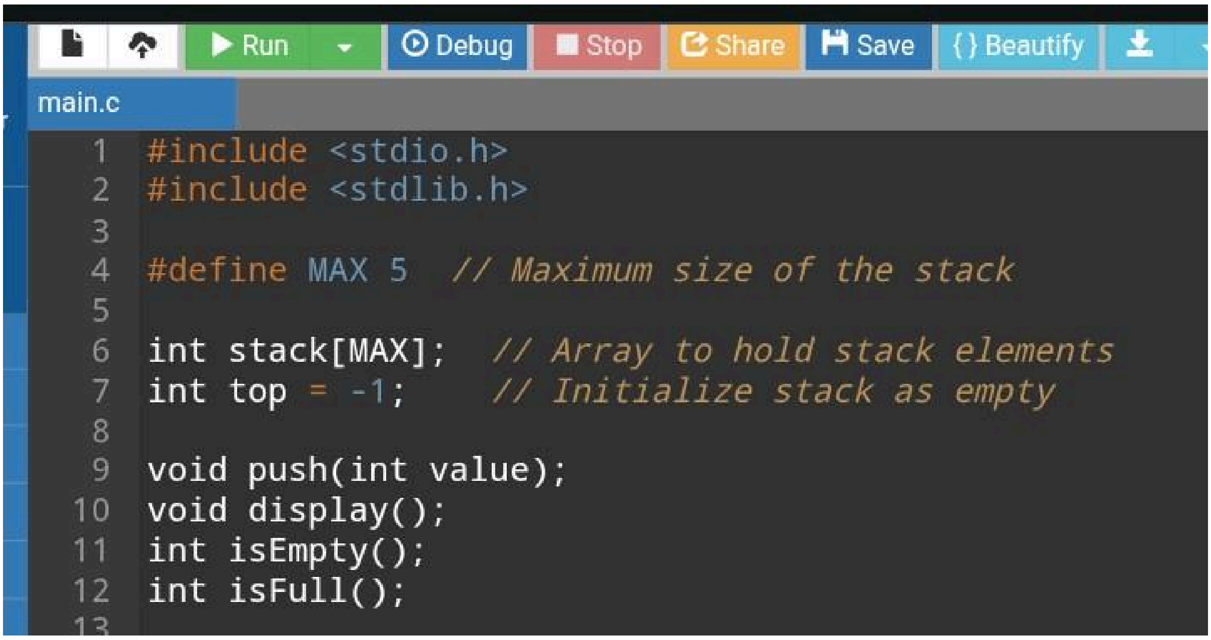
Experiment No: 04

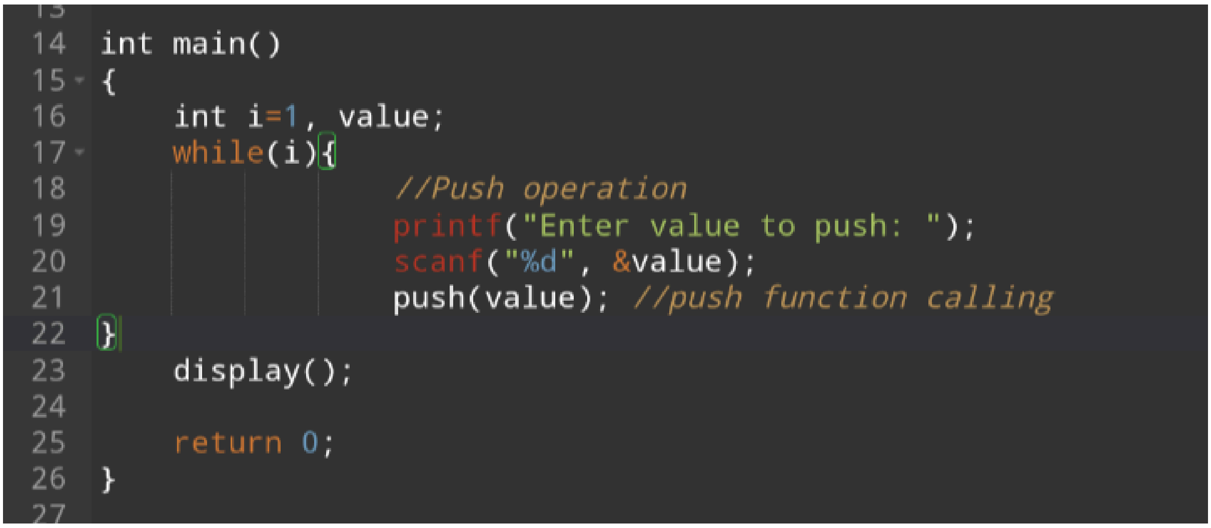
Experiment Name: Stack Push Operation

Objective: To implement the push operation of a stack using an array in C and understand how elements are inserted following the LIFO (Last In, First Out) principle.

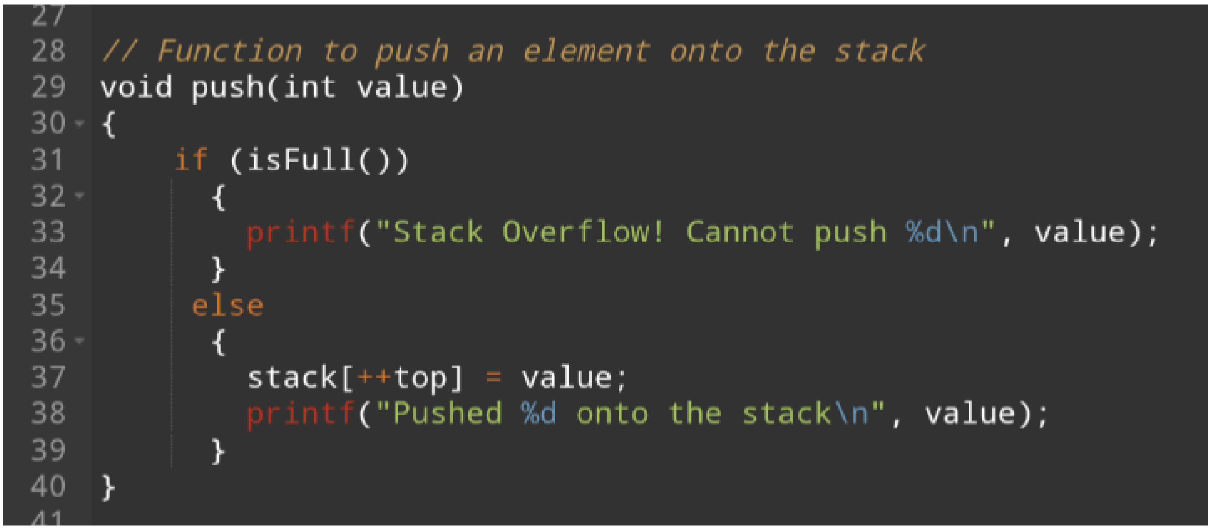
Source Code:



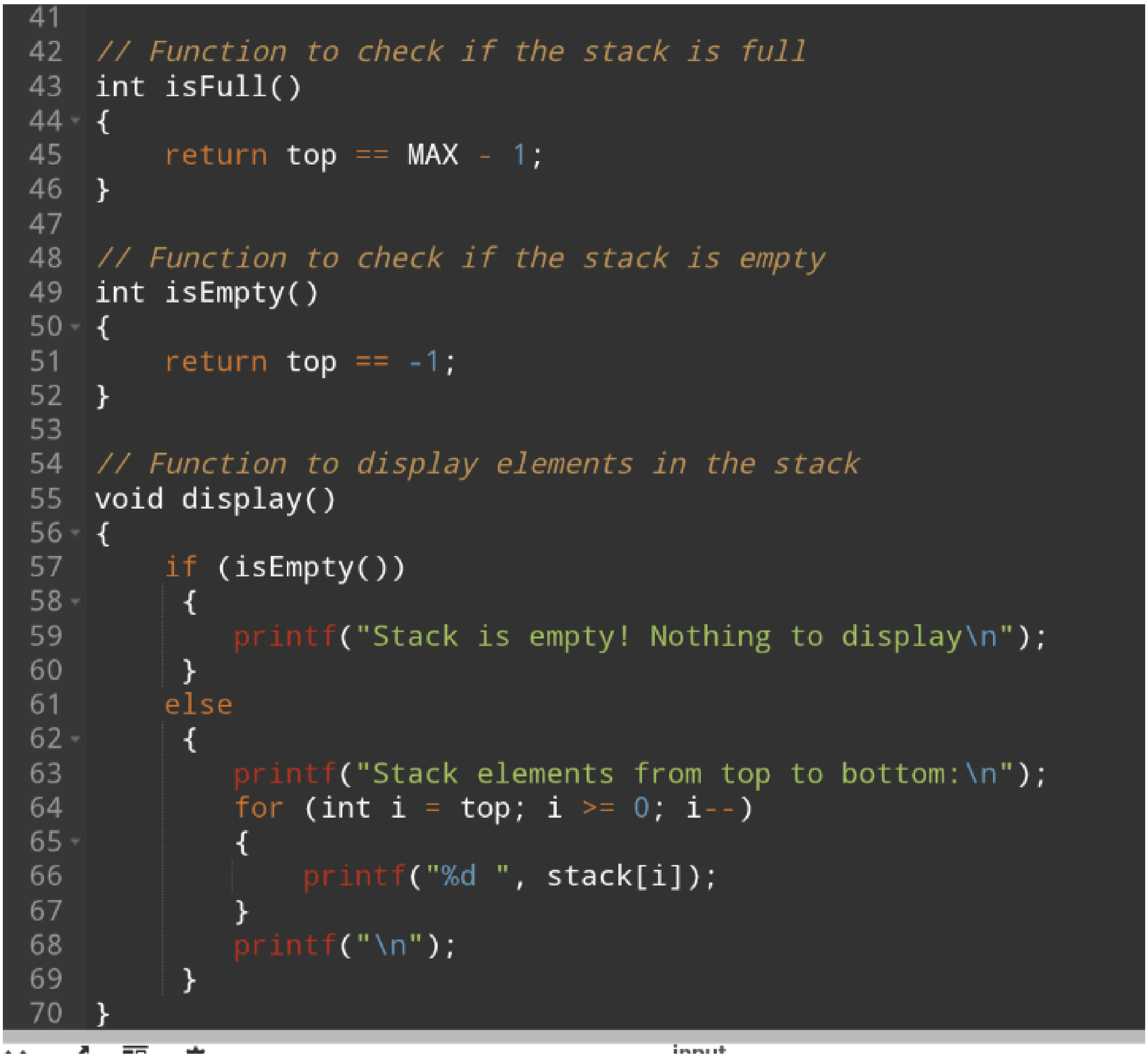
```c
#include <stdio.h>
#include <stdlib.h>

#define MAX 5  // Maximum size of the stack

int stack[MAX];   // Array to hold stack elements
int top = -1;     // Initialize stack as empty

void push(int value);
void display();
int isEmpty();
int isFull();

```



```c
int main()
{
    int i=1, value;
    while(i){
                //Push operation
                printf("Enter value to push: ");
                scanf("%d", &value);
                push(value); //push function calling
}
    display();

    return 0;
}

```



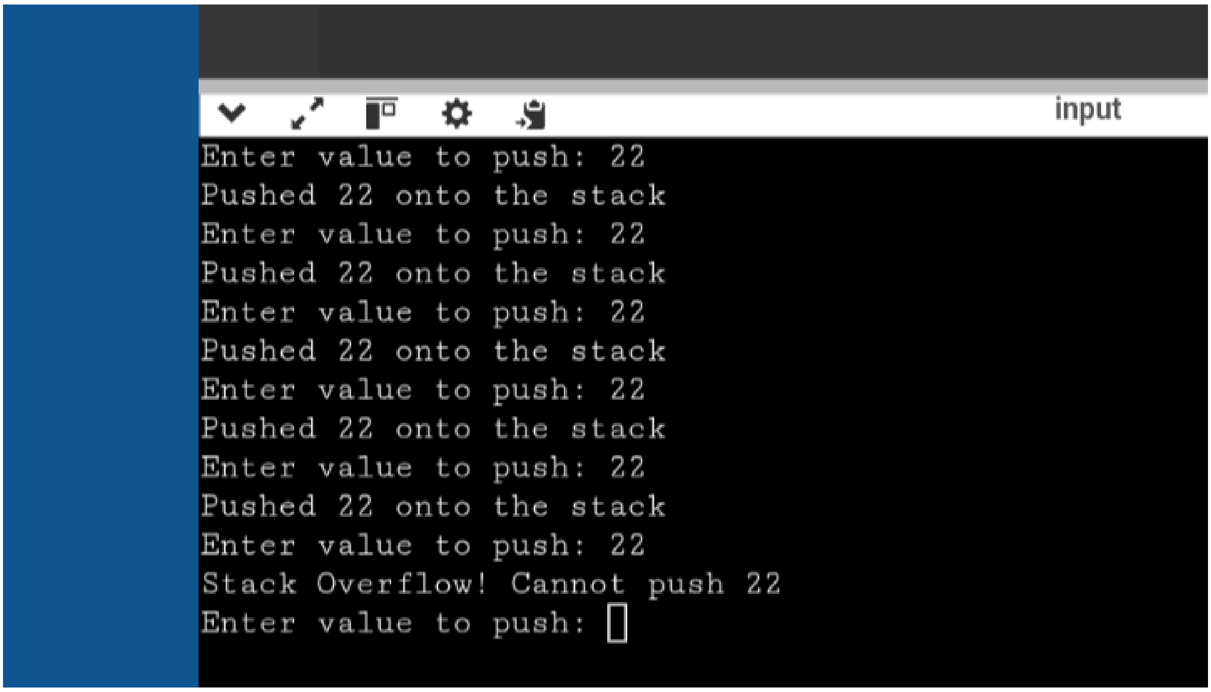
```c

// Function to push an element onto the stack
void push(int value)
{
    if (isFull())
      {
        printf("Stack Overflow! Cannot push %d\n", value);
      }
     else
      {
        stack[++top] = value;
        printf("Pushed %d onto the stack\n", value);
      }
}
```



```c

// Function to check if the stack is full
int isFull()
{
    return top == MAX - 1;
}

// Function to check if the stack is empty
int isEmpty()
{
    return top == -1;
}

// Function to display elements in the stack
void display()
{
    if (isEmpty())
     {
        printf("Stack is empty! Nothing to display\n");
     }
    else
     {
        printf("Stack elements from top to bottom:\n");
        for (int i = top; i >= 0; i--)
        {
            printf("%d ", stack[i]);
        }
        printf("\n");
     }
}
```

Fig 01: Stack Push Operation c code

Output:



```
Enter value to push: 22
Pushed 22 onto the stack
Enter value to push: 22
Pushed 22 onto the stack
Enter value to push: 22
Pushed 22 onto the stack
Enter value to push: 22
Pushed 22 onto the stack
Enter value to push: 22
Pushed 22 onto the stack
Enter value to push: 22
Stack Overflow! Cannot push 22
Enter value to push: 
```

Fig 02: Output

Discussion: In this experiment, the push operation of a stack was successfully implemented in C using an array. The program allows users to insert values interactively until they decide to stop. Before each insertion, the program checks whether the stack is full, which prevents overflow and ensures safe memory usage. The stack elements are displayed from top to bottom, clearly showing the LIFO order of insertion. This implementation helps to understand how the top pointer plays a crucial role in managing stack operations. Overall, the experiment demonstrates the practical use of conditional checks and functions in handling stack data structures efficiently.