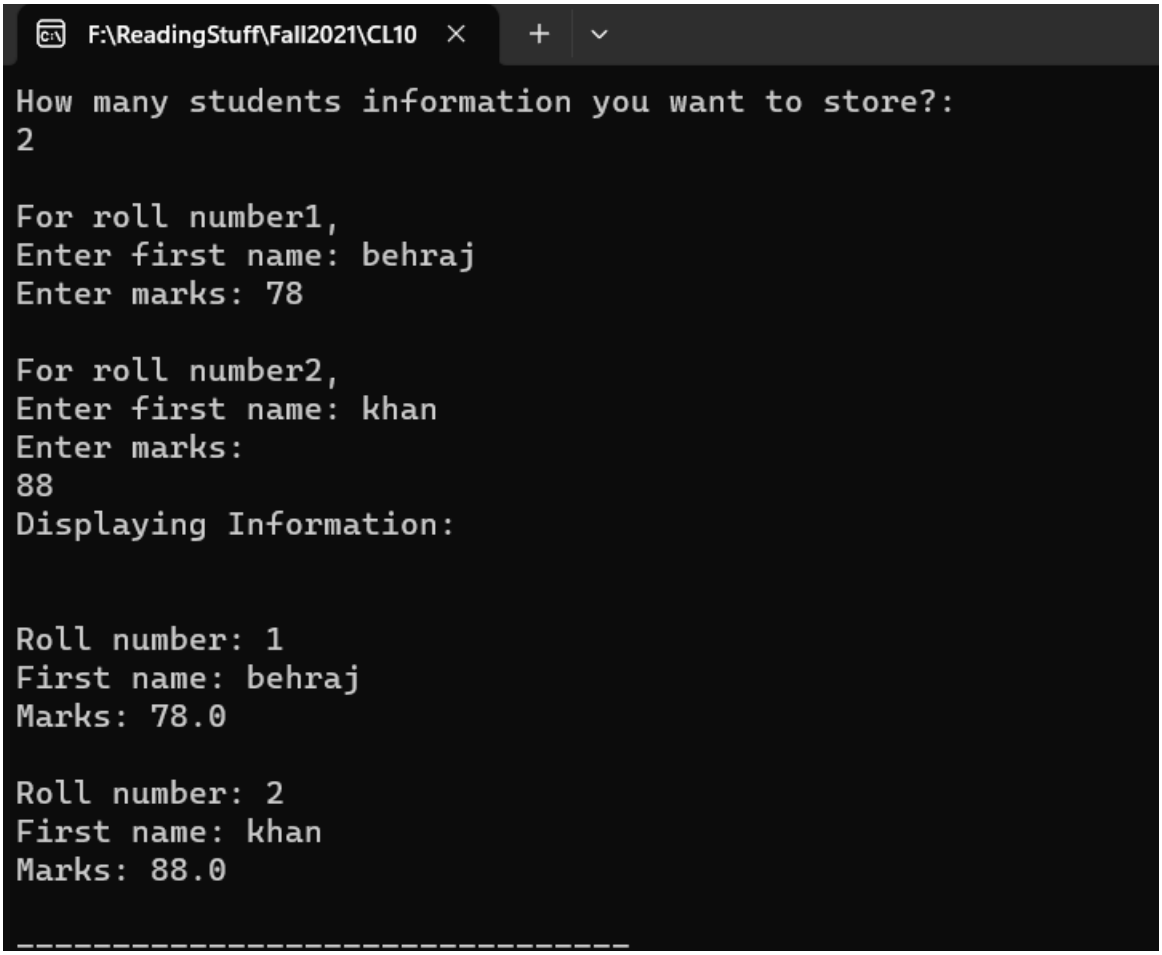
CGPA, courses (course name, GPA), address (consisting of street address, city, state, zip). Input 2 student records, compare and display which student have highest GPA in which course also Display which student has the highest CGPA .

HINT: define another structure to hold the courses and address.

Question 08:

Write a C Program to Store Information of N Students Using Structure, where N is provided by the user.

Sample output



Question 09:

Write a C program that uses functions to perform the following operations:

i) Reading a complex number

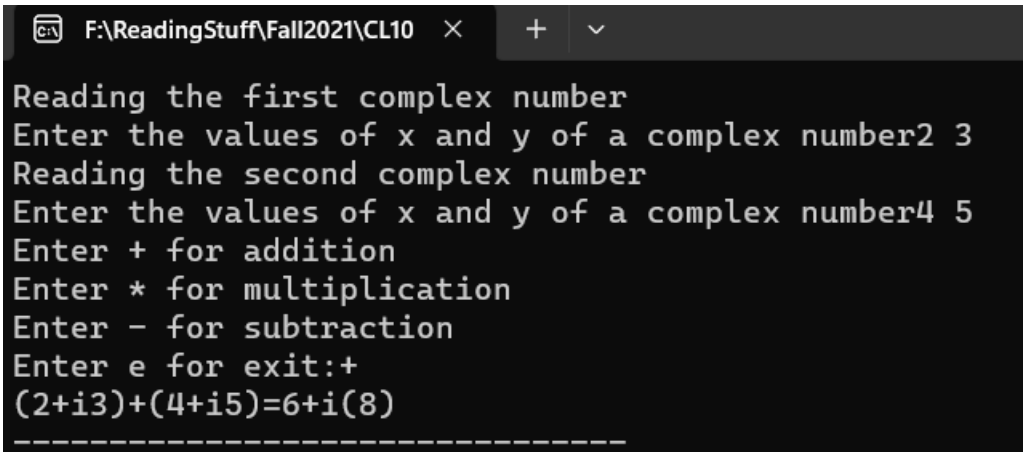
ii) Writing a complex number

iii) Addition of two complex numbers

iv) Multiplication of two complex numbers

(Note: represent complex numbers using a structure.)

Sample output


F:\ReadingStuff\Fall2021\CL10
Reading the first complex number
Enter the values of x and y of a complex number2 3
Reading the second complex number
Enter the values of x and y of a complex number4 5
Enter + for addition
Enter * for multiplication
Enter - for subtraction
Enter e for exit:+
(2+i3)+(4+i5)=6+i(8)