CS19003 Programming and Data Structures Laboratory
Assignment 4

▷ Name the files as ⟨assignment no⟩_⟨question no⟩_⟨roll no⟩.c, without the ⟨ and ⟩. Consult your mentor for any confusion. Penalty if the file names do not stick to this convention.

▷ This is an assignment on one-dimensional arrays, functions and recursion. Do not use two-dimensional/multi-dimensional arrays, pointers, structures, dynamic memory allocation or any other advanced concept.

---

1. Write a program to take two matrices as inputs, multiply them and print the product matrix on the screen. Note that you are not allowed to use two-dimensional/multi-dimensional arrays. Only one-dimensional arrays can be used. In this program, you may need to define arrays with variable length. Traditionally it is done using dynamic memory allocation that you will learn about later. As for this lab, you may use declaration of the form *int A[n];* Make sure that the variable n contains some valid data at the time of this declaration. In place of the variable n, you may also use an expression (eg. 2*n+3*m).

   Sample input/output:

   Input: an array of dimension m x n, and an array of dimension n x p

   Enter m, n, p: 2 3 4

   Input first matrix

   Enter element (1, 1): 2

   Enter element (1, 2): -1

   Enter element (1, 3): 3

   Enter element (2, 1): 0

   Enter element (2, 2): 4

   Enter element (2, 3): -5

   Input second matrix

   Enter element (1, 1): -5

   Enter element (1, 2): 8

   Enter element (1, 3): -3

   Enter element (1, 4): 5

   Enter element (2, 1): 6

Enter element (2, 2): -8

Enter element (2, 3): -9

Enter element (2, 4): 5

Enter element (3, 1): 2

Enter element (3, 2): -1

Enter element (3, 3): -5

Enter element (3, 4): -6

Product matrix:

-10 21 -12 -13

14 -27 -11 50

[100 points]

2. Write a program that

   ▷ takes a string as input from the user and stores it in a character array named *line*. The string may have whitespaces. Hence, please use scanf("%[^\n]s", line). Using only "%s" will not read the spaces. You may assume that the string contains only letters and blanskpace characters. Assume that the input string has at most 99 characters (except the terminal null character).

   ▷ removes all the blankspaces, so that *line* becomes free from all blank spaces.

   ▷ prints *line* using *printf("%s",line);*

   Your program should not use any array other than *line*. Do not use the string library.

   Caution: 1. The program must modify the array *line*. Just printing the string without the spaces is not enough,
   2. The string may end with blanks. Such terminal blanks should also be removed.

   Sample input/output:

   Enter string: I study in I I T Kharag p ur

   IstudyinIITKharagpur

   [100 points]

3. Suppose a frog can jump either 1 foot, or 2 feet or 3 feet. Write a function with appropriate prototype that takes in a positive integer n, and prints all possible ways in which a frog can cover n feet. The program should also print the total number of such ways. You may declare arrays of variable length as suggested in question 1. Feel free to write more than one user-defined function and use recursion.

   Sample input/output:

   Enter n: 4

   1 1 1 1

   1 1 2

   1 2 1

   1 3

2 1 1

2 2

3 1

Number of ways: 7

[100 points]