**2. Implementation of Stack**

**Aim:**

To implement Stack operations using Array.

**Algorithm:**

**1) Creation of an empty stack**

Step 1.1 - Declare all the functions used in stack implementation.

Step 1.2 - Define a integer variable 'top' and initialize with '-1'. (int top = -1)

Step 1.4 - In main method, display menu with list of operations and make suitable function calls to perform operation selected by the user on the stack.

**2) push(value) - Inserting value into the stack**

Step 2.1 - Check whether stack is FULL. (top >= n-1)

Step 2.2 - If it is FULL, then display "STACK is over flow”.

Step 2.3 - If it is NOT FULL, then increment top value by one (top++) and set stack[top] to value (stack[top] = x).

**3) pop() - Delete a value from the Stack**

Step 3.1 - Check whether stack is EMPTY. (top <= -1)

Step 3.2 - If it is EMPTY, then display "Stack is under flow”.

Step 3.3 - If it is NOT EMPTY, then delete stack[top] and decrement top value by one (top--).

**4) display() - Displays the elements of a Stack**

Step 4.1 - Check whether stack is has elements. (top >=0)

Step 4.2 - If it has elements, then define a variable 'i' and initialize with top. Display stack[i] value and decrement i value by one (i--).

Step 4.2 - If it is EMPTY, then display " The STACK is empty” and terminate the function.

Step 4.4 - Repeat above step until i value becomes '0'.

**Program:**

**// Name:Manojkumar C**

**// Roll no: 727721eucb031**

**#include<stdio.h>**

**int arr[100],choice,n,head,x,i;**

**void push();**

**void pop();**

**void display();**

**int main(){**

**head=-1;**

**printf("\nEnter the Size :\n");**

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

**printf(" 1.PUSH \n 2.POP \n 3.DISPLAY \n 4.EXIT");**

**do{**

**printf("Enter the choice:");**

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

**switch(choice){**

**case 1:**

**push();**

**break;**

**case 2:**

**pop();**

**break;**

**case 3:**

**display();**

**case 4:**

**printf("\nExit Point \n");**

**break;**

**default:**

**printf("\n Enter the choice 1 or 2 or 3 or 4 \n");**

**}**

**}**

**while(choice !=4);**

**return 0;**

**}**

**void push(){**

**if(head>=n-1){**

**printf("\n Stack is Overflowed\n");**

**}**

**else{**

**printf("Enter a Value to push:");**

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

**head++;**

**arr[head]=x;**

**}**

**}**

**void pop(){**

**if(head<=-1){**

**printf("\nStack is Underflow\n");**

**}**

**else{**

**printf("\nThe Popped elem is %d \n",arr[head]);**

**head--;**

**}**

**}**

**void display(){**

**if(head>=0){**

**printf("\n The stack Elements are:\n");**

**for(i=head; i>=0; i--){**

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

**printf("\n The Next choice:");**

**}**

**}**

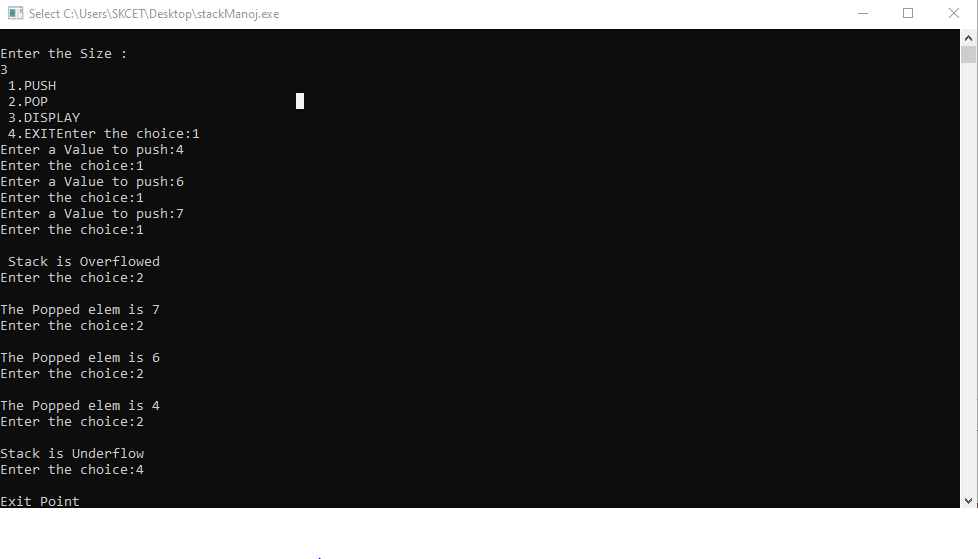
**else{**

**printf("\n The stack is Empty:");**

**}**

**}**

**Output:**



**Result:**

Thus, the implementation of Stack operations using Array is completed.