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 prefix expression

Algorithm:

- First reverse the given expression
- Now reverse the brackets so they align with each other
- Convert this expression into postfix format
- Reverse the obtained postfix string to obtain prefix expression

Program

```c
#include <stdio.h>
#include <ctype.h>
#include<string.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;
}
```

```
char infixtopost(char exp[], char temp[]) //converts given infix
expression to postfix expression
{
    char *a;
    char *e, x;
    e = exp;
    a= temp;
    while (*e!='\0')
    {
        if(isalnum(*e)){
            *a=*e;
            a++;
        }
        else if (*e=='(')
            push (*e);
        else if (*e==')')
        {
            while((x=pop())!='('){
                *a=x;
                a++;
            }
        }
        else
        {
            while (priority(stack[top])>=priority(*e)){
                *a=pop();
                a++;
                }
            push(*e);

        }
        e++;
    }
    while (top!=-1)
    {
        *a=pop();
        a++;
    }
}
void reverse(char arr[]){ // to reverse the given expression
```

```
    int i,j;
    char temp[100];
    for (i=strlen(arr)-1,j=0;i>=0;--i,++j) {
        temp[j]=arr[i];
    }
    temp[j]='\0';
    strcpy(arr,temp);
}
void revbrackets(char exp[]){ //To reverse the brackets after reversing
the string
    for(int i=0;i<strlen(exp)-1;i++){
        if(exp[i]=='('
            exp[i]==')';
        else if(exp[i]==')')
            exp[i]=='(';
    }
}
int main(){ //driver code
    char exp[40];
    printf("enter the exp ");
    scanf("%s",exp);
    reverse(exp); //reverses the expression
    revbrackets(exp);//reverses the brackets
    char temp[40];
    infixtopost(exp, temp);//converts the exp to postfix form
    reverse(temp);// reverses the postfix string to get prefix expression
    printf("%s", temp);
}
```

Output

```
enter the exp A+B+C
+A+BC
PS E:\code\prefix> cd "e:\code\prefix\" ; if ($?) { gcc prefix.c -o prefix } ; if ($?) { .\prefix }
enter the exp A*B+C
+*ABC
```