Lab 2 :-

Date __/__/__
Page______

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX 100

char stack[MAX];
int top = -1;

void push (char);
char pop();
int precedence (char);
void infixtoPostfix (char infix[], char postfix[]);

void push (char item)
{
    if (top == MAX-1)
     {
       printf ("Stack Overflow\n");
     }
    else
     {
        top++;
       stack[top] = item;
     }
char pop()
{
   if (top == -1)
   {
      printf (" Stack underflow.\n");
   }
   else
    {
      char popped = stack[top];
       top--;
    }  return popped;
```

```
int precedence (char symbol)
{
    if (symbol = = '^')
        return 3;
    else if (symbol = = '*' || symbol = = '/')
        return 2;
    else if ( symbol = = '+' || symbol = = '-')
        return 1;
    else
        return -1;
};

void infixToPostfix (char infix [], char postfix [])
{
    int i = 0,  j = 0;
    char symbol, temp;
    push ('#');

    while ((symbol = infix [i ++]) != '\0')
    {
        if (symbol = = '(')
        {
            push (symbol);
        }
        else if (isalnum (symbol))
        {
            postfix [j++] = symbol;
        }
        else if (symbol == ')')
        {
            while (stack [top] != '(')
            {
                postfix [j++] = pop ();
            }
temp= pop();
}
 else
  {
     while ( precedence (Stack[top]) >= precedence (symbol))
     {
         postfix[j++] = pop();
     }
     push (symbol);
  }
}
while (stack [top] != '#')
{
     postfix [j++] = pop();
}
postfix [j]= '\0';
}

int main()
{
   char infix [MAX], postfix [MAX];
   printf ("Enter a valid parenthesized infix expression: \n");
   scanf (" %s", infix);
   infix To Postfix (infix, postfix);
   printf (" The postfix expression is : %s \n", postfix);
    return 0;
}
```

[illegible]

~~Postfix evaluation~~

<u>Output</u> :-

```
Enter the choice:
1
Enter the element to be added in the stack:
5
Operations on the stack:
1. push the element
2. pop the element
3. Show
4. End
Enter the choice:
1
Enter the element to be added in the stack:
6
Operations on the stack:
1. push the element
2. pop the element
3. Show
4. end
Enter the choice:
3
Elements in the stack are:
6
5
```

postfix evaluation :-

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>

#define max 20

int stack [20];
int top=-1;

void push (int a) {
        stack [top ++]=a;
    }

  int pop () {
        return stack [top --];
    }

void main() {
        char postfix [max]= "12*34*+5-";
        char postfix [max];
        printf ("Enter the postfix expression:");
        scanf ("%s", postfix);

      int result= 0,a, b;
      for (int i=0; i <strlen(postfix); i++) {
            if(isalnum (postfix [i]))
          push (postfix [i]-'0');
else {
        b = pop();
        a = pop();
        switch (postfix [i])
        {
        Case '+';
           push (a+b);
           break;
        Case '-';
           push (a-b);
           break;
        Case '*';
           push (a*b);
           break;
        Case '/';
           push (a/b);
           break;
        Case '^';
           push (a^b);
           break;
        }
    }
  }
        result = pop();
        printf ("%.s = %d", postfix, result);

 }
```

Output :-

```
Enter the postfix expression : 23*31*+4
23*31*+5- = 4
```