**CSA1443- Compiler Design for Intraprocedural Analysis**

**192321047 – KAUSHIK NARAYANAN.V**

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

**Aim:**To implement a C program that computes the LEADING sets for operators in a given grammar.

**Code:**

#include <stdio.h>

#include <string.h>

void findLeading(char \*grammar, char terminal) {

printf("Leading(%c) = {", terminal);

for (int i = 0; i < strlen(grammar); i++) {

if (grammar[i] == terminal && grammar[i + 1] != '\0') {

printf(" %c ", grammar[i + 1]);

}

}

printf("}\n");

}

int main() {

char grammar[50], terminal;

printf("Enter grammar (like E->E+T|T): ");

scanf("%s", grammar);

printf("Enter terminal symbol: ");

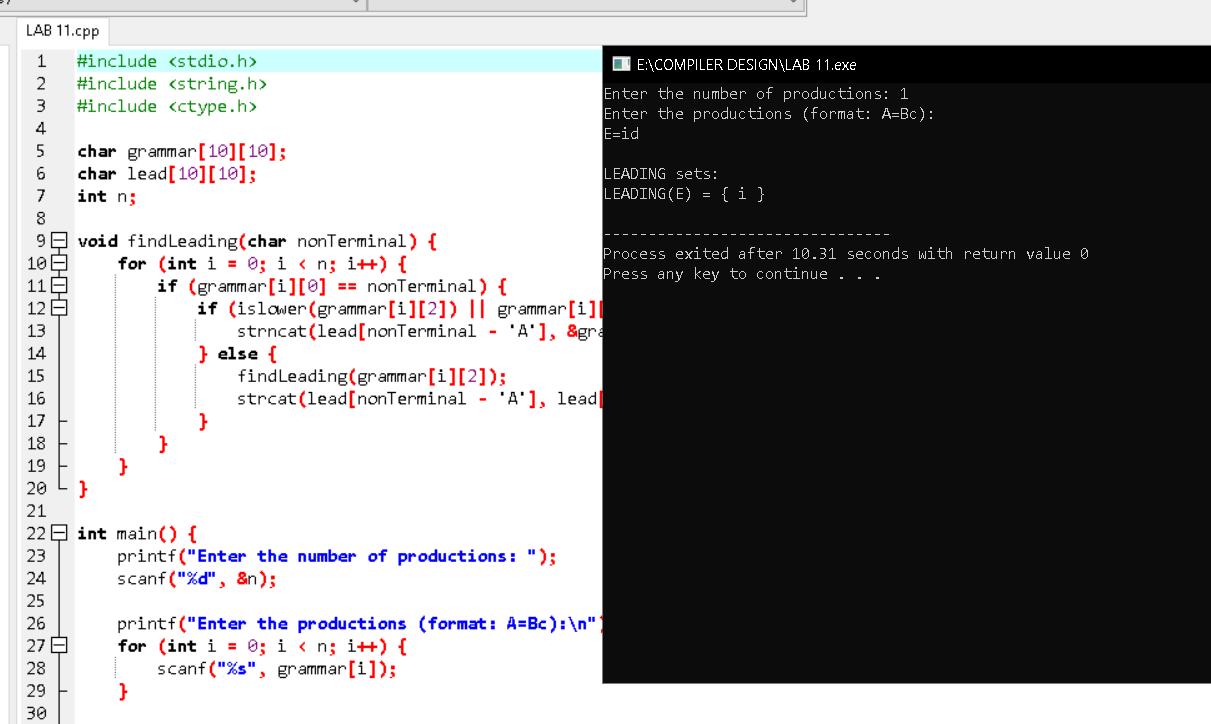
scanf(" %c", &terminal);

findLeading(grammar, terminal);

return 0;

}

**Output:**

****

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

**Aim:**

To implement a C program that computes the TRAILING sets for operators in a given grammar.

**Code:**

#include <stdio.h>

#include <string.h>

void findTrailing(char \*grammar, char terminal) {

printf("Trailing(%c) = {", terminal);

for (int i = 0; i < strlen(grammar); i++) {

if (grammar[i] == terminal && i > 0) {

printf(" %c ", grammar[i - 1]);

}

}

printf("}\n");

}

int main() {

char grammar[50], terminal;

printf("Enter grammar (like E->E+T|T): ");

scanf("%s", grammar);

printf("Enter terminal symbol: ");

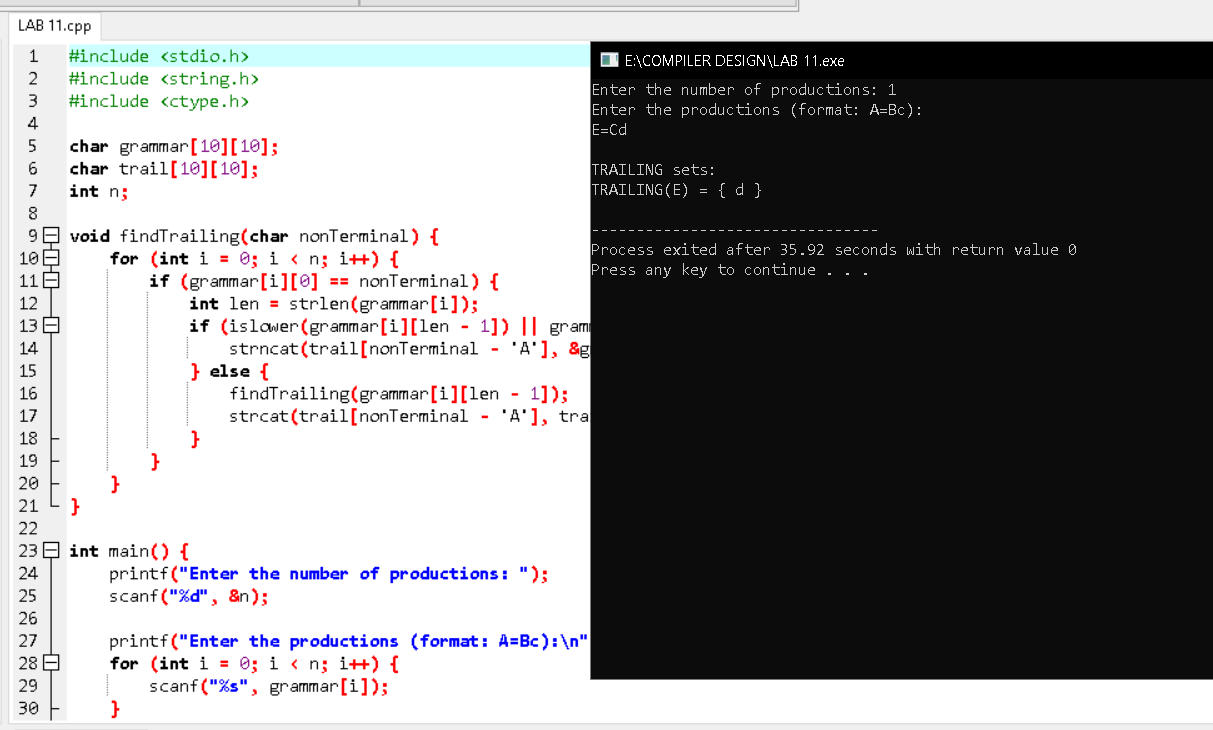
scanf(" %c", &terminal);

findTrailing(grammar, terminal);

return 0;

}

**Output:**

****