**CSA1443-COMPILER DESIGN FOR INTRAPROCEDURAL ANALYSIS**

**NAME: Ajay Kumar J**

**REG NO: 192372052**

**PROGRAM 8**

**Aim:**

To implement a C program that computes the **FOLLOW()** sets for a given context-free grammar (CFG) as part of a predictive parser. The **FOLLOW()** sets indicate which terminals can appear immediately to the right of a non-terminal in some sentential form.

**Code:**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

#define MAX 10

#define ALPHABET\_SIZE 26

int isTerminal(char c) {

return !isupper(c);

}

int isNonTerminal(char c) {

return isupper(c);

}

void findFollow(char grammar[MAX][MAX], int n, char nonTerminal, char follow[MAX]) {

int changed = 1;

while (changed) {

changed = 0;

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

for (int j = 2; grammar[i][j] != '\0'; j++) {

if (grammar[i][j] == nonTerminal) {

if (isTerminal(grammar[i][j + 1])) {

follow[strlen(follow)] = grammar[i][j + 1];

changed = 1;

} else if (isNonTerminal(grammar[i][j + 1])) {

follow[strlen(follow)] = grammar[i][j + 1];

changed = 1;

} else if (grammar[i][j + 1] == '\0') {

follow[strlen(follow)] = grammar[i][0]; // Left-hand side non-terminal

changed = 1;

}

}

}

}

}

}

int main() {

int n;

char grammar[MAX][MAX], follow[MAX];

char nonTerminals[MAX] = "SAB";

printf("Enter number of productions: ");

scanf("%d", &n);

getchar();

printf("Enter the productions (in the form: A->a or A->B):\n");

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

fgets(grammar[i], MAX, stdin);

grammar[i][strcspn(grammar[i], "\n")] = 0

}

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

follow[i] = '\0'; // Clear FOLLOW sets

}

follow[0] = '$';

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

char nonTerminal = nonTerminals[i];

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

memset(follow, 0, sizeof(follow)); // Clear the FOLLOW set

findFollow(grammar, n, nonTerminal, follow);

for (int j = 0; follow[j] != '\0'; j++) {

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

}

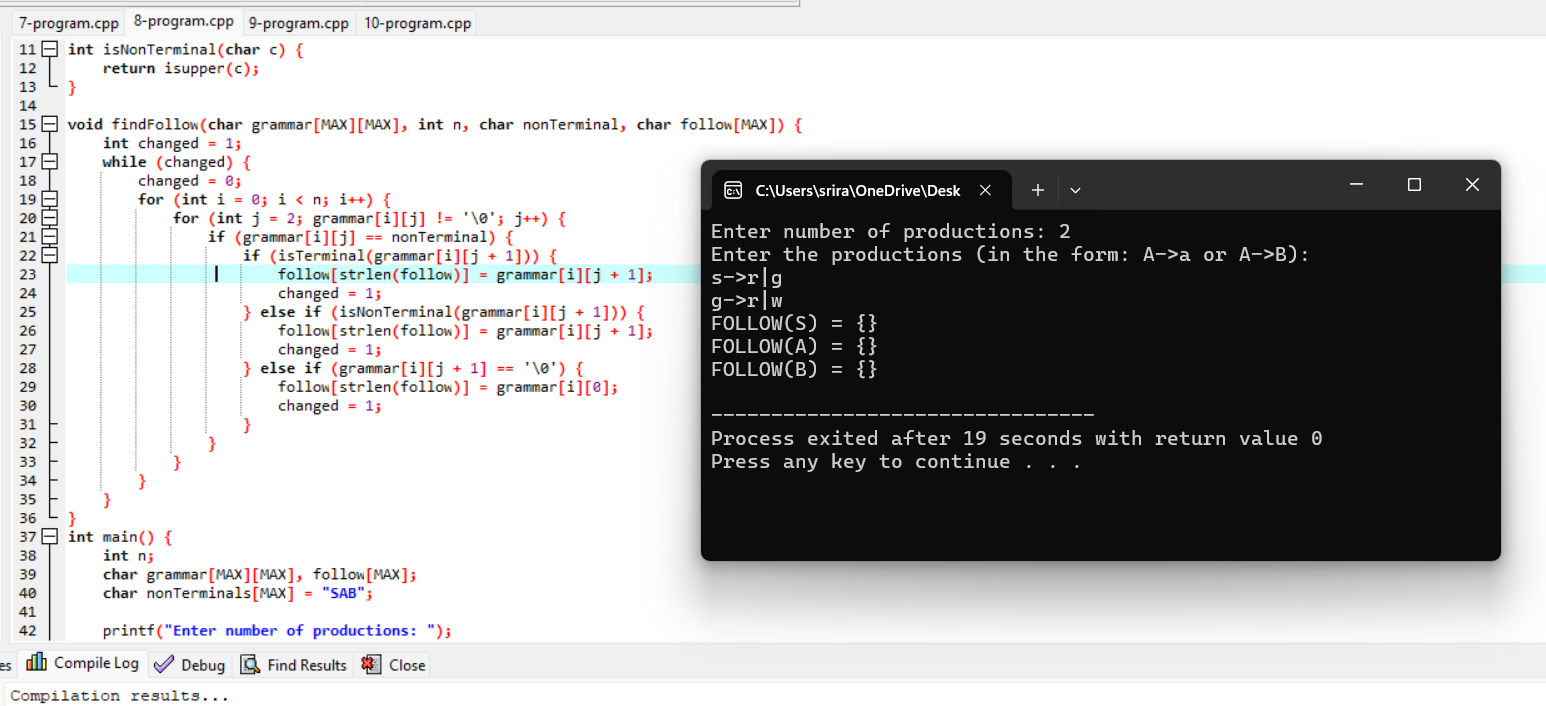
printf("}\n");

}

return 0;

}

**Output:**

****