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

**Program:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX 100

char \*grammar[MAX];

int numProductions = 0;

void addProduction(char \*production) {

grammar[numProductions] = (char \*)malloc(strlen(production) + 1);

strcpy(grammar[numProductions], production);

numProductions++;

}

void displayProductions() {

printf("Grammar Productions:\n");

for (int i = 0; i < numProductions; i++) {

printf("%s\n", grammar[i]);

}

}

void computeLeading() {

printf("Leading sets:\n");

for (int i = 0; i < numProductions; i++) {

char \*production = grammar[i];

char leadingChar = production[0];

printf("LEADING(%c) = {%c}\n", leadingChar, leadingChar);

}

}

int main() {

addProduction("E -> T E'");

addProduction("E' -> + T E' | ε");

addProduction("T -> F T'");

addProduction("T' -> \* F T' | ε");

addProduction("F -> ( E ) | id");

displayProductions();

computeLeading();

for (int i = 0; i < numProductions; i++) {

free(grammar[i]);

}

return 0;

}

**Output:**

