

SE(ECE)	Data Structures and Algorithms	
Experiment No: 6	Implement Stack using a) arrays and b) linked list.	Page: 1/6

Aim

To implement stack operations (Push, Pop, Peek, Display) using Array and Linked List.

Objectives

1. To understand the concept of stack data structure (LIFO - Last In First Out).
2. To implement stack operations using array-based implementation.
3. To implement stack operations using linked list-based implementation.
4. To compare array and linked list approaches for implementing stack.

Theory

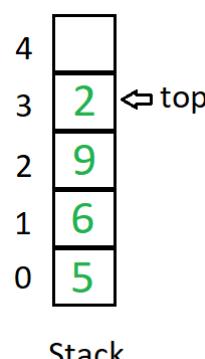
Stack Overview

A stack is a linear data structure that follows the LIFO (Last In First Out) principle.

- The element inserted last is the first one to be removed.
- Main operations:
 - **Push:** Insert an element into the stack.
 - **Pop:** Remove the top element of the stack.
 - **Peek (Top):** Retrieve the element at the top without removing it.
 - **Display:** Show all elements in the stack.

Stack using Array

- Implemented with a fixed-size array.
- The top variable tracks the index of the stack's current top element.
- Advantages: Simple implementation.
- Limitations: Fixed size, may cause overflow if stack exceeds capacity.



SE(ECE)	Data Structures and Algorithms	
Experiment No: 6	Implement Stack using a) arrays and b) linked list.	Page: 2/6

Algorithm (Array Implementation):

Push(x):

1. Increment top.
2. Insert element at arr[top].

Pop():

1. Retrieve element from arr[top].
2. Decrement top.

Peek():

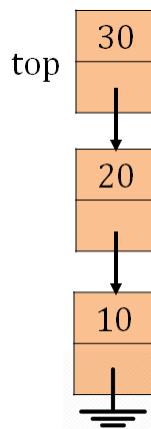
Return element at arr[top].

Display():

Traverse from top to 0 and print elements.

Stack using Linked List

- Implemented dynamically using **nodes**.
- Each node contains:
 - **data** (value)
 - **next** (pointer to the next node)
- Advantages: No fixed size, grows dynamically.
- Limitations: Requires extra memory for pointers.



SE(ECE)	Data Structures and Algorithms	
Experiment No: 6	Implement Stack using a) arrays and b) linked list.	Page: 3/6

Algorithm (Linked List Implementation):

Push(x):

1. Create a new node with value x.
2. Link new node's next to current top.
3. Update top to point to new node.

Pop():

1. If top == NULL, stack underflow.
2. Store top->data.
3. Move top to top->next.
4. Delete old top node.
5. Return stored data.

Peek():

Return top->data.

Display():

Traverse nodes from top until NULL.

Applications of Stack

1. Expression Evaluation: Used in evaluation of postfix/prefix expressions.
2. Function Calls: Maintains return addresses and local variables.
3. Undo/Redo Operations: In editors, browsers, etc.
4. Parenthesis Matching: Useful in compilers and parsers.
5. Backtracking Algorithms: E.g., maze solving, recursion.

References

1. Yedidyah Langsam, Moshe J Augenstein, Aaron M Tenenbaum – Data structures using C and C++ - PHI Publications (2nd Edition).
2. Ellis Horowitz, Sataraj Sahni- Fundamentals of Data Structures – Galgotia Books source.

Questions

1. form of access is used to add remove nodes from a stack.
 - a) LIFO
 - b) FIFO
 - c) Both A and B
 - d) None of these

SE(ECE)	Data Structures and Algorithms	
Experiment No: 6	Implement Stack using a) arrays and b) linked list.	Page: 4/6

- 2. The retrieval of items in a stack is operation.**
 - a) push
 - b) pop
 - c) retrieval
 - d) access
- 3. The term push and pop is related to**
 - a) Array
 - b) Lists
 - c) Stacks
 - d) Trees
- 4. Stack follows the strategy of**
 - a) LIFO
 - b) FIFO
 - c) LRU
 - d) RANDOM
- 5. Which of the following is an application of stack?**
 - a) finding factorial
 - b) tower of Hanoi
 - c) infix to postfix
 - d) all of the above
- 6. The data structure required to check whether an expression contains balanced parenthesis is?**
 - a) Stack
 - b) Queue
 - c) Array
 - d) Tree
- 7. What data structure would you mostly likely see in a non recursive implementation of a recursive algorithm?**
 - a) LinkList
 - b) Stack
 - c) Queue
 - d) Tree
- 8. Which data structure is needed to convert infix notation to postfix notation?**
 - a) Branch
 - b) Tree
 - c) Queue
 - d) Stack

SE(ECE)	Data Structures and Algorithms	
Experiment No: 6	Implement Stack using a) arrays and b) linked list.	Page: 5/6

9. Which of the following statement(s) about stack data structure is/are NOT correct?

- a) Stack data structure can be implemented using linked list
- b) New node can only be added at the top of the stack
- c) Stack is the FIFO data structure
- d) The last node at the bottom of the stack has a NULL link

10. Consider the following operation performed on a stack of size 5.

Push(1);

Pop();

Push(2);

Push(3);

Pop();

Push(4);

Pop();

Pop();

Push(5);

After the completion of all operation, the no of element present on stack are

- a) 1
- b) 2
- c) 3
- d) 4

11. If the elements “A”, “B”, “C” and “D” are placed in a stack and are deleted one at a time, in what order will they be removed?

- a) ABCD
- b) DCBA
- c) DCAB
- d) ABDC

12. Consider the following array implementation of stack:

```
#define MAX 10
struct STACK
{
    int arr [MAX];
    int top;
};
```

If the array index starts with 0, the maximum value of top which does not cause stack overflow is?

- a) 8
- b) 9
- c) 10
- d) 11

SE(ECE)	Data Structures and Algorithms	
Experiment No: 6	Implement Stack using a) arrays and b) linked list.	Page: 6/6

- 13. Following sequence of operations is performed on a stack push(1), push(2), pop, push(1),push(2) pop, pop, pop, push(2), pop. The sequence of popped out values are**
- a) 2,2,1,1,2
 - b) 2,2,1,2,2
 - c) 2,1,2,2,1
 - d) 2,1,2,2,2

Conclusion

A	C	O	T	Sign.