**Exp – 2 – Code and Output**

**Question – 1: -**

**Code: -**

#include<stdio.h>

#include<conio.h>

#include<ctype.h>

char stack[100];

int tos = -1;

void push(char x)

{

    stack[++tos] = x;

}

char pop()

{

    if(tos == -1)

    return -1;

    else

    return stack[tos--];

}

int priority(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("Enter the expression : ");

    scanf("%s",exp);

    printf("\n");

    e = exp;

    while(\*e != '\0')

    {

        if(isalnum(\*e))

        printf("%c ",\*e);

        else if(\*e == '(')

        push(\*e);

        else if(\*e == ')')

        {

            while((x = pop()) != '(')

            printf("%c ", x);

        }

        else

        {

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

            printf("%c ",pop());

            push(\*e);

        }

        e++;

    }

    while(tos != -1)

    {

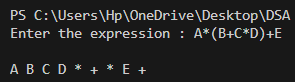
        printf("%c ",pop());

    }

    return 0;

}

**Output: -**

****

**Question – 2: -**

**Code: -**

#include <stdio.h>

#include <stdlib.h>

#include <ctype.h>

#define MAX 100

int stack[MAX];

int tos = -1;

void push(int item)

{

    if (tos >= MAX - 1)

    {

        printf("Stack overflow\n");

        return;

    }

    else

    {

        stack[++tos] = item;

    }

}

int pop()

{

    if (tos < 0)

    {

        printf("Stack underflow\n");

        return -1;

    }

    else

    {

        return stack[tos--];

    }

}

int evaluatePostfix(char \*postfix)

{

    int i = 0;

    while (postfix[i])

    {

        if (isdigit(postfix[i]))

        {

            push(postfix[i] - '0');

        }

        else

        {

            int val1 = pop();

            int val2 = pop();

            switch (postfix[i]) {

                case '+': push(val2 + val1); break;

                case '-': push(val2 - val1); break;

                case '\*': push(val2 \* val1); break;

                case '/': push(val2 / val1); break;

            }

        }

        i++;

    }

    return pop();

}

int main()

{

    char postfix[100];

    printf("Enter postfix expression: ");

    scanf("%s", postfix);

    printf("Result of postfix evaluation: %d\n", evaluatePostfix(postfix));

    return 0;

}

**Output: -**

****