Question
Design and implement an algorithm for conversion of an expression from one form to another. Demonstrate its working with suitable inputs

Aim: To convert given infix expression to postfix expression

Algorithm:

- Take a pointer that points towards the first character in the string
- Check whether the given character is alphanumeric
- If it is, then print it
- Else if it is '(' then push it into the stack
- Else if it's ')' then keep popping elements out of the stack and print them until the popped element is '('
- Else if the character is an operand then we compare it to the current element in the stack and check its priority, if the priority of the operand is greater than the current operand in the stack, then push it. Else keep popping the operands until the priority of the operand is lesser than the

Program

```c
#include <stdio.h>
#include <stdlib.h>
char stack[40];
int top =-1;
void push(char x)     //push function
{
    stack[++top]=x;
}
char pop()              //pop function
{
    if (top==-1)
        return -1;
    else
    {
        return stack[top--];
    }
}

int priority(char x) //checks priority of operations
{
    if(x =='(')
        return 0;
    if(x=='+'||x=='-')
```

```
        return 1;
    if(x=='*'||x=='/')
        return 2;
}
int main() //driver code
{
    char exp[40];
    char *e, x;
    printf("Enter the expression: \n");
    scanf("%s", exp);
    e = exp;
    while (*e!='\0')
    {
        if(isalnum(*e)) //checks if the character is alphanumeric
            printf("%c", *e);
        else if (*e=='(')
            push (*e);
        else if (*e==')')
        {
            while((x=pop())
            !='(')
                printf("%c", x);
        }
        else
        {
            while (priority(stack[top])>=priority(*e))
                printf("%c",pop());
            push(*e);
        }
        e++;
    }
    while (top!=-1)
    {
        printf("%c",pop());
    }
}
```

Output