**PROGRAMS: Postfix expression evaluation**

**Input:**

#include<stdio.h>

#include<string.h>

#include<math.h>

#include<stdlib.h>

#define BLANK' '

#define TAB '\t'

#define MAX 50

void push(long int symbol);

long int pop();

long int eval\_post();

int isEmpty();

char infix[MAX], postfix[MAX];

long int stack[MAX];

int top;

 main()

 {

    long int value;

    top = -1;

    printf("Enter postfix: ");

    gets(postfix);

    printf("Postfix: %s\n", postfix);

    value = eval\_post();

    printf("Value of expression: %ld\n", value);

 }

 void push(long int symbol)

 {

    if (top>MAX)

    {

        printf("Stack overflow\n");

        exit(1);

    }

    stack[++top] = symbol;

 }

 long int pop()

 {

    if(isEmpty())

    {

        printf("Stack underflow\n");

        exit(1);

    }

    return (stack[top--]);

 }

 int isEmpty()

 {

    if(top==-1)

        return 1;

    else

        return 0;

 }

 long int eval\_post()

 {

    long int a, b, temp, result;

    unsigned int i;

    for(i=0; i<strlen(postfix); i++)

    {

        if(postfix[i]<='9' && postfix[i]>='0')

            push(postfix[i]-'0');

        else

        {

            a = pop();

            b = pop();

            switch(postfix[i])

            {

            case '+':

                temp = b+a; break;

            case '-':

                temp = b-a; break;

            case '\*':

                temp = b\*a; break;

            case '/':

                temp = b/a; break;

            case '%':

                temp = b%a; break;

            case '^':

                temp = pow(b,a);

            }

            push(temp);

        }

    }

 result = pop();

 return result;

 }

**Output:**

****