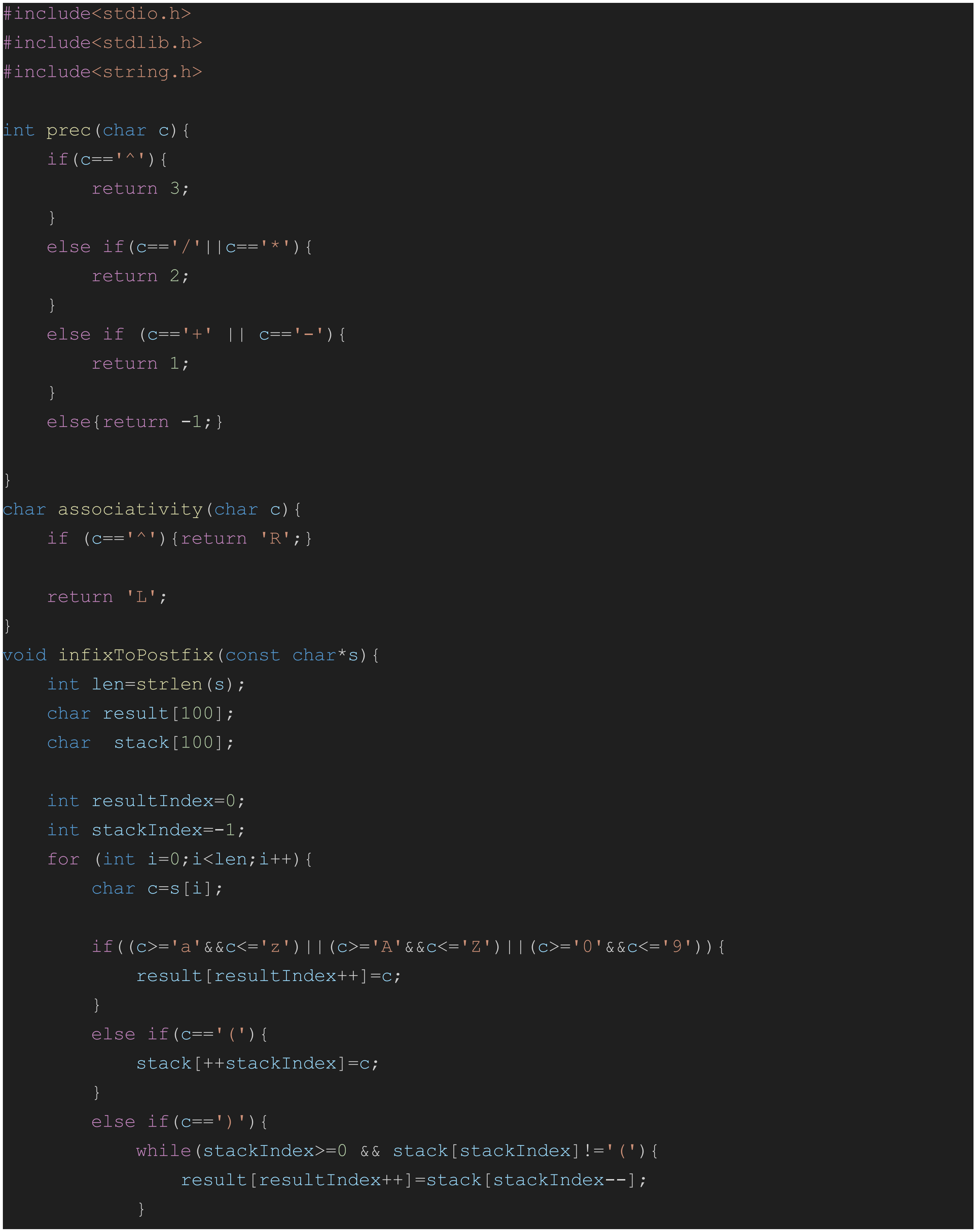
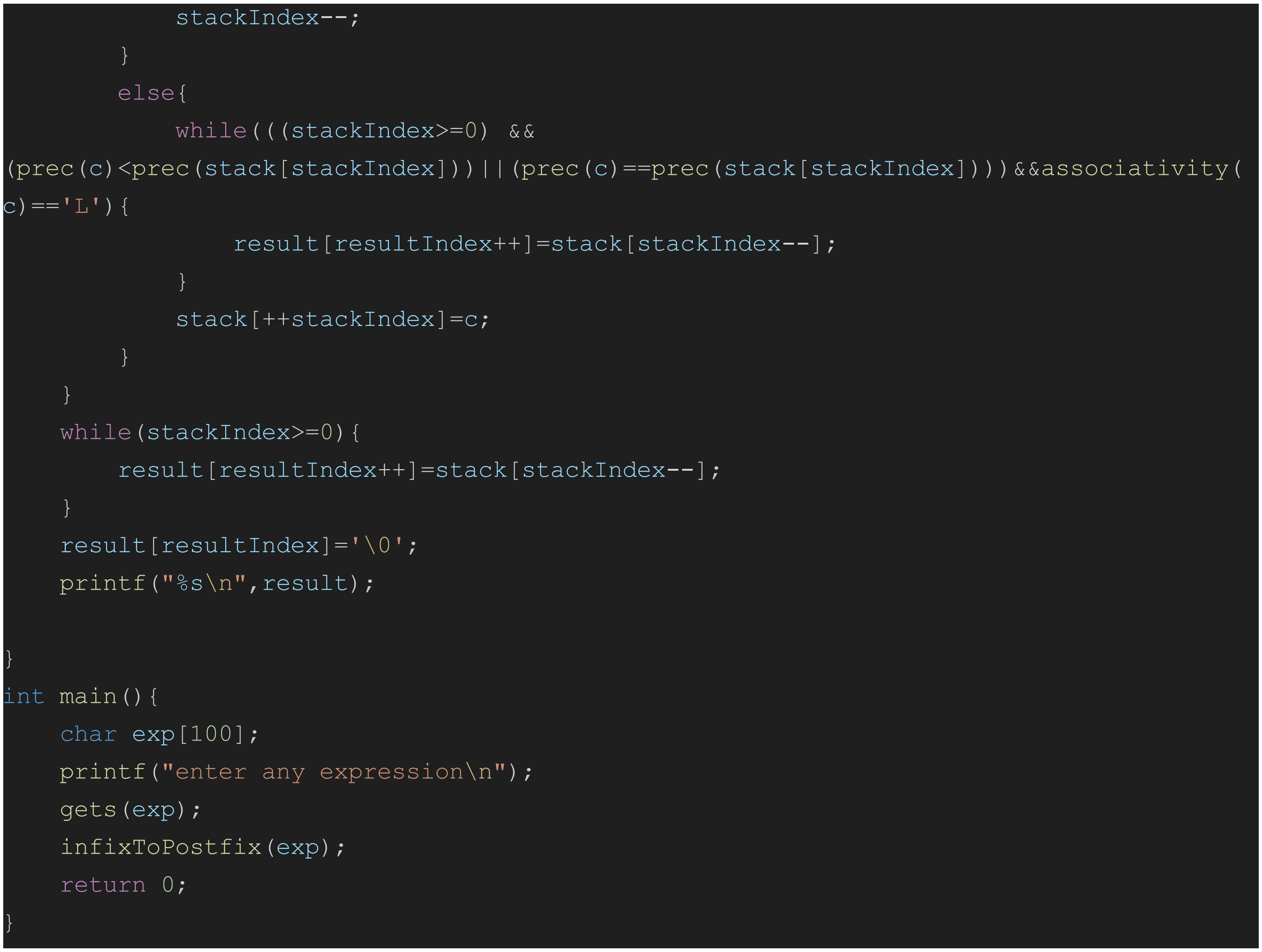
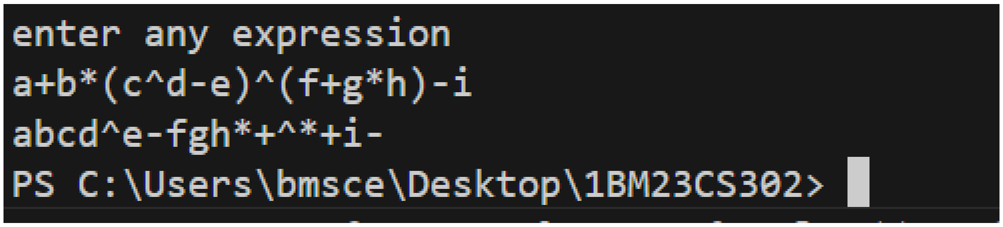


```c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

int prec(char c){
    if(c=='^'){
        return 3;
    }
    else if(c=='/'||c=='*'){
        return 2;
    }
    else if (c=='+' || c=='-'){
        return 1;
    }
    else{return -1;}

}
char associativity(char c){
    if (c=='^'){return 'R';}

    return 'L';
}
void infixToPostfix(const char*s){
    int len=strlen(s);
    char result[100];
    char  stack[100];

    int resultIndex=0;
    int stackIndex=-1;
    for (int i=0;i<len;i++){
        char c=s[i];

        if((c>='a'&&c<='z')||(c>='A'&&c<='Z')||(c>='0'&&c<='9')){
            result[resultIndex++]=c;
        }
        else if(c=='('){
            stack[++stackIndex]=c;
        }
        else if(c==')'){
            while(stackIndex>=0 && stack[stackIndex]!='('){
                result[resultIndex++]=stack[stackIndex--];
            }
```



```
            stackIndex--;
        }
        else{
            while(((stackIndex>=0) &&
(prec(c)<prec(stack[stackIndex]))||(prec(c)==prec(stack[stackIndex])))&&associativity(
c)=='L'){
                result[resultIndex++]=stack[stackIndex--];
            }
            stack[++stackIndex]=c;
        }
    }
    while(stackIndex>=0){
        result[resultIndex++]=stack[stackIndex--];
    }
    result[resultIndex]='\0';
    printf("%s\n",result);

}
int main(){
    char exp[100];
    printf("enter any expression\n");
    gets(exp);
    infixToPostfix(exp);
    return 0;
}
```



```
enter any expression
a+b*(c^d-e)^(f+g*h)-i
abcd^e-fgh*+^*+i-
PS C:\Users\bmsce\Desktop\1BM23CS302>
```