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

#include <stdio.h>

#include <ctype.h>

#include <string.h>

#define MAX 10

typedef struct {

char lhs;

char rhs[MAX][MAX];

int rhsCount;

} Production;

Production productions[MAX];

int productionCount = 0;

void findLeading(char nonTerminal, char leadingSet[MAX], int \*leadingCount) {

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

if (productions[i].lhs == nonTerminal) {

for (int j = 0; j < productions[i].rhsCount; j++) {

char first = productions[i].rhs[j][0];

if (!isupper(first)) {

if (strchr(leadingSet, first) == NULL) {

leadingSet[\*leadingCount] = first;

(\*leadingCount)++;

}

} else {

findLeading(first, leadingSet, leadingCount);

}

}

}

}

}

int main() {

int i, j;

char leadingSet[MAX];

int leadingCount;

printf("Enter the number of productions: ");

scanf("%d", &productionCount);

getchar();

for (i = 0; i < productionCount; i++) {

printf("Enter production %d (e.g., E=+TE): ", i + 1);

char input[MAX];

fgets(input, MAX, stdin);

input[strcspn(input, "\n")] = 0;

productions[i].lhs = input[0];

productions[i].rhsCount = 0;

char \*token = strtok(&input[2], "|");

while (token) {

strcpy(productions[i].rhs[productions[i].rhsCount++], token);

token = strtok(NULL, "|");

}

}

printf("\nLEADING sets:\n");

for (i = 0; i < productionCount; i++) {

char nonTerminal = productions[i].lhs;

leadingCount = 0;

memset(leadingSet, 0, sizeof(leadingSet));

findLeading(nonTerminal, leadingSet, &leadingCount);

printf("LEADING(%c) = { ", nonTerminal);

for (j = 0; j < leadingCount; j++) {

printf("%c", leadingSet[j]);

if (j < leadingCount - 1) printf(", ");

}

printf(" }\n");

}

return 0;

}

