11.Implementation of stack operations such as push pop and peek

Sample code:

```
#include <stdio.h>
#define SIZE 100

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

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

// Function to pop an element from the stack
void pop() {
    if (top == -1) {
        printf("Stack Underflow! Cannot pop.\n");
    } else {
        printf("%d popped from stack.\n", stack[top]);
        top--;
    }
}

// Function to peek at the top element of the stack
```

```
void peek() {
    if (top == -1) {
        printf("Stack is empty.\n");
    } else {
        printf("Top element is: %d\n", stack[top]);
    }
}

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

int main() {
    int choice, value;

    while (1) {
        printf("\n--- Stack Menu ---\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. Please try again.\n");
}
```

```
    }

    return 0;
}
```

Output: