ACADEMY OF TECHNOLOGY

NAME : PRIYADARSHAN GHOSH

COLLEGE ROLL NO: 72

UNIVERSITY ROLL NO: 16900319072

DEPARTMENT: ECE-1(Y)

SEMESTER: 3rd

PAPER CODE : ES-CS391

➢ Laboratory *Assignment #5*

Write menu driven program to perform following operations using stack:

A.CONVERT INFIX EXPRESSION TO POSTFIX EXPRESSION.

Ans:

```
#include <stdio.h>
#define MAX 100
long int stack[MAX];
int top=-1;
void push(char ch)
{
	if (top == MAX-1)
	{
		printf("\nStack Overflow");
		return;
	}
	stack[++top]=ch;
}
int pop()
{
	if (top==-1)
	{
		printf("\nStack Underflow");
		return -1;
	}
```

```
        return (stack[top--]);
}
int preced(char ch)
{
        if (ch == '^')
                return 5;
        else if (ch == '/')
                return 4;
        else if (ch == '*')
                return 3;
        else if (ch == '+')
                return 2;
        else if (ch == '-')
                return 1;
        else
                return 0;
}
void infixToPostfix(char infix[])
{
        char postfix[50], x, ch;
        int i=0, j=0;
        while(infix[i]!='\0')
        {
                ch = infix[i];
                switch (ch)
                {
                        case '(':   push (ch);
```

```
                        break;
                case ')':   while ((ch=pop()) != '(')
                                postfix[j++] = ch;
                        break;
                case '^':
                case '/':
                case '*':
                case '+':
                case '-':
                        while (preced(ch) < preced(stack[top]))
                                postfix[j++] = pop();
                        push(ch);
                        break;
                default:   postfix[j++] = ch;
            }
            i++;
        }
        postfix[j] = '\0';
        printf("\nPostfix expression: ");
        puts(postfix);
}
int main()
{
        char infix[50];
        printf("Enter infix expression: ");
        gets(infix);
        infixToPostfix(infix);
```

```
}
```

OUTPUT =>

```
Enter infix expression: a+(b^c-d/e)+f*g

Postfix expression: abc^de/-fg

...Program finished with exit code 0
Press ENTER to exit console.
```

B. EVALUATE POSTFIX EXPRESSION.

Ans:

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#define max 20
struct stack
{
        float value[max];
        int top;
};
struct stack stk;


int isFull()
{
        if(stk.top==max-1)
```

```
        return 1;

    return 0;
}

int isEmpty()
{
    if(stk.top==-1)
        return 1;

    return 0;
}

int push(float key)
{
    if(isFull())
        return -1;
    stk.value[++stk.top]=key;
    return stk.top;
}


float pop()
{
    if(isEmpty())
        return -1;
    return stk.value[stk.top--];
```

```
}

float postfixEvaluation(char postfix[], float value[])
{
  int i=0;
  char ch;
  float op1, op2, s;
  while(postfix[i]!=NULL)
  {
    ch = postfix[i];
    if((ch>=65 && ch<=90) || (ch>=97 && ch<=122))
                {
                push(value[i]);
       }
    else
    {
      op2 = pop();
      op1 = pop();
      switch(ch) {
        case '+': push(op1+op2);
                  break;
        case '-': push(op1-op2);
                  break;
                case '*': push(op1*op2);
                  break;
                case '/': push(op1/op2);
```

```
                break;
              case '^': push(pow(op1,op2));
                break;
      }
    }
    i++;
  }

  return pop();
}

int main()
{
  char postfix[20], ch;
  float value[20], result;
  int i=0;
  printf("ENTER A VALID POSTFIX EXPRESSION = ");
  gets(postfix);

  while(postfix[i]!='\0')
  {
    ch = postfix[i];
    if((ch>=65 && ch<=90) || (ch>=97 && ch<=122))
        {
       printf("\nENTER THE VALUE OF %c =",ch);
       scanf("%f",&value[i]);
    }
```

```
    i++;
  }
  result = postfixEvaluation(postfix, value);
  printf("\nRESULT = %f", result);
  return 0;
}
```

OUTPUT =>

```
ENTER A VALID POSTFIX EXPRESSION = abc^*

ENTER THE VALUE OF a =1

ENTER THE VALUE OF b =2

ENTER THE VALUE OF c =3

RESULT = 8.000000

...Program finished with exit code 0
Press ENTER to exit console.
```