**Name: Adithya M SRN: PES1UG20CS621 Sec: K**

Program 1:

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

#define MAX 50

void push(long int ch);

long int pop();

void toPostfix();

long int evalPost();

int priority(char ch);

int isEmpty();

int isWhiteSpace(char);

char infix[MAX], postfix[MAX];

long int stack[MAX];

int top = -1;

int main()

{

    long int value;

    printf("Enter infix\n");

    scanf("%s", infix);

    toPostfix();

    printf("\nPostfix %s\n", postfix);

    value = evalPost();

    printf("\nValue %ld\n", value);

    return 0;

}

void push(long int ch)

{

    if (top > MAX)

    {

        printf("Stack overflow\n");

        exit(1);

    }

    stack[++top] = ch;

}

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 isWhiteSpace(char ch)

{

    if (ch == ' ' || ch == '\t')

        return 1;

    else

        return 0;

}

void toPostfix()

{

    int i, p = 0;

    char next;

    char ch;

    for (i = 0; i < strlen(infix); i++)

    {

        ch = infix[i];

        if (!isWhiteSpace(ch))

        {

            switch (ch)

            {

            case '(':

                push(ch);

                break;

            case ')':

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

                    postfix[p++] = next;

                break;

            case '+':

            case '-':

            case '\*':

            case '/':

            case '%':

            case '^':

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

                    postfix[p++] = pop();

                push(ch);

                break;

            default:

                postfix[p++] = ch;

            }

        }

    }

    while (!isEmpty())

        postfix[p++] = pop();

    postfix[p] = '\0';

}

int priority(char ch)

{

    switch (ch)

    {

    case '(':

        return 0;

    case '+':

        return 2;

    case '-':

        return 1;

    case '\*':

        return 2;

    case '/':

        return 2;

    case '%':

        return 2;

    default:

        return 0;

    }

}

long int evalPost()

{

    long int a, b, temp;

    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;

            }

            push(temp);

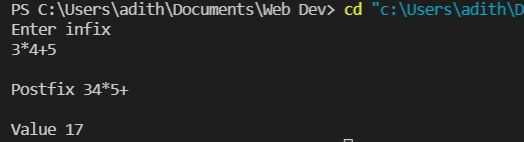
        }

    }

    return pop();

}

Output:



**Program 2:**

#include <stdio.h>

#include <string.h>

typedef struct st

{

    char words[100];

    int top;

} st;

void push(st \*s, char a)

{

    s->top++;

    s->words[s->top] = a;

}

char pop(st \*s)

{

    char temp;

    temp = s->words[s->top];

    s->top--;

    return temp;

}

void display(st s)

{

    while (s.top != -1)

    {

        printf("%c", s.words[s.top]);

        s.top--;

    }

}

int main()

{

    st s;

    s.top = -1;

    int check[27] = {0};

    char a[100];

    printf("Enter a string: ");

    scanf("%[^\n]s", a);

    printf("\n");

    for (int i = 0; a[i] != '\0'; i++)

    {

        if (a[i] == ' ')

            push(&s, a[i]);

        else if (check[a[i] - 'a'] == 0)

        {

            check[a[i] - 'a']++;

            push(&s, a[i]);

        }

    }

    st s1;

    s1.top;

    while (s.top != -1)

    {

        push(&s1, pop(&s));

    }

    display(s1);

}

Output:

