EXPERIMENT NO: 1

STACK IMPLEMENTATION USING ARRAY

AIM:

To develop a c program to implement a stack using array.

DATA STRUCTURE:

Arrays

ALGORITHM:

Global variables: Stack S[50], top=-1

Main function:

1. Start
2. Read the size of stack as n
3. Display the operation to be performed

   1. PUSH 2.POP 3.DISPLAY 4.EXIT
4. Read the choice of operation as ch
5. If ch=1 call function PUSH()
6. If ch=2

   1. call function POP() and store the return in item
   2. Display the item deleted as item
7. If ch=3, call function DISPLAY()
8. If ch=4 then exit from program
9. Stop

PUSH(size n)

1. Start
2. If top=n=1
3. Display stack overflow: Insertion not possible
4. Exit
5. Else
6. Read the elements to be inserted as item
7. top=top+1
8. S[top]=item
9. End if
10. Return

POP()

1. Start
2. If top= -1
3. Display stack empty: Deletion not possible
4. Exit
5. Else
6. item=s[top]
7. top= top-1
8. End if
9. Return item

Function Display()

1. Start

2. If top= -1

3. print “stack empty”

4. Exit

5. Else

6. i=top

7. while(i>=0)

8. print s[i]

9. i=i+1

10. End while

11. End if

12. Return

PROGRAM:

```
/* array implementation of stack*/


#include<stdio.h>
#include<stdlib.h>
int S[50], top=-1;
void push();
void pop();
void display();
main()
{
        int n,ch;
        printf("\n Enter the size of stack:");
        scanf("%d",&n);
        do
        {
                Printf("\n\t MENU");
                Printf("\n\t 1.PUSH\n\t 2.POP\n\t 3.DISPLAY\n\t 4.EXIT\n");
                Printf("Enter the choice:");
                Scanf("%d",&ch);
                Switch (ch)
                {
                        Case 1:
                                Push(n);
```

```
                    break;
                case 2:
                    pop();
                    break;
                case 3:
                    display();
                    break;
                case 4:
                    break;
                default;
                printf("\n\t invalid choice:");
            }
        } while (ch!==4);
}
Void push(int n)
{
        int item;
        if (top==n-1)
        {
            Printf("\n stack is overflow: insertion not possible");
        }
        else
        {
            printf("\n Enter the element to be inserted:");
```

```
            Scanf("%d",&item);
            top=top+1;
            S[top]=item;
        }
        return;
}
Void pop()
{
        int item;
        if(top==-1)
        {
            Printf("stack empty: deletion not possible");
        }
        else
        {
            item=s[top];
            top=top-1;
            printf("deleted item is %d",item);
        }
}
void display()
{
        int i;
        if(top==-1)
```

```
        {
                Printf("stack empty");
        }
        else
        {
                i=top;
                while(i>=0)
                {
                        Printf("%d",[i]);
                        i=i-1;
                }
        }
        return;
}
```

OUTPUT:

```
Enter the size of stack:10
        MENU
        1.PUSH
        2.POP
        3.DISPLAY
        4.EXIT
```

Enter the choice:1

Enter the element to be inserted:4

MENU

1.PUSH

2.POP

3.DISPLAY

4.EXIT

Enter the choice:2

Deleted item is 4

MENU

1.PUSH

2.POP

3.DISPLAY

4.EXIT

Enter the choice:3

Stack empty

MENU

1.PUSH

2.POP

3.DISPLAY

4.EXIT

Enter the choice:2

Stack empty:deletion not possible

MENU

1.PUSH

2.POP

3.DISPLAY

4.EXIT

Enter the choice:4

RESULT:

The program is executed successfully and output is verified.