**Write a C program to find FIRST( ) - predictive parser for the given grammar?**

**#include <stdio.h>**

**#include <ctype.h>**

**#include <string.h>**

**#define MAX 20**

**void findFirst(char, int, int);**

**void addToResultSet(char);**

**int numOfProductions;**

**char productionSet[MAX][MAX];**

**char firstSet[MAX];**

**int firstSetCount;**

**int main() {**

**int i, choice;**

**char c, ch;**

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

**scanf("%d", &numOfProductions);**

**for(i = 0; i < numOfProductions; i++) {**

**printf("Enter production %d: ", i + 1);**

**scanf("%s", productionSet[i]);**

**}**

**do {**

**firstSetCount = 0;**

**printf("\nFind the FIRST of: ");**

**scanf(" %c", &c);**

**findFirst(c, 0, 0);**

**printf("\nFIRST(%c) = { ", c);**

**for(i = 0; i < firstSetCount; i++) {**

**printf("%c ", firstSet[i]);**

**}**

**printf("}\n");**

**printf("\nDo you want to continue (1/0)? ");**

**scanf("%d", &choice);**

**} while(choice == 1);**

**return 0;**

**}**

**void findFirst(char c, int q1, int q2) {**

**int j;**

**if(!(isupper(c))) {**

**addToResultSet(c);**

**}**

**for(j = 0; j < numOfProductions; j++) {**

**if(productionSet[j][0] == c) {**

**if(productionSet[j][2] == '#') {**

**if(productionSet[q1][q2] == '\0')**

**addToResultSet('#');**

**else if(productionSet[q1][q2] != '\0' && (q1 != 0 || q2 != 0))**

**findFirst(productionSet[q1][q2], q1, (q2 + 1));**

**else**

**addToResultSet('#');**

**}**

**else if(!isupper(productionSet[j][2]))**

**addToResultSet(productionSet[j][2]);**

**else**

**findFirst(productionSet[j][2], j, 3);**

**}**

**}**

**}**

**void addToResultSet(char c) {**

**int i;**

**for(i = 0; i < firstSetCount; i++) {**

**if(firstSet[i] == c)**

**return;**

**}**

**firstSet[firstSetCount++] = c;**

**}**

A screenshot of a computer program

AI-generated content may be incorrect.