



COMSATS University, Islamabad

Islamabad Campus

Department of Computer Science

Read before Attempt

Assignment No. 2: STACK AND QUEUE	
Course code and Title: CSC211, Data Structure	
Instructor: Dr. Inayat Ur-Rehman	
Assigned Date: March 10, 2025	Due Date: March 23, 2025
Total Marks: --	
CLO-2: Apply linear data structure to various practical problems.	
Guide to Writing up Homework	
<p>You should be as <i>clear and precise</i> as possible in your write-up of solutions. <i>Understandability</i> of your answer is as desirable as <i>correctness</i>, because communication of technical material is an important skill. <i>Sloppy answers</i> will receive fewer points, even if they are correct, so make sure that your handwriting and your <i>thoughts are legible</i>. If writing your problem set by hand, it is a good idea to copy over your solutions to hand in, which will make your work neater and give you a chance to do sanity checks and correct bugs. If typesetting, reviewing the problem set while typing it in often has this effect. In either case, going over your solution at least once before submitting it is strongly recommended. You will often be called upon to "give an algorithm" to solve a certain problem. Your write-up should take the form of a short essay. A topic paragraph should summarize the problem you are solving and what your results are. The body of your essay should provide the following:</p> <ol style="list-style-type: none">A description of the algorithm in pseudo code.At least one worked example or diagram to show more precisely how your algorithm works. <p>Remember, your goal is to communicate. Graders will be instructed to take off points for convoluted and obtuse descriptions.</p>	
Instructions:	
<ol style="list-style-type: none">This is an individual assignment. You will submit your work individually through your logins (course portal)Try to get the concepts, consolidate your concepts and ideas from these questionsYou should concern recommended books for clarify your concepts as handouts are not sufficient.<u>Try to make solution by yourself and protect your work from other students. If I found the solution files of some students are same, then I will reward zero marks to all those students.</u>Deadline for this assignment is October 22, 2024. This deadline will not be extended.	

Question # 1

Indicate whether a stack would be a suitable data structure for each of the following application. Justify your answer.

Application	Stack Suitable or not[Y/N]	Reason
A bank simulation of its teller operation to see how waiting times would be affected by adding another teller		
An address book		
A program to receive data that are to be saved and processes in the reverse order		
A word processor to have a PF key that causes the preceding command to be redisplayed. Every time the user press the PF key, the program shows the command that preceded the one currently displayed.		
A program to evaluate arithmetic expressions according to the specific order of operators		
A dictionary of words used by a spelling checker to be built and maintained		
A data structure used to keep track of return addresses for nested function while a program is running		
A program to keep track of patients as they check into a medical clinic, assigning patient to doctor on a first come, first served basis		

Question # 2

Two stacks of positive integers are needed, one containing elements with values less than or equal to 1,000 and other containing elements with values larger than 1,000. The total number of elements in the small – value stack and the large – value stack combined are not more than 200 at any time, but we cannot predict how many are in each stack. (All of the elements could be in the small –value stack, they could be evenly divided, both stacks could be empty, and so on). Can you think of a way to implement both stacks in one array?

- Draw a diagram of how the stack might look.
- Write the definitions for such a double – stack structure.

- c) Write the algorithm for Push operation; it should store the new item into the correct stack according to its value.

Question # 3

In each plastic container of Pez candy, the colors are stored in random order (See figure below).



Your little brother likes only yellow ones, so he painstakingly takes out all the candies, one by one, eats the yellow ones, and keeps the other in order, so that he can return them to container in exactly the same order as before- minus the yellow candies, of course. Write the algorithm to simulate this process. You may use any of the stack operations defined in the stack ADT, but may not assume any knowledge of stack's implementation.

ANSWER IN THIS BOX

Question # 4 (Applications of Stack)

Write an equivalent postfix expression for the infix expression. Write each step of this conversion.

1. $(A + B) * (C - D)$
2. $A \wedge B * C - D + E / F$
3. $A / (B + C * D - E)$
4. $A - B * C + D / E$
5. $(A + B) \wedge 2 - (C - D) / 2$

ANSWER IN THIS BOX

Question # 5

What is the value of the postfix expression? Write each step of this conversion.

1. $A B + C D - *$
2. $A B ^ C * D - E F / +$
3. $A B C D * + E - /$
4. $A B C * - D E / +$
5. $A B + 2 ^ C D - 2 / -$

Where $A = 12$, $B = 3$, $C = 7$, $D = 4$, $E = 2$ and $F = 5$

ANSWER IN THIS BOX

Question # 6

- i. Write an algorithm for converting an infix expression into prefix expression
- ii. Write an algorithm for evaluation of prefix expression

ANSWER IN THIS BOX

Question # 7

- i. Convert the following infix expression into equivalent prefix expression
$$A^B * C - D + E / F / (G + H)$$
- ii. Evaluate the following prefix expression:
$$+ A * B + C D$$
 if $A = 2, B = 3, C = 4, D = 5$

ANSWER IN THIS BOX

Question # 8 (Recursion)

Consider the following recursion for integer multiplication of two positive number a and b:

$$a * 1 = a$$

$$a * b = a(b - 1) + a$$

This can be implemented using following recursive algorithm as follows:

Algorithm *recursive_multiplication*(a, b)

if b = 1 **then**

return a

else

return a + *recursive_multiplication* (a, b-1)

a) Convert above **recursive algorithm** in to an **iterative algorithm**? Present your iterative version into Pseudo code.

b) Mathematically the following definition for integer multiplication is valid:

$$\mathbf{a \text{ multiply by } b : a * b = a(b+1) - a}$$

Can we have **recursive algorithm** for calculating integer multiplication based on above definition? Explain carefully.

Question # 9

Indicate whether each of the following application would be a suitable for a queue. Justify your answer.

- a. An ailing company wants to evaluate employee records so as to lay off some workers on the basis of service time (the most recently hired employees are laid off first).
- b. A program is to keep track of patients as they check into a clinic, assigning them to doctors on a first come, first served basis.
- c. A program to solve a maze is to backtrack to an earlier position (the last place where a choice was made) when a dead-end position is reached.
- d. An inventory of parts is to be processed by part number.
- e. An operating system is to process requests for computer resources by allocating the resources in the order in which they are requested.
- f. A grocery chain wants to run a simulation to see how the average customer wait time would be affected by changing the number of checkout lines in its stores.
- g. A dictionary of words used by a spelling checker is to be initialized.
- h. Customers are to take numbers at a bakery and be served in order when their numbers come up.
- i. Gamblers are to take numbers in the lottery and win if their numbers are picked.

ANSWER IN THIS BOX

Question # 10

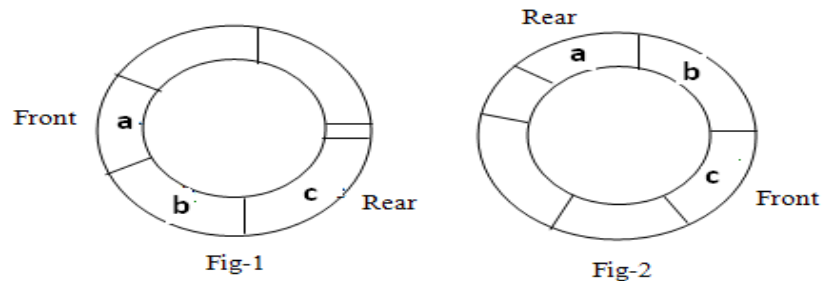
- i. How would you implement a queue of stacks?
- ii. How would you implement a stack of queues?
- iii. How would you implement a queue of queues?
 - a. Draw a diagram of how each of these data structures might look.
 - b. Write algorithms/ routines to implement the appropriate operations of each of these data structures

Question # 11

Write algorithms (for insertion and deletion) that implement two queues in one array where first queue will start from 0th position and second queue will start from last position of the array.

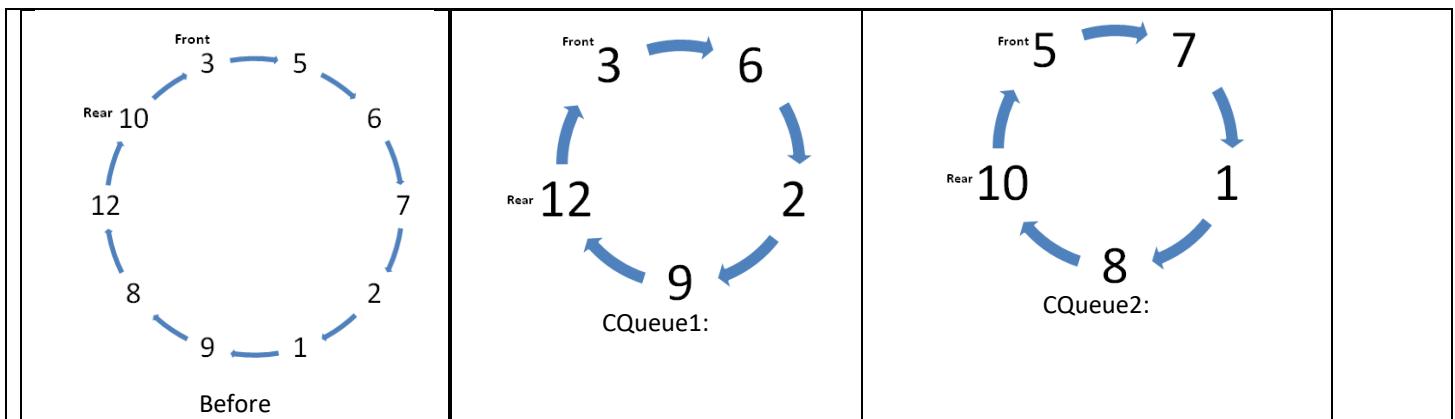
Question # 12

Write an algorithm that uses stack in order to reverse the elements of a circular queue, which is stored in an array. For example, if the initial queue is that given in Fig-1 as under, then the resulting Queue is that given in Fig-2.



Question # 13

It is required to split a circular queue into two circular queues (say CQueue1 and CQueue 2) so that all the elements in odd positions are in one queue and those in even positions are in another queue as shown in the following figure. Write a C++ program to accomplish this. Assume that queue is maintained in an array.



Question # 14

Can a queue be represented by a circular linked list with only one pointer pointing to the tail of the queue. Write C++ functions for add and delete operations on such a queue.

Question # 15

A **DEQUE** is a data structure consisting of a list of items, on which the following operations are possible:

PUSH (X,D) : Insert item X on the front end of **DEQUE** D.

POP(D) : Remove the front item from **DEQUE** D and return it.

Inject(X, D) : Insert item X on the rear end of **DEQUE** D.

Eject(D) : Remove the rear item from **DEQUE** D and return it.

Write C++ program (complete program) that support the above **DEQUE** operations.