CAM2003C - Data Structures and Algorithms with C and C++

Lab Exercise -6: Stack Implementation using Array and Linked Lists and Applications of Stack ADT

Practical Questions on Stack

1. Write a C program to implement a stack using an array with the following operations:

    1. push()
    2. pop()
    3. peek()
    4. display()

2. Repeat the above with menu driven approach.

3. Reimplement Q-1 in C++ using a class-based approach.

4. Implement a stack using a singly linked list with push, pop, and display in C.

5. Repeat the above with menu driven approach.

6. Implement stack operations using Linked List in C++ with classes and objects.

7. Write a program to demonstrate the use of the stack container in C++ STL by performing:

    - Push elements
    - Pop elements
    - Access top() element
    - Display size and empty status

8. Write a program in C to implement two stacks in one array such that:

    - Stack 1 grows from left to right.
    - Stack 2 grows from right to left.
    - The program should support push1(), push2(), pop1(), pop2() and check for overflow/underflow conditions.

Hint:

- Use a single array arr[MAX].
- Maintain two top variables:

```
int top1 = -1;
```

int top2 = MAX;

- Stack1 elements go into arr[++top1].
- Stack2 elements go into arr[--top2].
- Overflow occurs if top1 + 1 == top2.

9. Repeat the above in C++ using a class-based implementation.

Applications of Stack

10. Write a program to reverse a string using a stack implemented with arrays.

11. Write a program to reverse a string in C++ using the stack STL.

12. Write a program in c to reverse an integer array using a stack (implemented with array).

13. Write a program in C to check whether a given expression has balanced parentheses using a stack (array/linked list).

14. Repeat the above using STL stack.in

15. Write a program in C to convert an infix expression to a postfix expression using a stack.

16. Repeat the above in C++ using STL stack.

17. Write a program in C to convert an infix expression to a prefix expression using a stack.

18. Repeat the above in C++ using STL stack.

19. Write a program in C to evaluate a postfix expression.

20. Repeat the above in C++ using STL stack.

21. Write a program in C to evaluate a prefix expression.

22. Repeat the above in C++ using STL stack.