**PROGRAMS: Infix to postfix expression**

**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();

void infix\_to\_postfix();

int priority(char symbol);

int isEmpty();

int white\_space(char symbol);

char infix[MAX], postfix[MAX];

long int stack[MAX];

int top;

 main()

 {

    long int value;

    top = -1;

    printf("Enter infix: ");

    gets(infix);

    infix\_to\_postfix();

    printf("Postfix: %s\n", postfix);

 }

 void infix\_to\_postfix()

 {

    unsigned int i, p = 0;

    char next;

    char symbol;

    for(i=0; i<strlen(infix); i++)

    {

        symbol = infix[i];

        if(!white\_space(symbol))

        {

            switch(symbol)

            {

                case '(':

                push(symbol);

Expt no:7

                break;

                case ')':

                while((next=pop())!='(')

                    postfix[p++] = next;

                break;

                case '+':

                case '-':

                case '\*':

                case '/':

                case '%':

                case '^':

                    while(!isEmpty()&&priority(stack[top])>=priority(symbol))

                        postfix[p++] = pop();

                    push(symbol);

                    break;

                default:

                    postfix[p++] = symbol;

            }

        }

    }

    while(!isEmpty())

        postfix[p++] = pop();

    postfix[p] = '\0';

 }

 int priority(char symbol)

 {

    switch(symbol)

    {

        case '(':

            return 0;

        case '+':

        case '-':

            return 1;

        case '\*':

        case '/':

        case '%':

            return 2;

        case '^':

            return 3;

        default:

            return 0;

    }

 }

 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;

 }

 int white\_space(char symbol)

 {

    if(symbol==BLANK || symbol==TAB)

        return 1;

    else

    return 0;

Expt no:7

 }

**Output:**

