**Name:** Kunal Pal

**USN:** 22MCAR0070

**Problem**: Write a C program that uses stack operations to convert a given infix expression into

its postfix equivalent.

**Code:**

#include <stdio.h>

#include <ctype.h>

//Stack size

char stack[100];

int top = -1;

//pushing element

void push\_ele(char x)

{

    stack[++top] = x;

}

//Popping element

char pop\_ele()

{

    if (top == -1)

        return -1;

    else

        return stack[top--];

}

//A priority check is done, when we encounter a ‘)’ in the expression

int priority\_check(char x)

{

    if (x == '(')

        return 0;

    if (x == '+' || x == '-')

        return 1;

    if (x == '\*' || x == '/')

        return 2;

    return 0;

}

int main()

{

    char exp[100];

    char \*e, x;

    printf("Infix Expression : ");

    scanf("%s", exp);

    printf("\n");

    e = exp;

    printf("Postfix expression : ");

    while (\*e != '\0')

    {

//checking for alphanumeric characters

        if (isalnum(\*e))

            printf("%c ", \*e);

        else if (\*e == '(')

            push\_ele(\*e);

        else if (\*e == ')')

        {

            while ((x = pop\_ele()) != '(')

                printf("%c ", x);

        }

        else

        {

            while (priority\_check(stack[top]) >= priority\_check(\*e))

                printf("%c ", pop\_ele());

            push\_ele(\*e);

        }

        e++;

}

//after reaching end of stack, popping every element stored in stack

    while (top != -1)

    {

        printf("%c ", pop\_ele());

    }

    return 0;

}

**Output:**

