**LAB NO.**

**EXP.NO:**

**DATE: 7-10-22**

**STUDENT NAME: VAISHNAVI SINGH**

**ROLL NO: 2100290110181**

**AIM / OBJECTIVE:** Write a program in c to implement primitive operations(PUSH,POP & TRAVERSE) on stack using array.

**SAMPLE INPUT:**

**EXPECTED OUTPUT:**

**THEORY:** A stack is a linear data structure that follows the Last in, First out principle (i.e. the last added elements are removed first).

In array implementation, the stack is formed by using the array.

**PROEDURE:**

For PUSH-Adding an element into the top of the stack is referred to as push operation. Push operation involves following two steps.

1. Increment the variable Top so that it can now refere to the next memory location.
2. Add element at the position of incremented top. This is referred to as adding new element at the top of the stack.

For POP-Deletion of an element from the top of the stack is called pop operation. The value of the variable top will be incremented by 1 whenever an item is deleted from the stack. The top most element of the stack is stored in an another variable and then the top is decremented by 1. the operation returns the deleted value that was stored in another variable as the result.

**SOURCE CODE (TYPED):**

**#include <stdio.h>**

**int stack[100],i,j,choice=0,n,top=-1;**

**void push();**

**void pop();**

**void display();**

**void main ()**

**{**

**printf("Enter the number of elements in the stack ");**

**scanf("%d",&n);**

**printf("\*Stack operations using array\*");**

**while(choice != 4)**

**{**

**printf("\nChose one from the below options.\n");**

**printf("\n1.Push\n2.Pop\n3.display\n4.Exit");**

**printf("\n Enter your choice \n");**

**scanf("%d",&choice);**

**switch(choice)**

**{**

**case 1:**

**{**

**push();**

**break;**

**}**

**case 2:**

**{**

**pop();**

**break;**

**}**

**case 3:**

**{**

**display();**

**break;**

**}**

**case 4:**

**{**

**printf("Exiting....");**

**break;**

**}**

**default:**

**{**

**printf("Please Enter valid choice ");**

**}**

**};**

**}**

**}**

**void push ()**

**{**

**int val;**

**if (top == n )**

**printf("\n Overflow");**

**else**

**{**

**printf("Enter the value?");**

**scanf("%d",&val);**

**top = top +1;**

**stack[top] = val;**

**}**

**}**

**void pop ()**

**{**

**if(top == -1)**

**printf("Underflow");**

**else**

**top = top -1;**

**}**

**void display()**

**{**

**for (i=top;i>=0;i--)**

**{**

**printf("%d\n",stack[i]);**

**}**

**if(top == -1)**

**{**

**printf("Stack is empty");**

**}**

**}**

**OUTPUT (SCREEN SHOT):**

