```
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#define MAX 20
char productions[MAX][MAX];
char firstSet[MAX][MAX];
char followSet[MAX][MAX];
int n;
void findFirst(char* result, char c);
void findFollow(char* result, char c);
int isTerminal(char c);
void addToResult(char* result, char c);
int main() {
 int i;
 char result[20];
 printf("Enter number of productions: ");
 scanf("%d", &n);
 printf("Enter productions (E.g., E=E+T or E=a):\n");
 for (i = 0; i < n; i++) {
 scanf("%s", productions[i]);
 }
 printf("\nFIRST sets:\n");
 for (i = 0; i < n; i++) {
 char nonTerminal = productions[i][0];
 char result[20] = "";
 findFirst(result, nonTerminal);
 printf("FIRST(%c) = { %s }\n", nonTerminal, result);
 strcpy(firstSet[i], result);
 }
 printf("\nFOLLOW sets:\n");
 for (i = 0; i < n; i++) {
 char nonTerminal = productions[i][0];
 char result[20] = "";
 findFollow(result, nonTerminal);
 printf("FOLLOW(%c) = { %s }\n", nonTerminal, result);
 strcpy(followSet[i], result);
 }
 return 0;
}
int isTerminal(char c) {
 return !isupper(c) && c != '#';
}
void addToResult(char* result, char c) {
 if (strchr(result, c) == NULL) {
 int len = strlen(result);
 result[len] = c;
 result[len + 1] = '\0';
 }}
void findFirst(char* result, char c) {
 if (isTerminal(c)) {
 addToResult(result, c);
 return;
 }
 for (int i = 0; i < n; i++) {
 if (productions[i][0] == c) {
 if (productions[i][2] == '#') {
 addToResult(result, '#');
 } else {
 int j = 2;
 while (productions[i][j] != '\0') {
```

```
 char sym = productions[i][j];
 char temp[20] = "";
 findFirst(temp, sym);
 for (int k = 0; k < strlen(temp); k++) {
 if (temp[k] != '#')
 addToResult(result, temp[k]);
 }
 if (strchr(temp, '#') == NULL) break;
 j++;
 } }}}}
void findFollow(char* result, char c) {
 if (c == productions[0][0]) {
 addToResult(result, '$');
 }
 for (int i = 0; i < n; i++) {
 for (int j = 2; j < strlen(productions[i]); j++) {
 if (productions[i][j] == c) {
 if (productions[i][j + 1] != '\0') {
 char temp[20] = "";
 findFirst(temp, productions[i][j + 1]);
 for (int k = 0; k < strlen(temp); k++) {
 if (temp[k] != '#')
 addToResult(result, temp[k]);
 }
 if (strchr(temp, '#') != NULL) {
 char temp2[20] = "";
 findFollow(temp2, productions[i][0]);
 for (int k = 0; k < strlen(temp2); k++)
 addToResult(result, temp2[k]);
 }
 } else if (productions[i][j + 1] == '\0' && productions[i][0] != c) {
 char temp[20] = "";
 findFollow(temp, productions[i][0]);
 for (int k = 0; k < strlen(temp); k++)
 addToResult(result, temp[k]);
 } } } } }
```