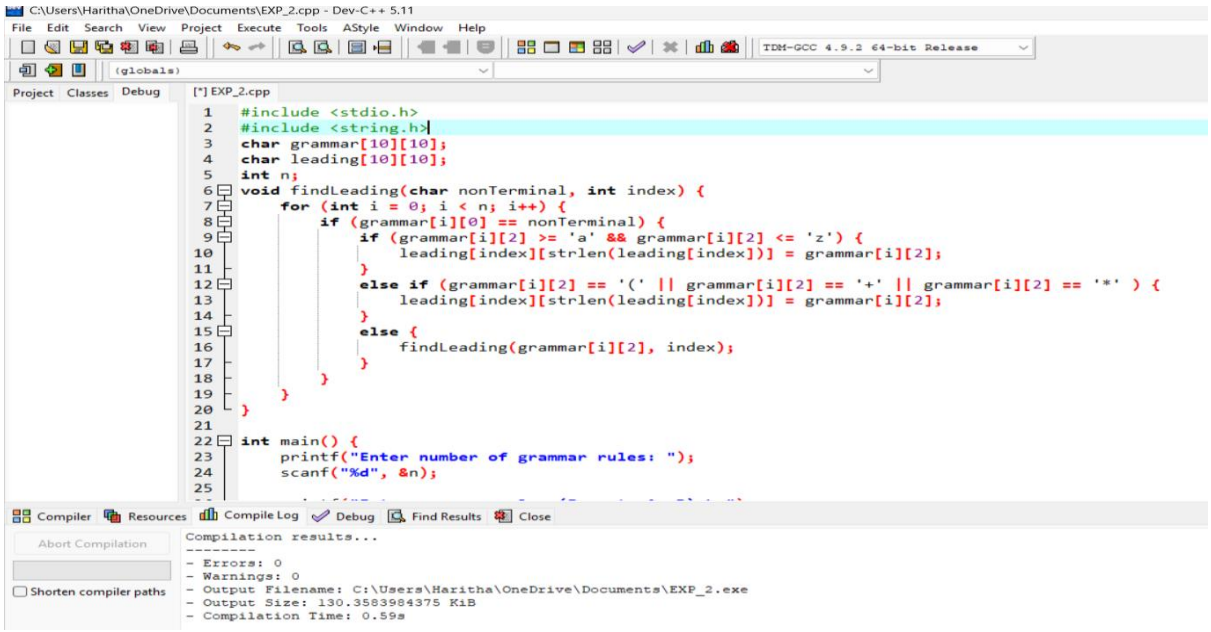
EXPERIMENT-19

AIM:

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

PROGRAM:

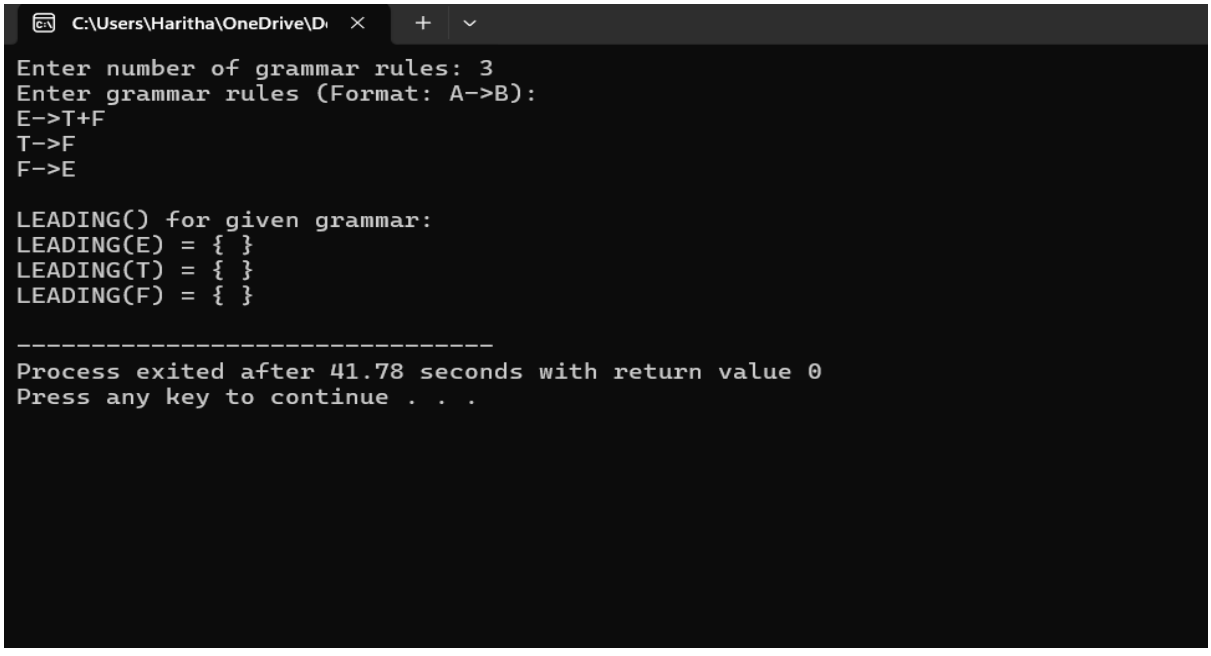


```
#include <stdio.h>
#include <string.h>
char grammar[10][10];
char leading[10][10];
int n;
void findLeading(char nonTerminal, int index) {
    for (int i = 0; i < n; i++) {
        if (grammar[i][0] == nonTerminal) {
            if (grammar[i][2] >= 'a' && grammar[i][2] <= 'z') {
                leading[index][strlen(leading[index])] = grammar[i][2];
            }
            else if (grammar[i][2] == '(' || grammar[i][2] == '+' || grammar[i][2] == '*' ) {
                leading[index][strlen(leading[index])] = grammar[i][2];
            }
            else {
                findLeading(grammar[i][2], index);
            }
        }
    }
}

int main() {
    printf("Enter number of grammar rules: ");
    scanf("%d", &n);
```

```
Compilation results...
--------
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Haritha\OneDrive\Documents\EXP_2.exe
- Output Size: 130.3583984375 KiB
- Compilation Time: 0.59s
```

OUTPUT:



```
C:\Users\Haritha\OneDrive\D
Enter number of grammar rules: 3
Enter grammar rules (Format: A->B):
E->T+F
T->F
F->E

LEADING() for given grammar:
LEADING(E) = { }
LEADING(T) = { }
LEADING(F) = { }

--------------------------------
Process exited after 41.78 seconds with return value 0
Press any key to continue . . .
```