

LAB PROGRAM - 01

1. Write a program to simulate the working of stack using an array with the following:

- a) Push
- b) Pop
- c) Display

The program should print appropriate messages for stack overflow, stack underflow

```
#include <stdio.h>
#define MAXSIZE 5 // Maximum size of stack

int stack[MAXSIZE];
int top = -1; // Initialize stack as empty

// Function to push an element
void push(int value) {
    if (top == MAXSIZE - 1)
    {
        printf("Stack Overflow! Cannot push %d\n", value);
    }
    else
    {
        top = top + 1; // Increment top
        stack[top] = value; // Insert value
        printf("%d pushed into stack.\n", value);
    }
}

// Function to pop an element
void pop()
{
    if (top == -1)
    {
        printf("Stack Underflow! Nothing to pop.\n");
    }
    else
    {
        printf("Popped element: %d\n", stack[top]);
        top = top - 1; // Decrement top
    }
}
```

```

// Function to peek (view top element)
void peek()
{
    if (top == -1)
    {
        printf("Stack is empty! Nothing to peek.\n");
    } else {
        printf("Top element: %d\n", stack[top]);
    }
}

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

int main()
{
    int choice, value;

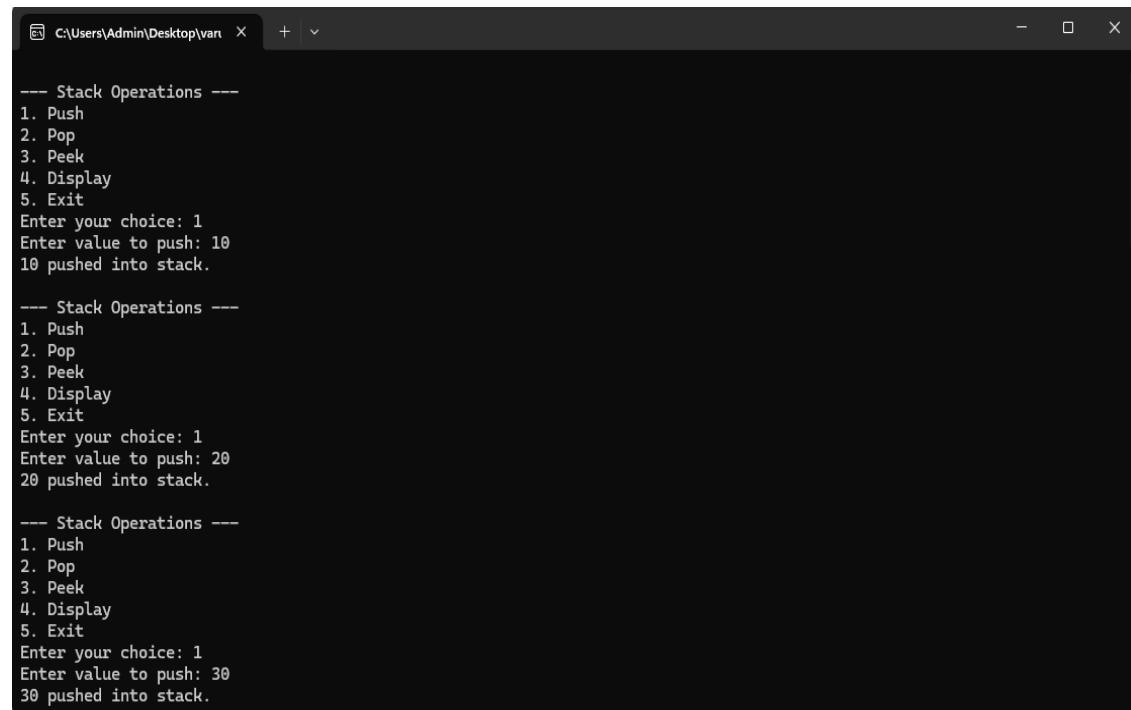
    while (1)
    {
        printf("\n--- Stack Operations ---\n");
        printf("1. Push\n2. Pop\n3. Peek\n4. Display\n5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice)
        {
            case 1:
                printf("Enter value to push: ");
                scanf("%d", &value);
                push(value);
                break;
        }
    }
}

```

```
case 2:  
    pop();  
    break;  
case 3:  
    peek();  
    break;  
case 4:  
    display();  
    break;  
case 5:  
    printf("Exiting program...\n");  
    return 0;  
default:  
    printf("Invalid choice! Try again.\n");  
}  
}  
}
```

Output:



The screenshot shows a terminal window titled 'C:\Users\Admin\Desktop\var' with a dark theme. It displays three separate runs of a stack operation program. Each run starts with a menu of five options: Push, Pop, Peek, Display, and Exit. The user enters '1' for Push, followed by a value to push. The program then outputs a confirmation message like '10 pushed into stack.' or similar.

```
--- Stack Operations ---  
1. Push  
2. Pop  
3. Peek  
4. Display  
5. Exit  
Enter your choice: 1  
Enter value to push: 10  
10 pushed into stack.  
  
--- Stack Operations ---  
1. Push  
2. Pop  
3. Peek  
4. Display  
5. Exit  
Enter your choice: 1  
Enter value to push: 20  
20 pushed into stack.  
  
--- Stack Operations ---  
1. Push  
2. Pop  
3. Peek  
4. Display  
5. Exit  
Enter your choice: 1  
Enter value to push: 30  
30 pushed into stack.
```

```
C:\Users\Admin\Desktop\var + - X

--- Stack Operations ---
1. Push
2. Pop
3. Peek
4. Display
5. Exit
Enter your choice: 1
Enter value to push: 40
40 pushed into stack.

--- Stack Operations ---
1. Push
2. Pop
3. Peek
4. Display
5. Exit
Enter your choice: 1
Enter value to push: 50
50 pushed into stack.

--- Stack Operations ---
1. Push
2. Pop
3. Peek
4. Display
5. Exit
Enter your choice: 1
Enter value to push: 60
Stack Overflow! Cannot push 60
```

```
C:\Users\Admin\Desktop\var + - X

--- Stack Operations ---
1. Push
2. Pop
3. Peek
4. Display
5. Exit
Enter your choice: 2
Popped element: 50

--- Stack Operations ---
1. Push
2. Pop
3. Peek
4. Display
5. Exit
Enter your choice: 3
Top element: 40
```

```
--- Stack Operations ---
1. Push
2. Pop
3. Peek
4. Display
5. Exit
Enter your choice: 4
Stack elements are:
40
30
20
10

--- Stack Operations ---
1. Push
2. Pop
3. Peek
4. Display
5. Exit
Enter your choice: 5
Exiting program...

Process returned 0 (0x0) execution time : 209.140 s
Press any key to continue.
```