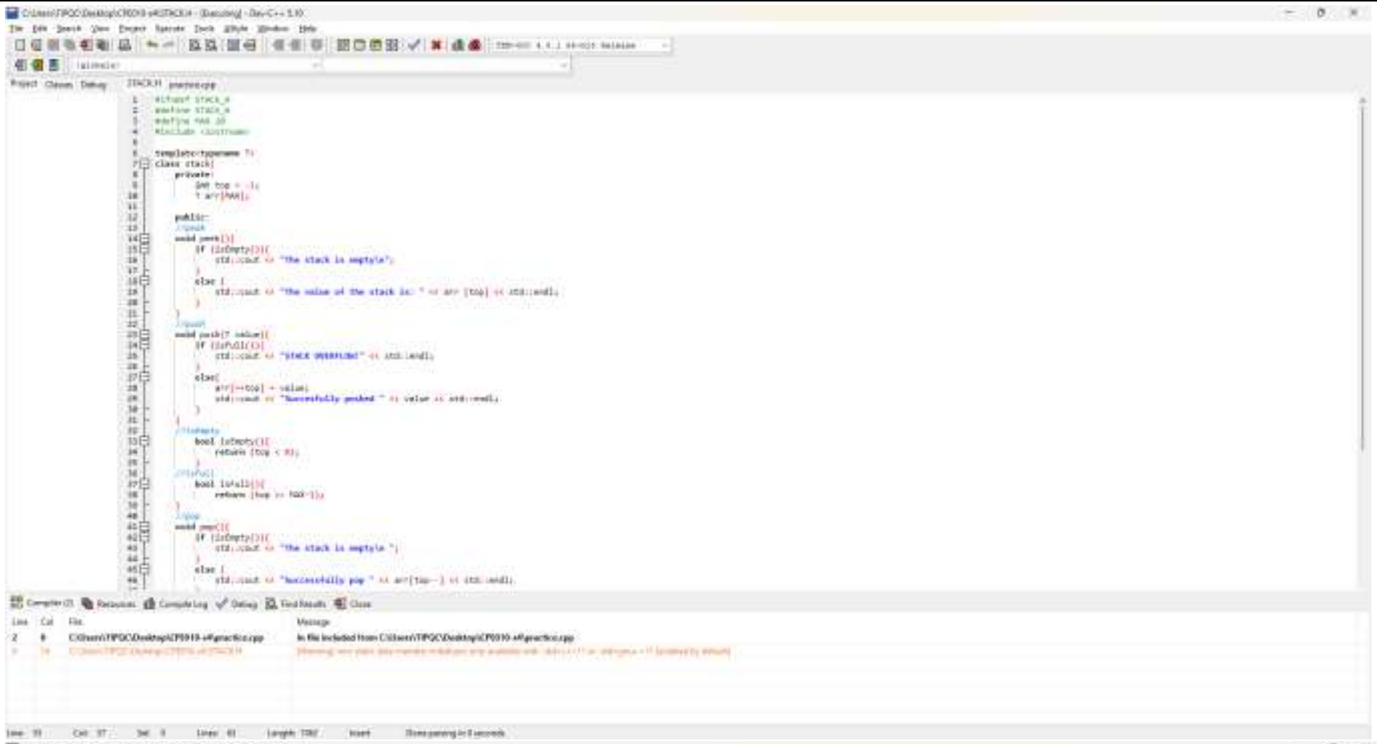
Activity No. 4.2	
4.2 Stacks	
Course Code: CPE010	**Program:** Computer Engineering
Course Title: Data Structures and Algorithms	**Date Performed:8/28/2025**
Section: CPE21S4	**Date Submitted:8/28/2025**
Name(s):Crishen Luper S. Pulgadp	**Instructor: Engr. Jimlord Quejado**

6. Output

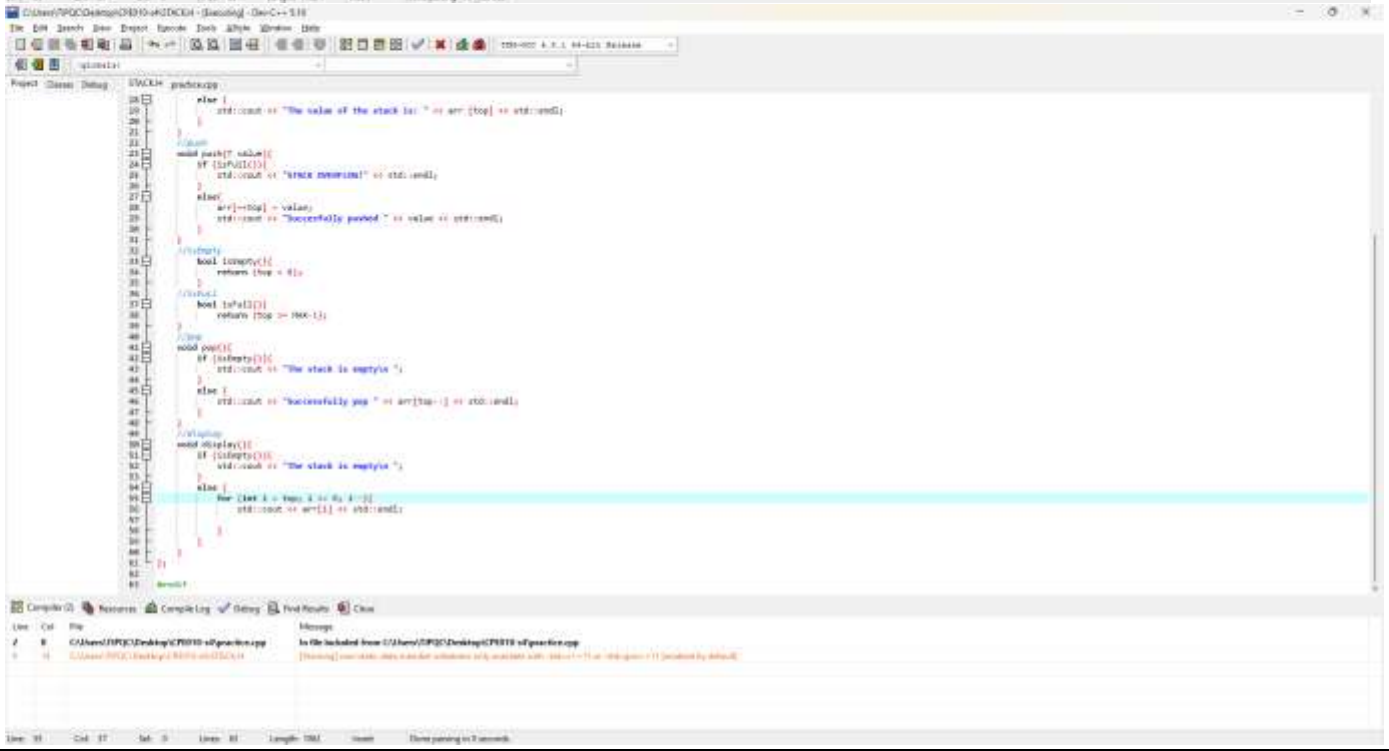


```cpp
#ifndef STACK_H
#define STACK_H
#define MAX 10
#include <iostream>

template<typename T>
class stack{
    private:
        int top = -1;
        T arr[MAX];

    public:
    //peek
    void peek(){
        if (isEmpty()){
            std::cout << "The stack is empty\n";
        }
        else {
            std::cout << "The value of the stack is: " << arr [top] << std::endl;
        }
    }
    //push
    void push(T value){
        if (isFull()){
            std::cout << "STACK OVERFLOW!" << std::endl;
        }
        else{
            arr[++top] = value;
            std::cout << "Successfully pushed " << value << std::endl;
        }
    }
    //isEmpty
        bool isEmpty(){
            return (top < 0);
        }
    //isFull
        bool isFull(){
            return (top >= MAX-1);
    }
    //pop
    void pop(){
        if (isEmpty()){
            std::cout << "The stack is empty\n ";
        }
        else {
            std::cout << "Successfully pop " << arr[top--] << std::endl;
```

```
Line  Col  File                                             Message
2     8    C:\Users\TIPQC\Desktop\CPE010-s4\practice.cpp    In file included from C:\Users\TIPQC\Desktop\CPE010-s4\practice.cpp
9     14   C:\Users\TIPQC\Desktop\CPE010-s4\STACK.H         [Warning] non-static data member initializers only available with -std=c++11 or -std=gnu++11 [enabled by default]
```

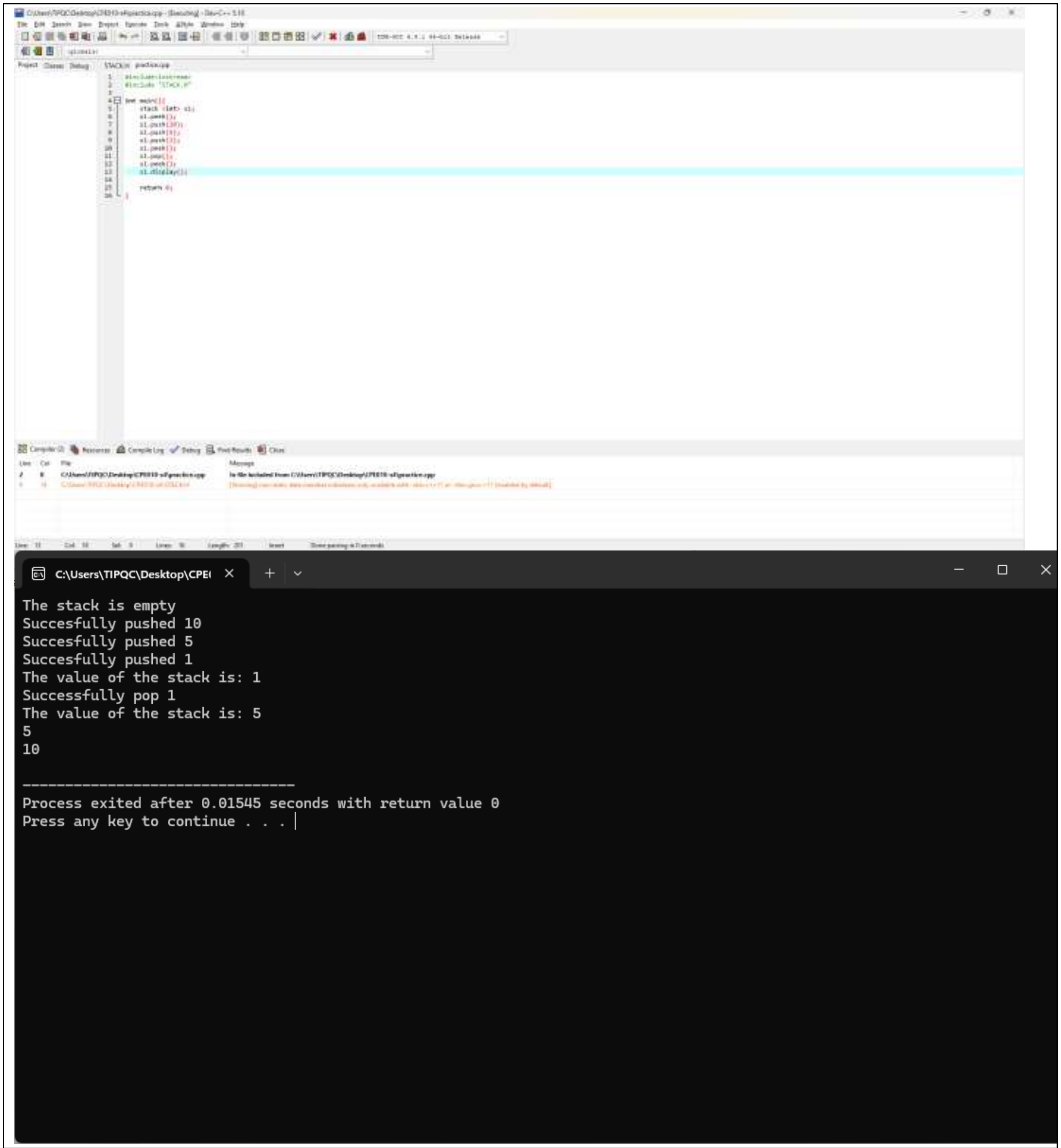


```cpp
        else {
            std::cout << "The value of the stack is: " << arr [top] << std::endl;
        }
    }
    //push
    void push(T value){
        if (isFull()){
            std::cout << "STACK OVERFLOW!" << std::endl;
        }
        else{
            arr[++top] = value;
            std::cout << "Successfully pushed " << value << std::endl;
        }
    }
    //isEmpty
        bool isEmpty(){
            return (top < 0);
        }
    //isFull
        bool isFull(){
            return (top >= MAX-1);
    }
    //pop
    void pop(){
        if (isEmpty()){
            std::cout << "The stack is empty\n ";
        }
        else {
            std::cout << "Successfully pop " << arr[top--] << std::endl;
        }
    }
    //display
    void display(){
        if (isEmpty()){
            std::cout << "The stack is empty\n ";
        }
        else {
            for (int i = top; i >= 0; i--){
                std::cout << arr[i] << std::endl;

            }
        }
    }
};

#endif
```

```
Line  Col  File                                             Message
2     8    C:\Users\TIPQC\Desktop\CPE010-s4\practice.cpp    In file included from C:\Users\TIPQC\Desktop\CPE010-s4\practice.cpp
9     14   C:\Users\TIPQC\Desktop\CPE010-s4\STACK.H         [Warning] non-static data member initializers only available with -std=c++11 or -std=gnu++11 [enabled by default]
```



```cpp
#include<iostream>
#include "STACK.h"

int main(){
	stack <int> s1;
	s1.peek();
	s1.push(10);
	s1.push(5);
	s1.push(1);
	s1.peek();
	s1.pop();
	s1.peek();
	s1.display();

	return 0;
}
```

```
The stack is empty
Succesfully pushed 10
Succesfully pushed 5
Succesfully pushed 1
The value of the stack is: 1
Successfully pop 1
The value of the stack is: 5
5
10

--------------------------------
Process exited after 0.01545 seconds with return value 0
Press any key to continue . . .
```

7. Supplementary Activity

8. Conclusion

To conclude, we can use stacks in arrays with declaring its fixed size. Through using the operations in stack, we can change the elements inside via adding or removing by following the properties of LIFO. In this activity, I think I had learned well how does stack work and arrays.

9. Assessment Rubric