EXPERIMENT NO. 3

AIM: Write a program to perform various operations on Stack.

SCOPE: Insertion and Deletion are the basic data structure operation. Through this experiment we will come to know that how insertion and deletion operation will be performed on Stack data structure.

FACILITIES: Software Needed: Turbo C

THEORY:

A stack is an Abstract Data Type (ADT), commonly used in most programming languages. It is named stack as it behaves like a real-world stack, for example – a deck of cards or a pile of plates, etc.

A real-world stack allows operations at one end only. For example, we can place or remove a card or plate from the top of the stack only. Likewise, Stack ADT allows all data operations at one end only. At any given time, we can only access the top element of a stack.

This feature makes it LIFO data structure. LIFO stands for Last-in-first-out. Here, the element which is placed (inserted or added) last, is accessed first. In stack terminology, insertion operation is called PUSH operation and removal operation is called POP operation.

Basic Operations

Stack operations may involve initializing the stack, using it and then de-initializing it. Apart from these basic stuffs, a stack is used for the following two primary operations −

- push() − Pushing (storing) an element on the stack.
- pop() − Removing (accessing) an element from the stack.

IMPLEMENTATION:

```
// PROGRAM TO PERFORM VARIOUS OPERATION ON STACK
#include <stdio.h>

#define MAX_SIZE 100

int stack[MAX_SIZE];
int top = -1;

// Function to push an element onto the stack
void push(int element) {
    if (top == MAX_SIZE - 1) {
        printf("Stack overflow\n");
        return;
    }
    stack[++top] = element;
    printf("%d pushed to stack\n", element);
}

// Function to pop an element from the stack
int pop() {
    if (top == -1) {
        printf("Stack underflow\n");
        return -1;
    }
    printf("%d popped from stack\n", stack[top]);
    return stack[top--];
}

// Function to display the elements in the stack
void display() {
    if (top == -1) {
        printf("Stack is empty\n");
        return;
    }
    printf("Elements in stack are:\n");
    for (int i = top; i >= 0; i--) {
        printf("%d\n", stack[i]);
    }
}

int main() {
    int choice, element;
    while (1) {
        printf("\nStack Operations:\n");
        printf("1. Push\n");
        printf("2. Pop\n");
        printf("3. Display\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter element to push: ");
                scanf("%d", &element);
                push(element);
                break;
            case 2:
                pop();
                break;
            case 3:
                display();
                break;
            case 4:
                printf("Exiting program\n");
                return 0;
            default:
                printf("Invalid choice\n");
        }
    }
```

```
}
```

Output:

```
Stack Operations:
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 1
Enter element to push: 24
24 pushed to stack

Stack Operations:
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 1
Enter element to push: 38
38 pushed to stack

Stack Operations:
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 1
Enter element to push: 56
56 pushed to stack

Stack Operations:
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 2
Enter element to push: 56
56 pushed to stack

Stack Operations:
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 3
Elements in stack are:
38
34
```

RESULT: In this way we have implemented Insertion and Deletion i.e. PUSH & POP operations on stack with Turbo Cand tested with examples.