**DSA Assignment: 2**

**Exp 2: Conversion of Infix Expression to Postfix Expression using Stack**

Shashwat Tripathi

D10A Roll No: 60

**AIM:** In this experiment, we will learn about how to convert a given infix expression into postfix expression using push and pop operations of Stack.

**CODE:**

#include <stdio.h>

#include <ctype.h>

#include <conio.h>

typedef struct

{

    char a[100];

    int top;

} stack;

void push(stack \*s, char x);

int pop(stack \*s);

int precede(char c1, char c2);

void conversion(char in[], char post[]);

void main()

{

    printf("Shashwat Tripathi\_D10A\_60\n");

    char infix[100], postfix[100];

    printf("Enter an Infix Expression:\n");

    gets(infix);

    (conversion(infix, postfix));

    printf("Corresponding Postfix Expression is:\n");

    puts(postfix);

}

void push(stack \*s, char x)

{

    if (s->top == 99)

        printf("Stack is full.");

    else

        s->a[++s->top] = x;

}

int pop(stack \*s)

{

    char x;

    if (s->top < 0)

        printf("Stack is empty");

    else

    {

        x = s->a[s->top--];

        return x;

    }

}

int precede(char c1, char c2)

{

    switch (c1)

    {

    case '-':

    case '+':

        if (c2 == '-' || c2 == '+' || c2 == ')')

            return 1;

        else

            return 0;

    case '/':

    case '\*':

        if (c2 == '(')

            return 0;

        else

            return 1;

    case '(':

        return 0;

    }

}

void conversion(char in[], char post[])

{

    stack s;

    int i, j = 0;

    char top;

    s.top = -1;

    for (i = 0; in[i] != '\0'; ++i)

    {

        if (isalpha(in[i]))

            post[j++] = in[i];

        else

        {

            while (s.top >= 0 && precede(s.a[top], in[i]))

            {

                post[j++] = pop(&s);

            }

            if (in[i] == ')')

                s.top--;

            else

                push(&s, in[i]);

        }

    }

    while (s.top >= 0)

    {

        post[j++] = pop(&s);

        post[j] = '\0';

    }

}

**OUPUT:**

