ESS111 : Programming 1 (C Programming)
LAB - 8

Due: 10 February, 2021 @ 11:59 pm

Part A (to be submitted)

Problem 1: A data structure is a way of organizing data. One data structure that you have explored is array, which organizes data linearly and offers indexed access to it. Another one is stack. Unlike array where you can add an element at any index, stacks only support insertion and deletion at one end, which is called the 'top' of the stack (like a water bottle - you can only fill or pour water from one end of the bottle).

You need to implement a stack using an integer array of size 5. The following functions must be implemented-

- Push() - This function inserts an element at the top of the stack. If the stack has already reached maximum capacity, the function should print "STACK OVERFLOW" and return -1. Otherwise return 0.
- Pop() - This function removes the top element. If the stack is empty, the function should print "STACK UNDERFLOW" and return -1. Otherwise return the element which was removed.
- Display() - This function should print all the elements of the stack by successively popping from the stack. If the stack is empty, then print "STACK EMPTY". This function need not return anything.

You are free to define any auxiliary functions if required.

Input Format :
The character a denotes PUSH operation, b denotes POP operation, and c denotes DISPLAY. So you are given a sequence of instructions, which could be a x, b, c (where x is a number given through input that will be pushed onto the stack).The input sequence will always end with c instruction, and c will be given only once (i.e. at the end). If an error message is received after performing an instruction, display relevant error message and do not take further input.

Output Format :
If all operations are valid, the elements of the stack, displayed from top to bottom. If not, then relevant error message.

Sample Input 1 :
a 1
a 2
c

Output 1 :
2 1

Sample Input 2:
a 1
b
b
c

Output 2: STACK UNDERFLOW

Hint - Define a variable to track where the top of the stack currently is, and define functions if needed.

Problem 2: You are given 2 arrays A and B. Create a new array C composed of the elements of A and B sorted in ascending order.

Requirements -

- You have to make a separate function named combine(), which takes the 2 arrays and size of each as parameter. You are free to perform any operation you wish on A and B. However, you are not allowed to perform sorting operations on C at any stage.
- You can define any auxiliary functions required.
- In your main() function all you are allowed to do is take input and call your combine() function. So your combine function needs to display the elements of C.

Input format -:
Size of first array
Elements of first array
Size of second array
Elements of second array

Output format :

Elements of final array C displayed in ascending order

Sample Input 1:
3
2 1 3
2
2 1

Output 1 :
1 1 2 2 3

Sample Input 2:
5
1 3 2 5 4
6
11 9 10 7 8 6

Output 2:
1 2 3 4 5 6 7 8 9 10 11

Problem 3: Given an array of integers $a[n]$, write a function to shift it circularly left by x positions. Example - {1,2,3,4,5} when shifted circularly left by 1 position would be {2,3,4,5,1}, and by 2 positions would be {3,4,5,1,2}.

Consider a $m \times n$ matrix M where each element is given by the formula $M_{ij} = \{i + j\}$. (We consider 0-based indexing here). Given a list of shifts, call the function you defined earlier to shift all the rows of M. So for a matrix with 3 rows, if the list of shifts is 1 2 3, you would shift the first row circularly left by 1, the second row by 2, and third row by 3. Finally, display the matrix.

Input Format :

Number of rows and number of columns
(Shift for 1^{st} row) (Shift for 2^{nd} row) ... (Shift for m^{th} row)

Output Format :
Resultant matrix, displayed in m lines with n elements in each line.

Sample Input 1:
3 4
1 2 3

Output 1 :
1 2 3 0
3 4 1 2
5 2 3 4.

Sample Input 2 :
2 3
1 2

Output 2:
1 2 0
3 1 2

Problem 4: Write a program to display the product of two rectangular matrices M_1 and M_2 of dimensions $m_1 \times n_1$ and $m_2 \times n_2$ respectively.You must use multidimensional arrays to represent these matrices. If multiplication is not possible, then print "NOT POSSIBLE".

Requirement - You need to make a function that takes the matrices as input and displays the product of the two matrices. You are free to pass more parameters to the function if required.
Input Format :
m_1 n_1 m_2 n_2
$m_1 \times n_1$ elements of the first matrix, displayed in m_1 lines
$m_2 \times n_2$ elements of the second matrix displayed in m_2 lines

Output Format : The resultant matrix, with m_1 elements on each line.

Sample Input 1 :
1 2 1 2
1 0
1 0

Output 1 :
NOT POSSIBLE

Sample Input 2 :
2 3 3 2
1 1 4
1 2 3
1 3
4 5
6 7

Output 2 :
29 36
27 34

Part B (need not be submitted)

Consider the following statements and indicate which of them is/are correct :

1. A dequeue is an ordered set of elements in which elements may be inserted or retrieved from either end. Use an array to simulate a dequeue of characters and the operations retrieve left, retrieve right, insert left, insert right. Exceptional conditions such as dequeue full or empty should be indicated. Two pointers (namely, left and right) are needed in this simulation.
2. Assuming that January 1, 0 CE was a Monday, write a program that takes the calendar date as input and returns which day of the week it falls on.
3. Write a function which computes the inverse of a given nxn matrix, or if singular return that it has no inverse. Hint - use recursion for computing determinant and adjoint of the matrix.
4. Using the function defined, write a function that takes in $n \times n$ matrix A and nx1 matrix B as input and returns a solution to the equation $Ax = B$ if it exists.