

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:

The screenshot shows a C++ IDE with the following components:

- Source Code (experiment3.cpp):**

```
1 #include <stdio.h>
2 #define SIZE 100
3 int stack[SIZE];
4 int top = -1;
5 void push(int value) {
6     if (top == SIZE - 1) {
7         printf("Stack Overflow! Cannot push %d\n", value);
8     } else {
9         top++;
10        stack[top] = value;
11        printf("%d pushed to stack.\n", value);
12    }
13 }
14 void pop() {
15     if (top == -1) {
16         printf("Stack Underflow! Cannot pop.\n");
17     } else {
18         printf("%d popped from stack.\n", stack[top]);
19         top--;
20     }
21 }
22 void peek() {
23     if (top == -1) {
24         printf("Stack is empty.\n");
25     } else {
26         printf("Top element is: %d\n", stack[top]);
27     }
28 }
29 void display() {
30     if (top == -1) {
31         printf("Stack is empty.\n");
32     } else {
```
- Output Window:** Displays the execution output, showing successful push and pop operations and error messages for overflow and underflow.
- Compiler Log:** Shows compilation results with 0 errors and 0 warnings. The output file is named C:\Users\chendi\OneDrive\servo.pro\Pictures\Documents\experiment 3.exe.
- Status Bar:** Indicates the current line (13), column (2), and other file statistics.