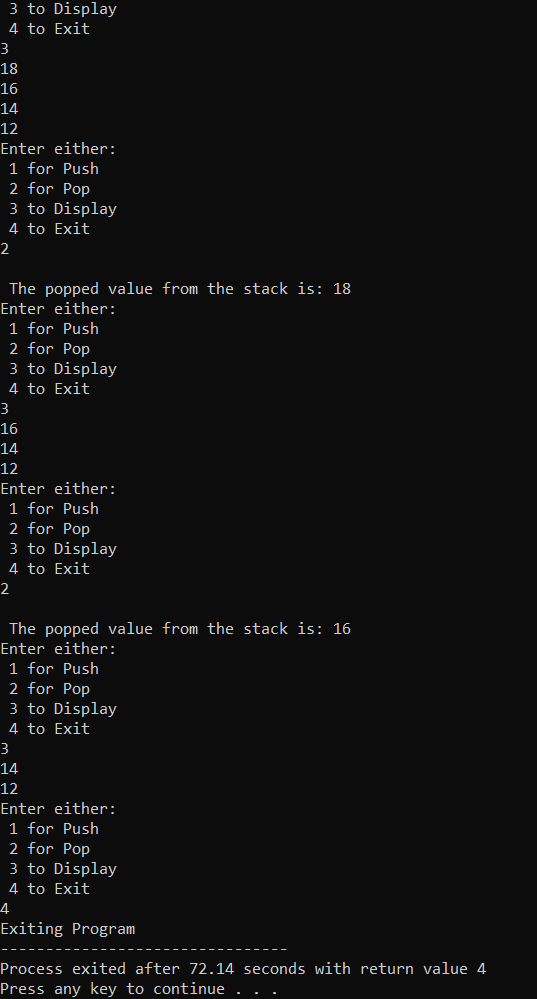
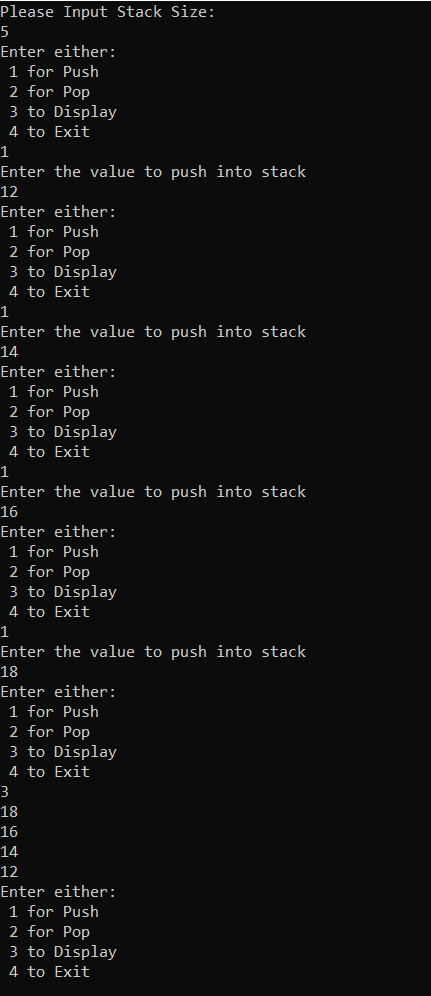
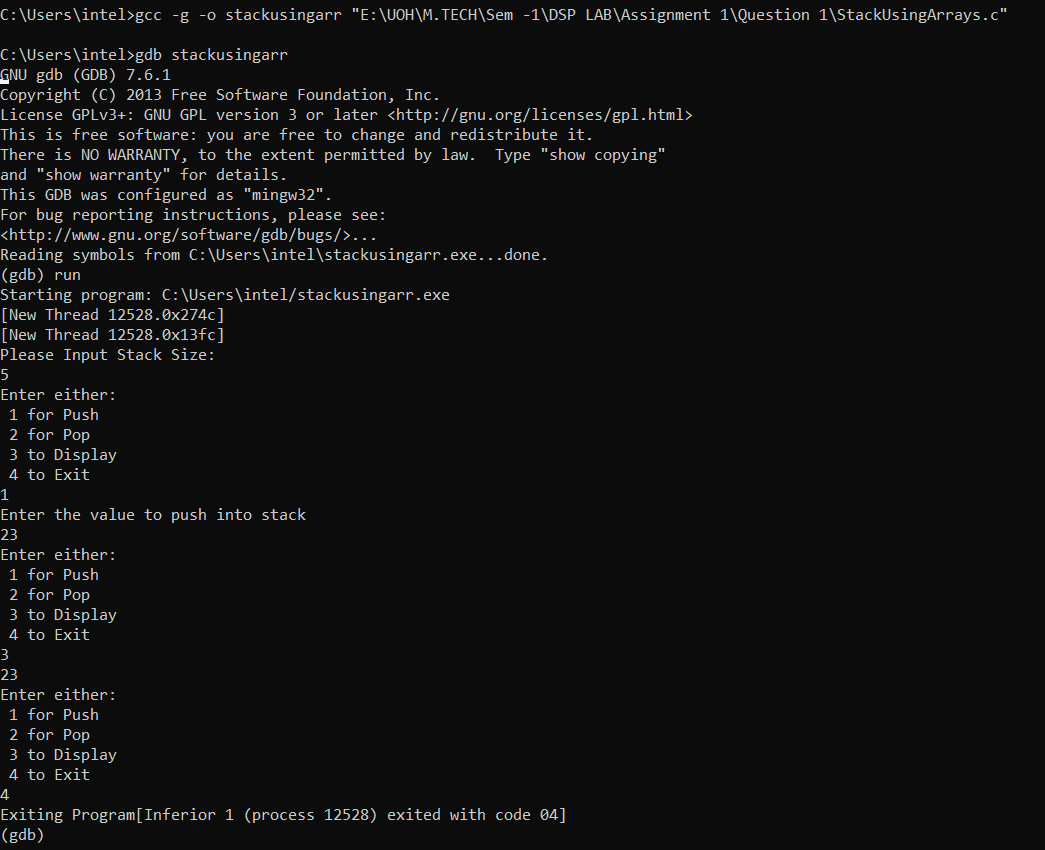
* 1. **Implement Stack using Arrays [Push, Pop, Display].**

Stack has been implemented using Arrays seperately.

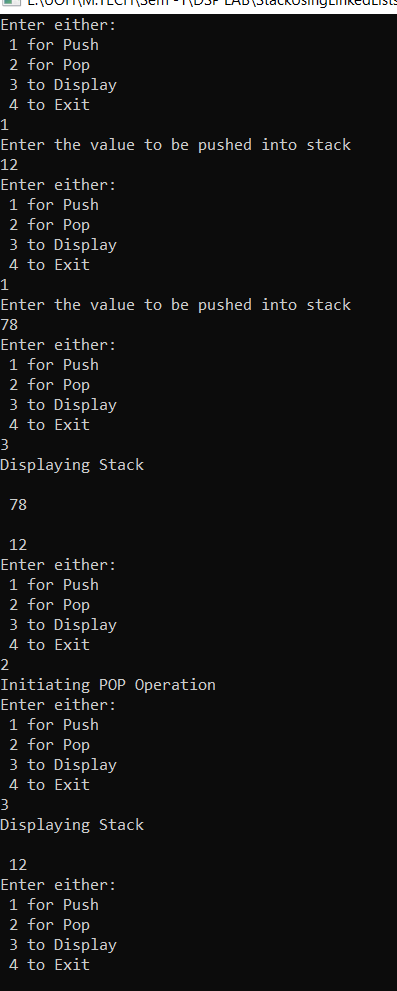
* Seperate functions are created: Push(), Pop(), and Display()
* **Top** is used as a variable which points to latest entered element in the array of stack.
* Push(): pushes the current element in the array and top is incremented by one.
* Pop(): is used to delete the topmost element in the stack, on deleting it, the top pointer is decremented by one, to point to the second last entered element.
* Display(): Uses a For Loop to traverse through the stack array and displays all the elements in the stack, If stack is empty it prints “Empty Stack”
* Main(): function uses a switch case to help the user decide what operation they want to perform, among push/pop/display/exit.
* **Execution Screenshots:  
  **

Debugger: 

* 1. **Implement Stack using Linked List [Push, Pop, Display].**

Stack has been implemented using LinkedLists seperately.

* Seperate functions are created: Push(), Pop(), and Display()
* An empty node is created at the beginning, and a **head** pointer is initialized.
* Head points to latest entered element in the array of stack.
* Push(): pushes the current element in the linkedlist by adding a newnode to the linked list and then the new node becomes head.
* Pop(): is used to delete the node data of the head node, on deleting it, the head node leaves the current node, and moves to head->next, and the current node is freed.
* Display(): Uses another pointer diplayptr and a while loop to traverse through the linked list and displays all the elements, If stack is empty it prints “Empty Stack”
* Main(): function uses a switch case to help the user decide what operation they want to perform, among push/pop/display/exit.
* uses a switch case to help the user decide what operation they want to perform, among push/pop/display/exit.

**Execution Screenshots:   
**

Debug Screenshots:  
