PROGRAM-6

OBJECTIVE-Write a program to implement stack operations using array.

CODE –

```
#include<stdio.h>
#include<stdlib.h>
int a[50];
int top=-1;
int push();
int pop();
void display();
void main()
{int choice;
lab:
printf("\nMenu\n1.Push\n2.Pop\n3.Display\n4.Exit\n");
scanf("%d",&choice);
switch(choice)
{case 1:push();
break;
case 2:pop();
break;
case 3:display();
break;
case 4:exit(0);
default:printf("Invalid choice");
break;
}
goto lab;}
int push()
{int ele;
if(top>100)
{printf("Stack Overflow");
  exit(0);}
else
{printf("Enter the element to push:");
   scanf("%d",&ele);
   top++;
   a[top]=ele;}}
int pop()
{if(top==-1)
   printf("Underflow.");
else
{printf("Poped element:%d",a[top]);
```

```
   top--;}}
void display()
{int i;
if(top==-1)
   printf("Underflow.");
else{
printf("Elements:\n");
for(i=0;i<top+1;i++)
printf("%d ",a[i]);}
}
```

OUTPUT-

```
Menu
1.Push
2.Pop
3.Display
4.Exit
1
Enter the element to push:2

Menu
1.Push
2.Pop
3.Display
4.Exit
1
Enter the element to push:3

Menu
1.Push
2.Pop
3.Display
4.Exit
1
Enter the element to push:4

Menu
1.Push
2.Pop
3.Display
4.Exit
2
Poped element:4
Menu
1.Push
2.Pop
3.Display
4.Exit
3
Elements:
2 3
```

PROGRAM-7

OBJECTIVE-Write a program to implement stack using linked list.

CODE –

```
#include <stdio.h>
#include <stdlib.h>
void push();
void pop();
void display();
struct node
{int info;
   struct node *next;
   struct node *prev;
}*head=NULL, *tail=NULL;
void main()
{int ch=0;
   while (1)
   {printf("\nEnter \n1 for Push \n2 for Pop \n3 for Display \n4 for Exit \nEnter Your
Choice...");
      scanf("%d", &ch);
      switch(ch)
      {case 1: push();
               break;
         case 2:  pop();
               break;
         case 3: display();
               break;
         case 4: exit(0);}}}
void push()
{struct node *temp, *ptr;
   ptr=tail;
   int data;
   temp=(struct node*)malloc(sizeof(struct node));
   printf("Enter info\n");
   scanf("%d",&data);
   temp->info=data;
   temp->prev=ptr;
   tail=temp;
   if (ptr!=NULL)
      ptr->next=temp;}
void pop()
{struct node *ptr;
   ptr=tail;
   if (ptr==NULL)
      printf("Stack is Empty\n");
```

```
    else
    {tail=ptr->prev;
        printf("Element %d deleted Successfully\n", ptr->info);
        free(ptr);}}
void display()
{struct node *ptr;
    ptr=tail;
    while (1)
    {printf("%d\t",ptr->info);
        ptr=ptr->prev;
        if (ptr==NULL)
            break;}
    printf("\n");}
```

OUTPUT-

```
Enter
1 for Push
2 for Pop
3 for Display
4 for Exit
Enter Your Choice...1
Enter info
2

Enter
1 for Push
2 for Pop
3 for Display
4 for Exit
Enter Your Choice...1
Enter info
3

Enter
1 for Push
2 for Pop
3 for Display
4 for Exit
Enter Your Choice...1
Enter info
4

Enter
1 for Push
2 for Pop
3 for Display
4 for Exit
Enter Your Choice...2
Element 4 deleted Successfully

Enter
1 for Push
2 for Pop
3 for Display
4 for Exit
Enter Your Choice...3
3       2
```