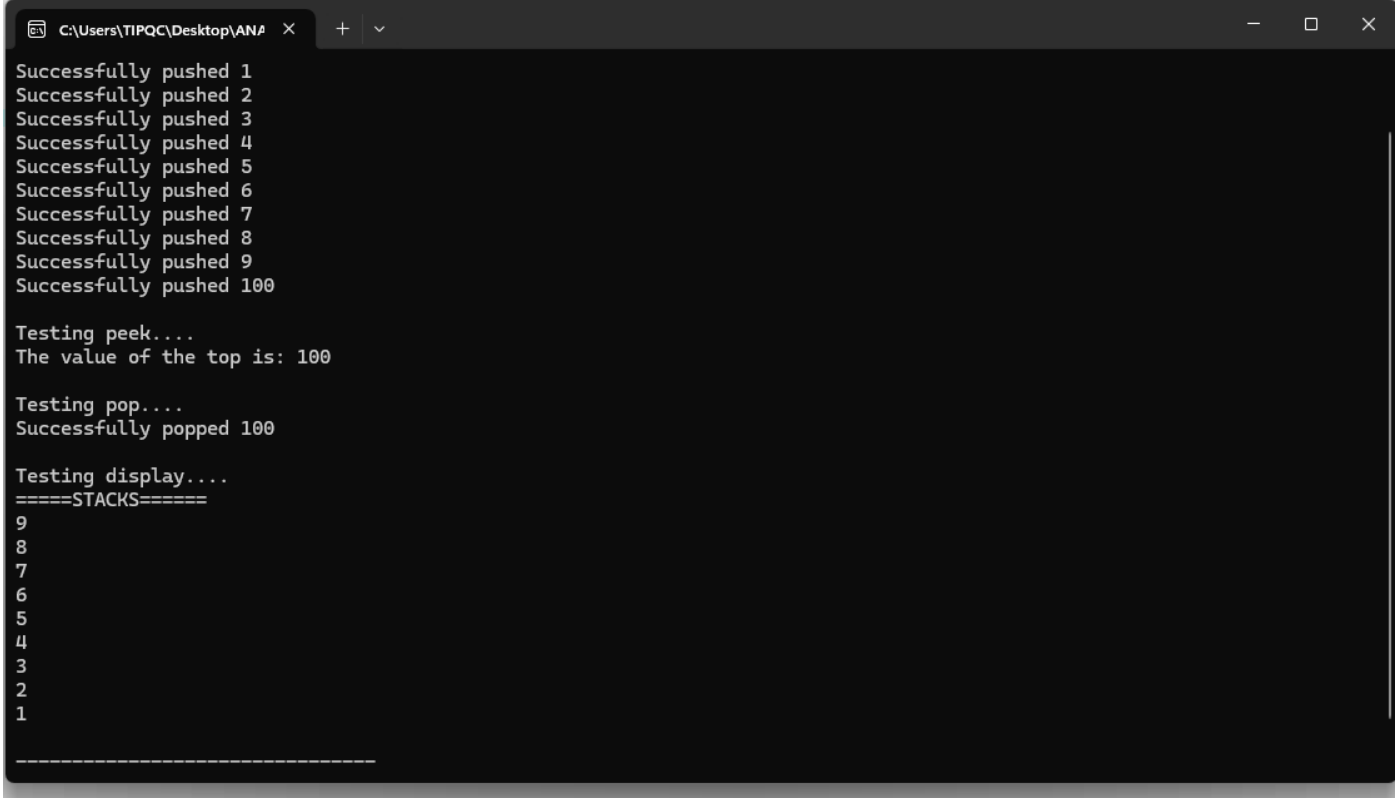


Activity No. <4.2>	
<STACKS>	
Course Code: CPE010	Program: Computer Engineering
Course Title: Data Structures and Algorithms	Date Performed: 8/28/2025
Section: CPE21S4	Date Submitted: 8/28/2025
Name(s): Anastacio, Lester Arvid P.	Instructor: Engr. Jimlord Quejado

6. Output

SCREENSHOTS:

A screenshot of a terminal window with a dark background and white text. The window title bar shows the file path 'C:\Users\TIPQC\Desktop\ANA#'. The terminal output shows a series of 'Successfully pushed' messages for values 1 through 100. It then shows 'Testing peek...' followed by 'The value of the top is: 100'. Next is 'Testing pop...' followed by 'Successfully popped 100'. Finally, it shows 'Testing display...' followed by a header '====STACKS====' and a list of numbers from 9 down to 1. A dashed line is at the bottom of the output.

```
C:\Users\TIPQC\Desktop\ANA# x + v
Successfully pushed 1
Successfully pushed 2
Successfully pushed 3
Successfully pushed 4
Successfully pushed 5
Successfully pushed 6
Successfully pushed 7
Successfully pushed 8
Successfully pushed 9
Successfully pushed 100

Testing peek...
The value of the top is: 100

Testing pop...
Successfully popped 100

Testing display...
====STACKS====
9
8
7
6
5
4
3
2
1
-----
```

7. Supplementary Activity

Main File:

```
2  #include "stack.h"
3
4  int main(){
5      stack <int> s1;
6
7      //isEmpty
8      std::cout<<"Testing isEmpty: \n";
9      std::cout<<s1.isEmpty()<<std::endl<<std::endl;
10
11     //push
12     for (int i = 1; i < 10; i++){
13         s1.push(i);
14     }
15
16     //stack overflow
17     s1.push(100);
18
19     //peek
20     std::cout<<std::endl;
21     std::cout<<"Testing peek....\n";
22     s1.peek();
23
24     //pop
25     std::cout<<std::endl;
26     std::cout<<"Testing pop....\n";
27     s1.pop();
28
29     //display
30     std::cout<<std::endl;
31     std::cout<<"Testing display....\n";
32     s1.display();
33
34
35
36     return 0;
37 }
```

Header File:

```
1  #ifndef STACK_H
2  #define STACK_H
3  #define MAX 10
4  #include <iostream>
5  template <typename T>
6  class stack{
7  private:
8      int top = -1;
9      T arr[MAX];
10
11 public:
12     //Functions of a stack
13     //isEmpty
14     bool isEmpty(){
15         return (top < 0);
16     }
17
18     //isFull
19     bool isFull(){
20         return (top >= MAX-1);
21     }
22
23     //peek
24     void peek(){
25         if (isEmpty()){
26             std::cout << "The Stack is Empty\n";
27         }
28         else{
29             std::cout<<"The value of the top is: "<<arr[top]<<std::endl;
30         }
31     }
32     //push
33     void push(T value){
34         if (isFull()){
35             std::cout<<"STACK OVERFLOW!!"<<std::endl;
36         }
37         else{
38             arr[++top] = value;
39             std::cout<<"Successfully pushed "<<value<<std::endl;
40         }
41     }
42
43     //pop
44     void pop(){
45         if (isEmpty()){
46             std::cout<<"The Stack is Empty\n";
47         }
48         else{
49             std::cout<<"Successfully popped "<<arr[top--]<<std::endl;
50         }
51     }
52 }
```

```

    }
}
//pop
void pop(){
    if (isEmpty()){
        std::cout<<"The Stack is Empty\n";
    }
    else{
        std::cout<<"Successfully popped "<<arr[top--]<<std::endl;
    }
}
//display
void display(){
    if (isEmpty()){
        std::cout<<"The Stack is Empty\n";
    }
    else{
        std::cout<<"====STACKS====\n";
        for (int i = top; i >= 0; i--){
            std::cout<<arr[i]<<std::endl;
        }
    }
}
};

#endif

```

8. Conclusion

During this activity, I've learned how to properly create stacks from scratch and also learned the uses of header files, perimeters and templates to be able to use all the needed data types for the whole code to work especially the push function as it needed perimeters and such to actually add in the data that we will input onto the stack.

9. Assessment Rubric