

Department of Information Technology

A.Y. 2023-2024

Class: SE-ITA/B, Semester: III

Subject: DATA STRUCTURE LAB

Experiment – 1: Stack and Queue menu driven program

1. Aim: Write a C program to create Array implementation of Stack and Queue.

2. Objectives: After study of this experiment, the student will be able to

- Understand how to create a stack and queue
- Implement an algorithm using computer to solve the given problem

3. Outcomes:

- Develop algorithms for various problems on basic concepts and principles of stacks and queues

4. Prerequisite: Stack, queue and its operations

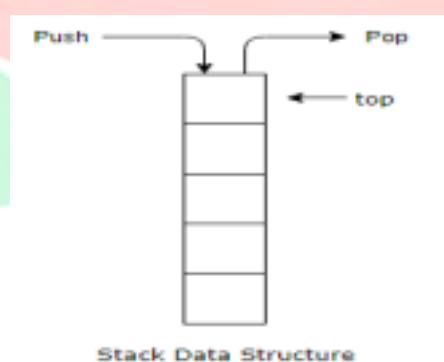
5. Requirements: PC and Turbo C compiler version 3.0 , Codeblock with gcc

6. Pre-Experiment Exercise:

Brief Theory:

1) What is Stack?

Stack is a linear data structure which follows a particular order in which the operations are performed. The order may be LIFO (Last In First Out) or FILO (First In Last Out).



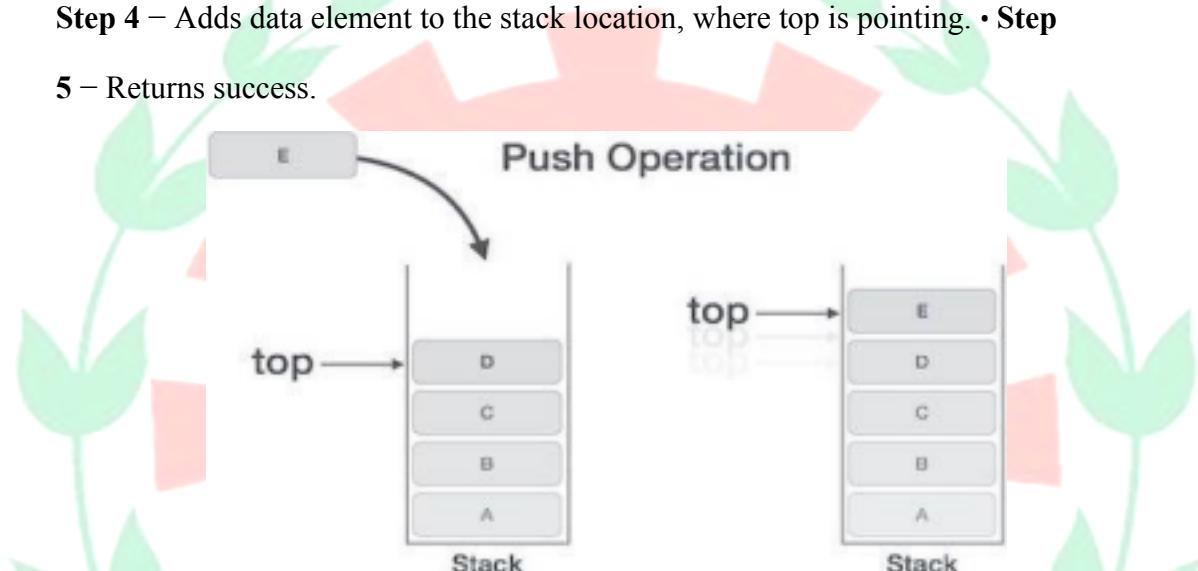
There are many real-life examples of a stack. Consider an example of plates stacked over one another in the canteen. The plate which is at the top is the first one to be removed, i.e. the plate which has been placed at the bottommost position remains in the stack for the longest period of time. So, it can be simply seen to follow LIFO (Last In First Out)/FILO(First In Last Out) order.

2) Operations on Stack.

Push Operation

The process of putting a new data element onto stack is known as a Push Operation. Push operation involves a series of steps –

- **Step 1** – Checks if the stack is full.
- **Step 2** – If the stack is full, produces an error and exit.
- **Step 3** – If the stack is not full, increments **top** to point next empty space.
- **Step 4** – Adds data element to the stack location, where top is pointing.
- **Step 5** – Returns success.



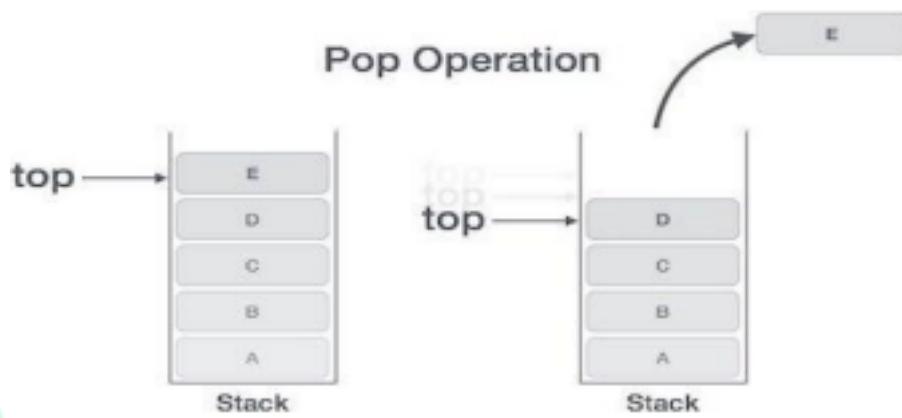
If the linked list is used to implement the stack, then in step 3, we need to allocate space dynamically.

Pop Operation

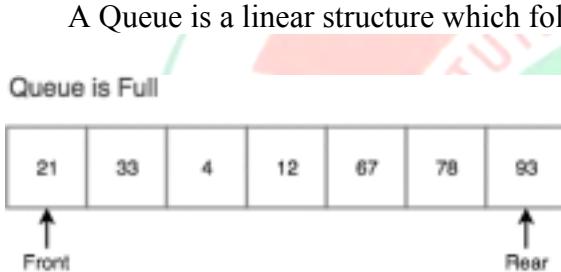
Accessing the content while removing it from the stack, is known as a Pop Operation. In an array implementation of pop() operation, the data element is not actually removed, instead **top** is decremented to a lower position in the stack to point to the next value. But in linked-list implementation, pop() actually removes data element and deallocates memory space.

A Pop operation may involve the following steps –

- **Step 1** – Checks if the stack is empty.
- **Step 2** – If the stack is empty, produces an error and exit.
- **Step 3** – If the stack is not empty, accesses the data element at which **top** is pointing.
- **Step 4** – Decreases the value of top by 1.
- **Step 5** – Returns success.



3) What is Queue?



A Queue is a linear structure which follows a particular order in which the operations are performed. The order is First In First Out (FIFO). A good example of a queue is any queue of consumers for a resource where the consumer that came first is served first. The difference between stacks and queues is in removing. In a stack we remove the item the most recently added; in a queue, we remove the item the least recently added.

4) Operations on Queue

Queue operations may involve initializing or defining the queue, utilizing it, and then completely erasing it from the memory. Here we shall try to understand the basic operations associated with queues –

- **enqueue()** – add (store) an item to the queue.
- **dequeue()** – remove (access) an item from the queue.

Few more functions are required to make the above-mentioned queue operation efficient. These are –

- **peek()** – Gets the element at the front of the queue without removing it.
- **isfull()** – Checks if the queue is full.
- **isempty()** – Checks if the queue is empty.

In queue, we always dequeue (or access) data, pointed by **front** pointer and while enqueueing (or storing) data in the queue we take help of **rear** pointer.

7. Laboratory Exercise

A. Procedure: Stack

Write a C program to implement the stack using switch case that includes operations like

- i. Push
- ii. Pop
- iii. Peek
- iv. Display

B. Result/Observation/Program code: The snapshot of the output obtained is to be printed by students.

C. Procedure: Queue

Write a C program to implement the queue using switch case that includes operations like

- i. Enqueue
- ii. Dequeue
- iii. Display

D. Result/Observation/Program code: The snapshot of the output obtained is to be printed by students.

8. Post-Experiments Exercise

Questions:

- i. Applications of Stack.
- ii. Applications of Queue.
- iii. MCQ

9. Conclusion:

- i. Summary of Experiment
- ii. Importance of Experiment
- iii. Application of Experiment

10. References:

1. Ellis Horowitz, Sartaj Sahni; Fundamentals of Data Structures; Galgotia Publications; 2010.
2. S. K Srivastava, Deepali Srivastava; Data Structures through C in Depth; BPB Publications; 2011.
3. Reema Thareja; Data Structures using C; Oxford.