AIM:- Implementation of Stack Data Structure using array.

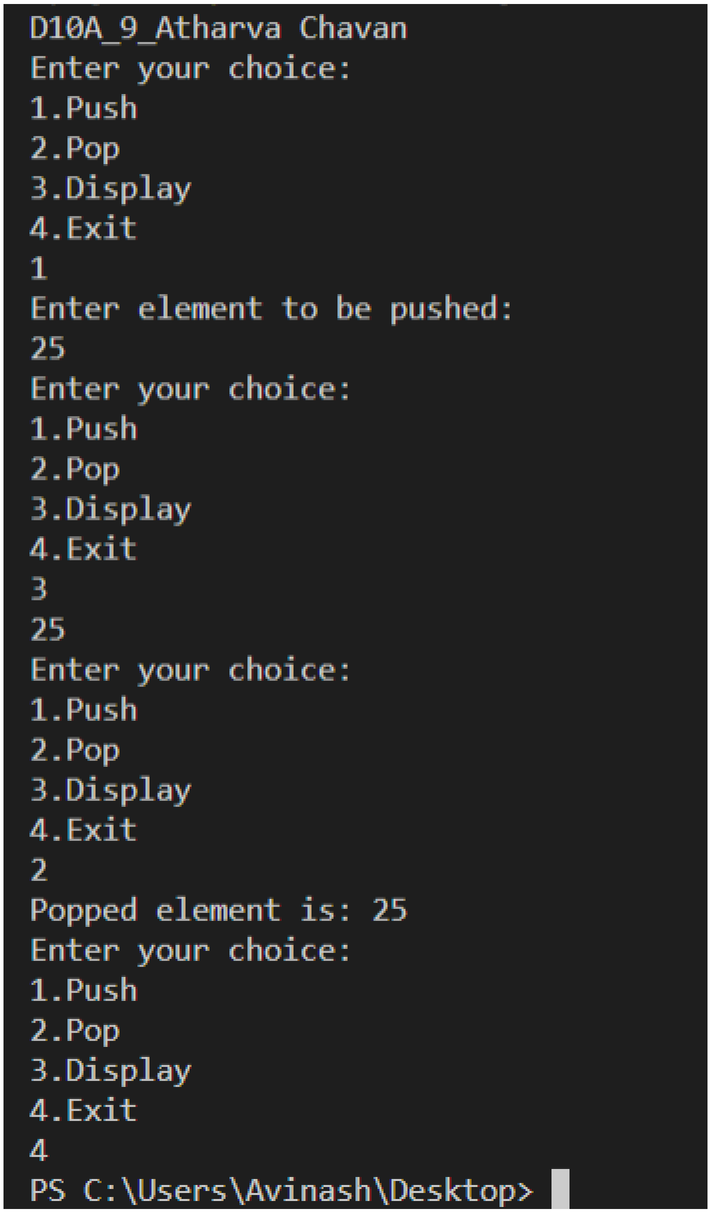
Stack is a linear data structure that follows a particular order in which the operations are performed. The order may be LIFO(Last In First Out). The array based stack implementation is studied as follows:

CODE:-

```
#include<stdio.h>
void push(int a[],int*top,int x)
{
 int n=100;
if(*top==n-1)
printf("Stack is full");
else
 {
 *top=*top+1;
 a[*top]=x;
 }
}
int pop(int a[],int*top)
{
int x;
if (*top<0)
 {
 printf("Stack is empty");
 return 0;
 }
 else
 {
 x=a[*top];
 *top=*top-1;
 return x;
 }
}
```

```
void display (int a[],int top)
{
int i;
for(i=top;i>=0;--i)
printf("%d\n",a[i]);
}
void main()
{
    printf("D10A_9_Atharva Chavan\n");
 int a[100],x,i;
 int top=-1;
 int choice;
 do
 {
 printf("Enter your choice:\n1.Push\n2.Pop\n3.Display\n4.Exit\n");
 scanf("%d",&choice);
 switch(choice)
 {
 case 1:printf("Enter element to be pushed:\n");
 scanf("%d",&x);
 push(a,&top,x);
 break;
 case 2: x=pop(a,&top);
 printf("Popped element is: %d\n",x);
 break;
 case 3:display(a,top);
 break;
 case 4: break;
 }
}
while(choice!=4);
}
```

OUTPUT:-



```
D10A_9_Atharva Chavan
Enter your choice:
1.Push
2.Pop
3.Display
4.Exit
1
Enter element to be pushed:
25
Enter your choice:
1.Push
2.Pop
3.Display
4.Exit
3
25
Enter your choice:
1.Push
2.Pop
3.Display
4.Exit
2
Popped element is: 25
Enter your choice:
1.Push
2.Pop
3.Display
4.Exit
4
PS C:\Users\Avinash\Desktop>
```