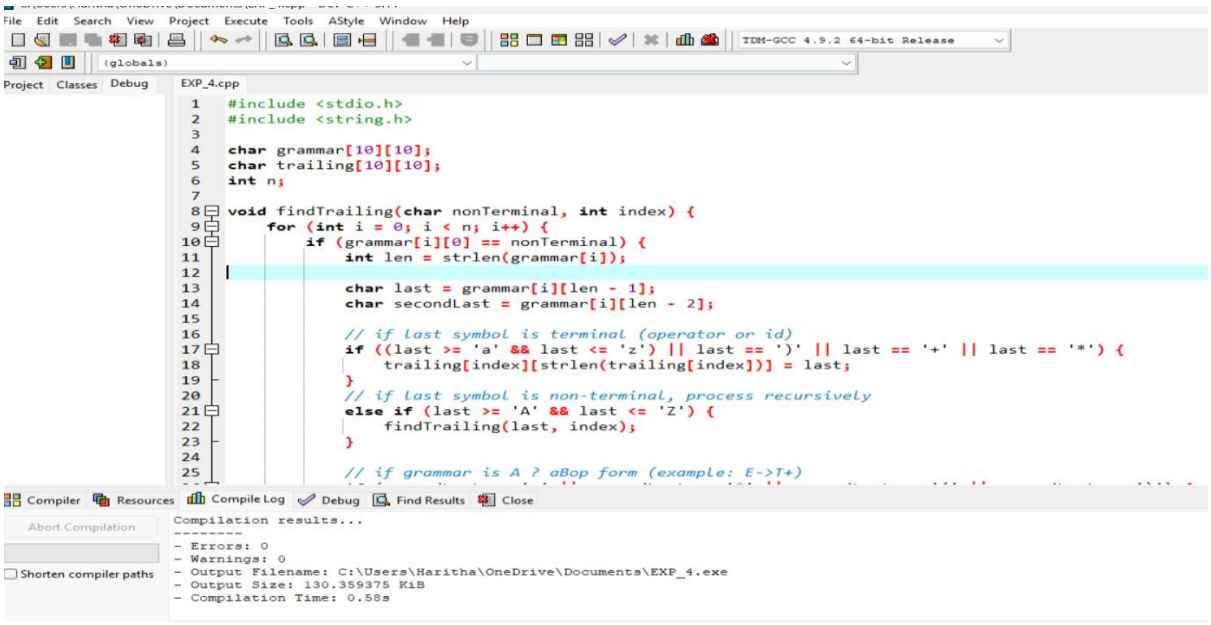
EXPERIMENT-20

AIM:

Write a C program to compute TRAILING( ) – operator precedence parser for the given grammar

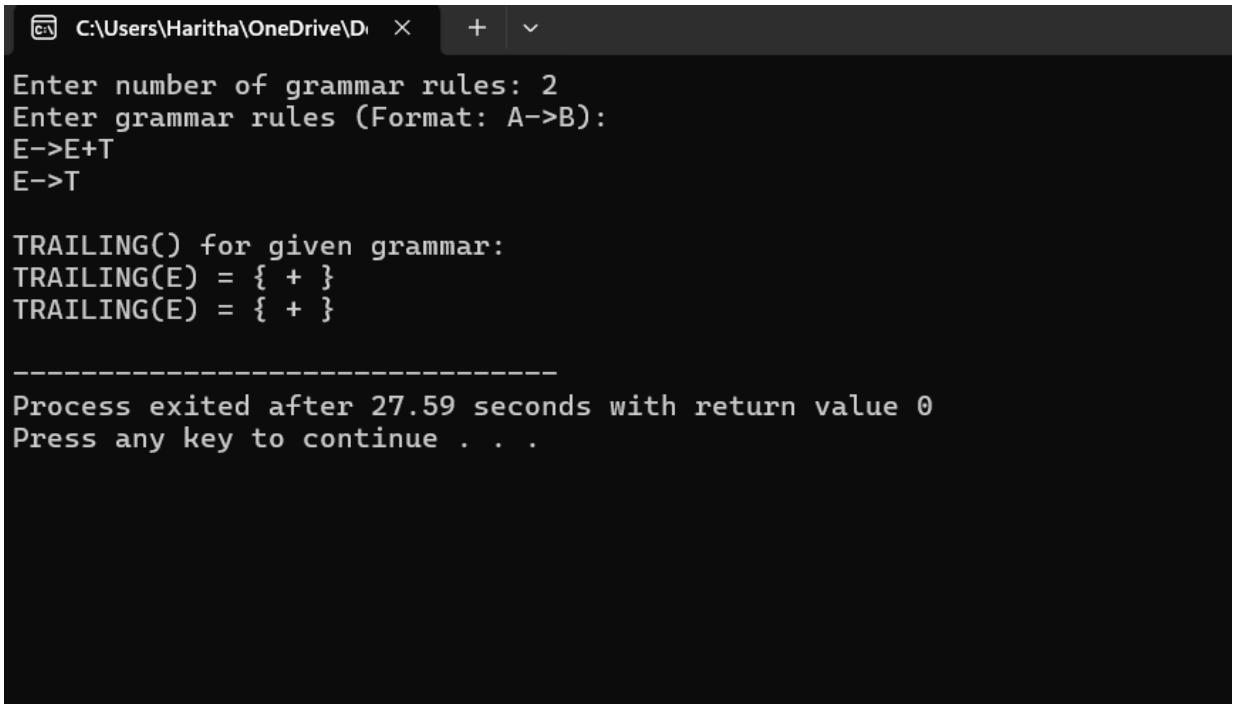
PROGRAM:



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

char grammar[10][10];
char trailing[10][10];
int n;

void findTrailing(char nonTerminal, int index) {
    for (int i = 0; i < n; i++) {
        if (grammar[i][0] == nonTerminal) {
            int len = strlen(grammar[i]);

            char last = grammar[i][len - 1];
            char secondLast = grammar[i][len - 2];

            // if last symbol is terminal (operator or id)
            if ((last >= 'a' && last <= 'z') || last == ')' || last == '+' || last == '*') {
                trailing[index][strlen(trailing[index])] = last;
            }
            // if last symbol is non-terminal, process recursively
            else if (last >= 'A' && last <= 'Z') {
                findTrailing(last, index);
            }

            // if grammar is A ? aBop form (example: E->T+)
```

```
Compilation results...
--------
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Haritha\OneDrive\Documents\EXP_4.exe
- Output Size: 130.359375 KiB
- Compilation Time: 0.58s
```

OUTPUT:



```
C:\Users\Haritha\OneDrive\D
Enter number of grammar rules: 2
Enter grammar rules (Format: A->B):
E->E+T
E->T

TRAILING() for given grammar:
TRAILING(E) = { + }
TRAILING(E) = { + }

--------------------------------
Process exited after 27.59 seconds with return value 0
Press any key to continue . . .
```