**CSA1443-COMPILER DESIGN FOR INTRAPROCEDURAL ANALYSIS**

**NAME: Ajay Kumar J**

**REG NO: 192372052**

**PROGRAM 7**

**Aim:**

To implement a C program that computes the **FIRST()** sets for a given context-free grammar (CFG) as part of a predictive parser.

**Code:**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

#define MAX 10

int isTerminal(char c) {

return !isupper(c);

}

void findFirst(char grammar[MAX][MAX], int n, char nonTerminal, char first[MAX]) {

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

if (grammar[i][0] == nonTerminal) {

if (isTerminal(grammar[i][2])) {

first[strlen(first)] = grammar[i][2];

} else {

first[strlen(first)] = grammar[i][2];

}

}

}

}

int main() {

int n;

char grammar[MAX][MAX], first[MAX];

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 < n; i++) {

char nonTerminal = grammar[i][0

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

memset(first, 0, sizeof(first));

findFirst(grammar, n, nonTerminal, first);

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

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

}

printf("}\n");

}

return 0;

}

**Output:**

