CSB 353: Compiler Design

LAB 7

Submitted By:

Name: PREM KUMAR

Roll No: 191210037

Branch: CSE

Semester: 6 th

Submitted To: Dr. Shelly Sachdeva

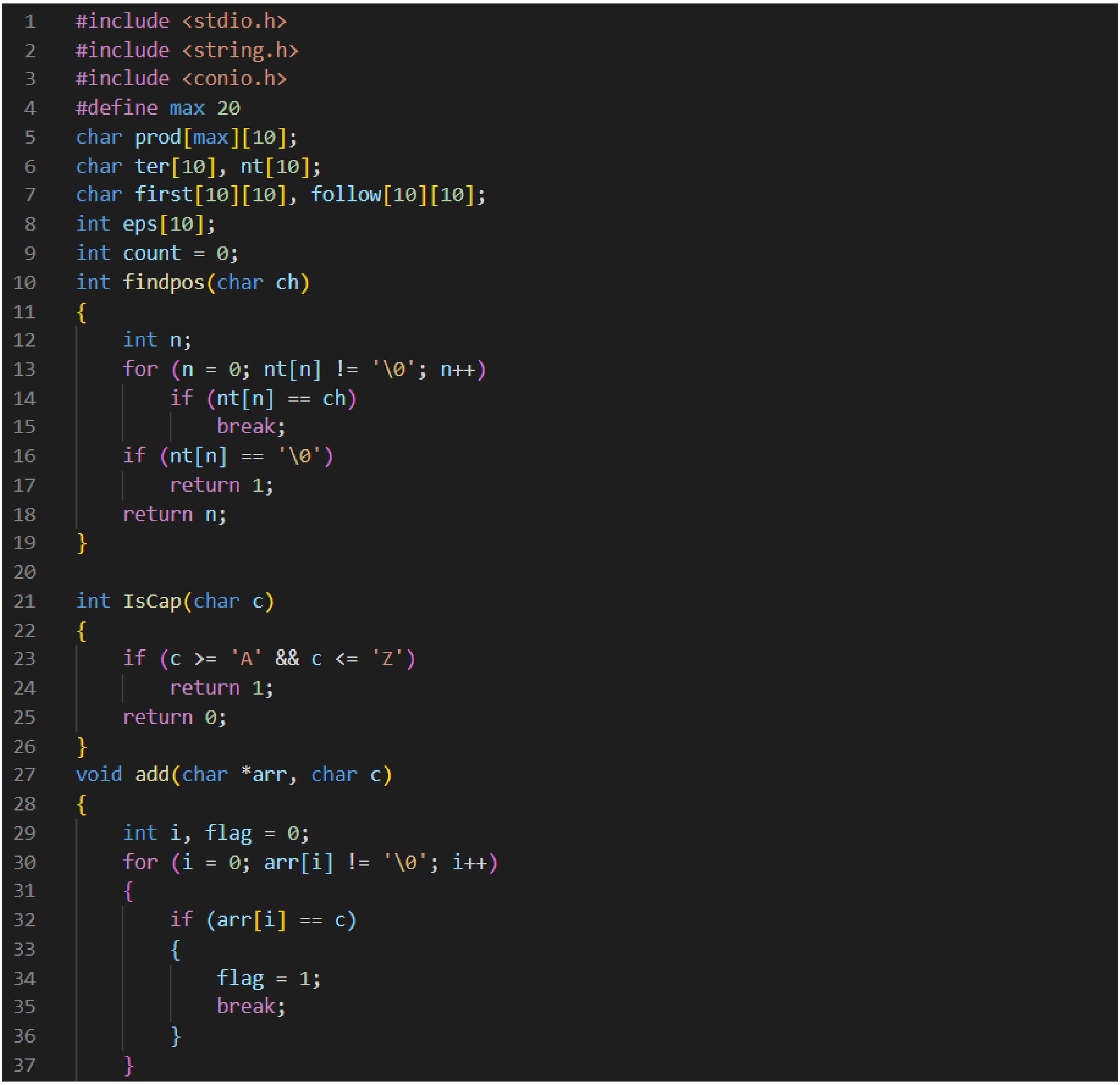
Department of Computer Science and Engineering

NATIONAL INSTITUTE OF TECHNOLOGY DELHI

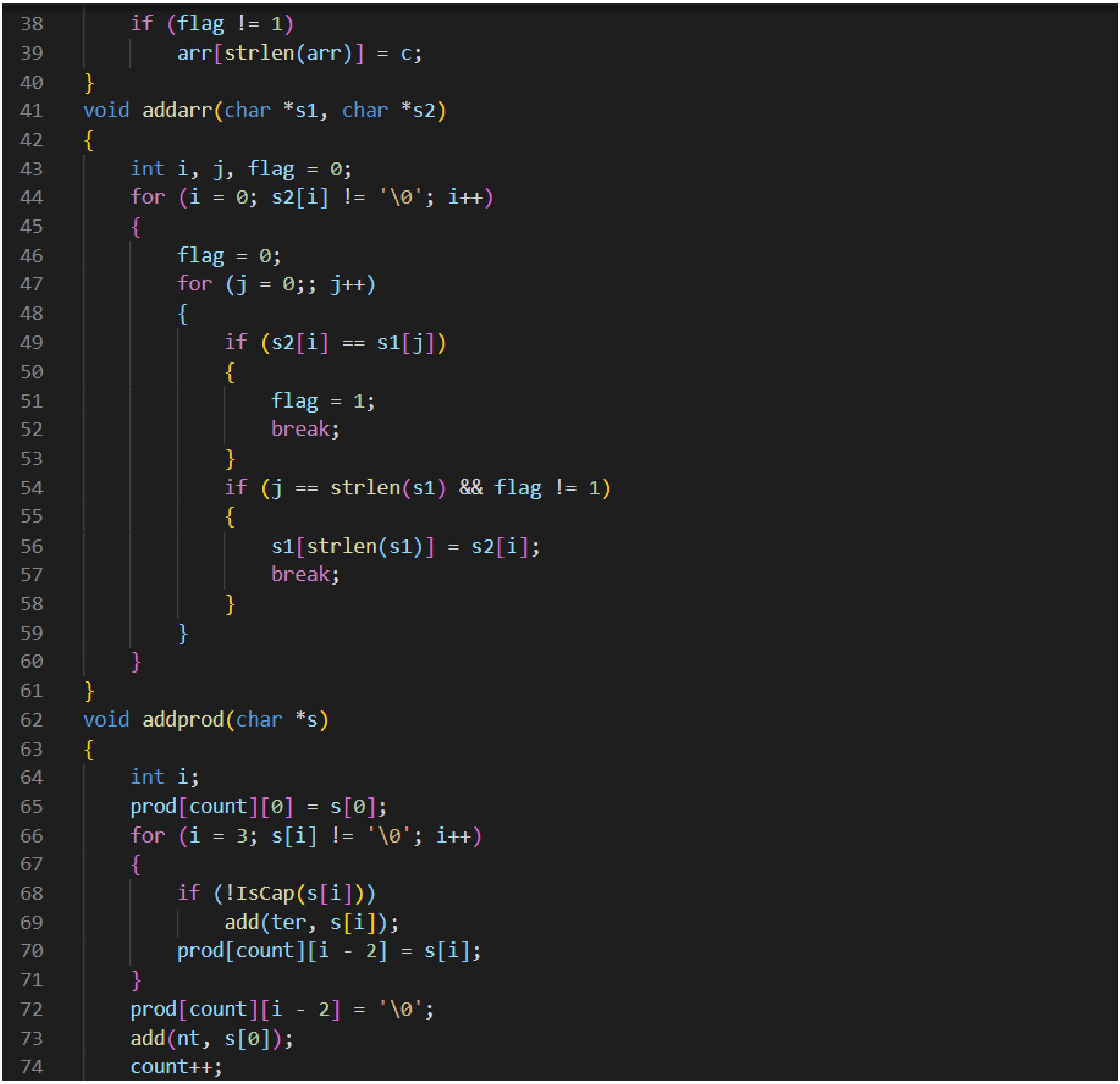
2019-2023

Ques 1. Write a C program to find First and Follow for all non-terminals.

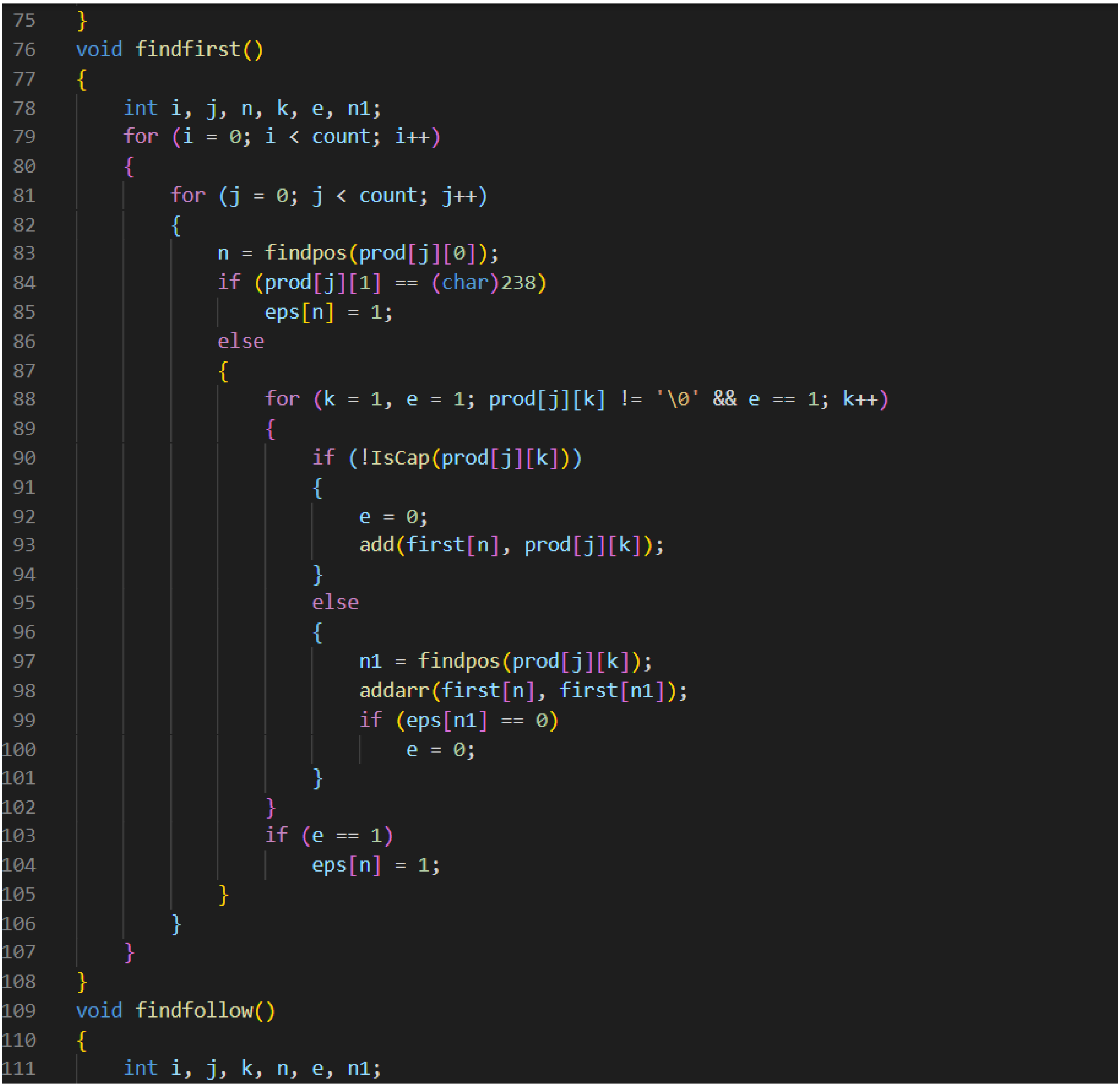
Code:



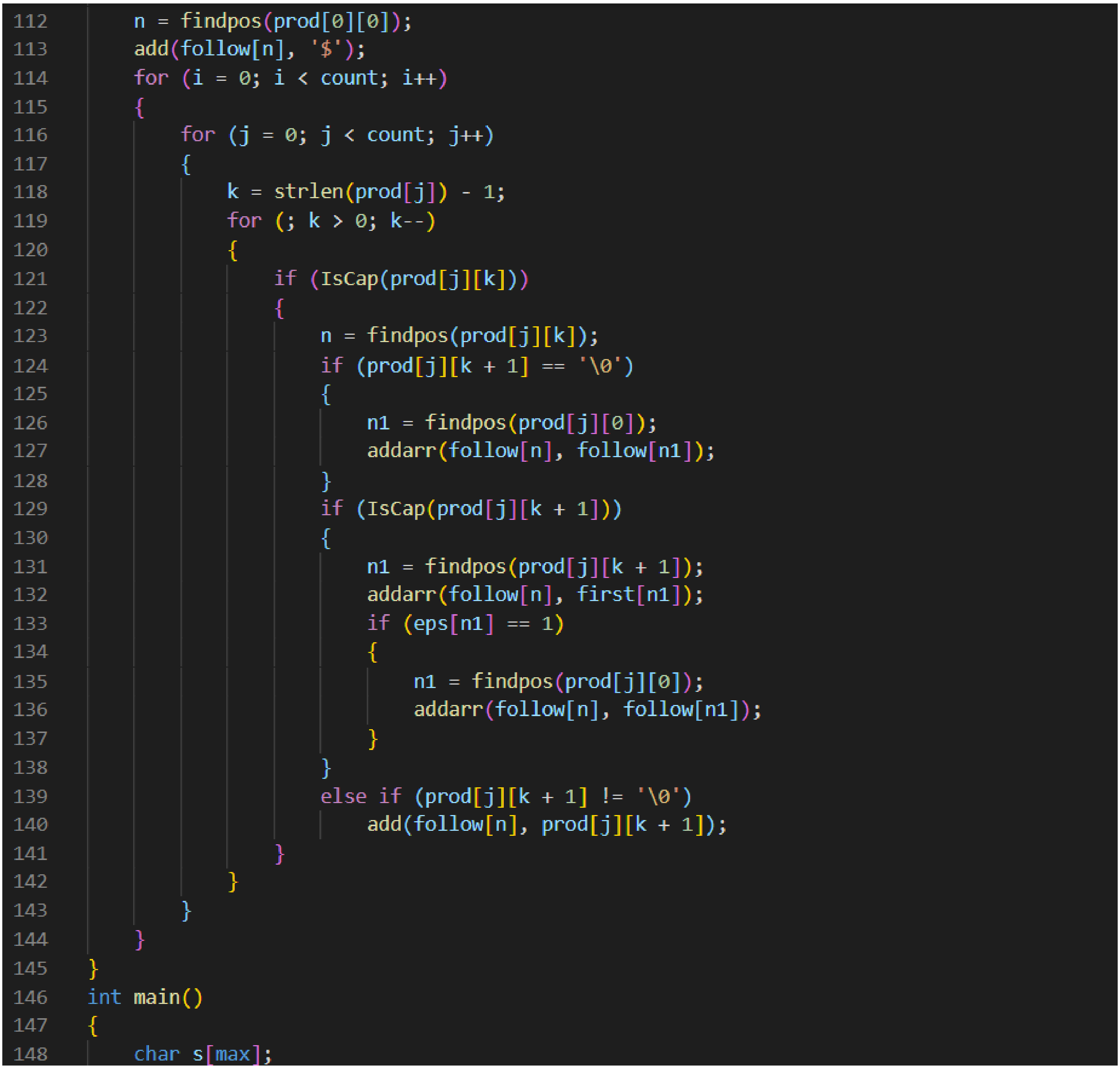
```c
#include <stdio.h>
#include <string.h>
#include <conio.h>
#define max 20
char prod[max][10];
char ter[10], nt[10];
char first[10][10], follow[10][10];
int eps[10];
int count = 0;
int findpos(char ch)
{
    int n;
    for (n = 0; nt[n] != '\0'; n++)
        if (nt[n] == ch)
            break;
    if (nt[n] == '\0')
        return 1;
    return n;
}

int IsCap(char c)
{
    if (c >= 'A' && c <= 'Z')
        return 1;
    return 0;
}
void add(char *arr, char c)
{
    int i, flag = 0;
    for (i = 0; arr[i] != '\0'; i++)
    {
        if (arr[i] == c)
        {
            flag = 1;
            break;
        }
    }
```



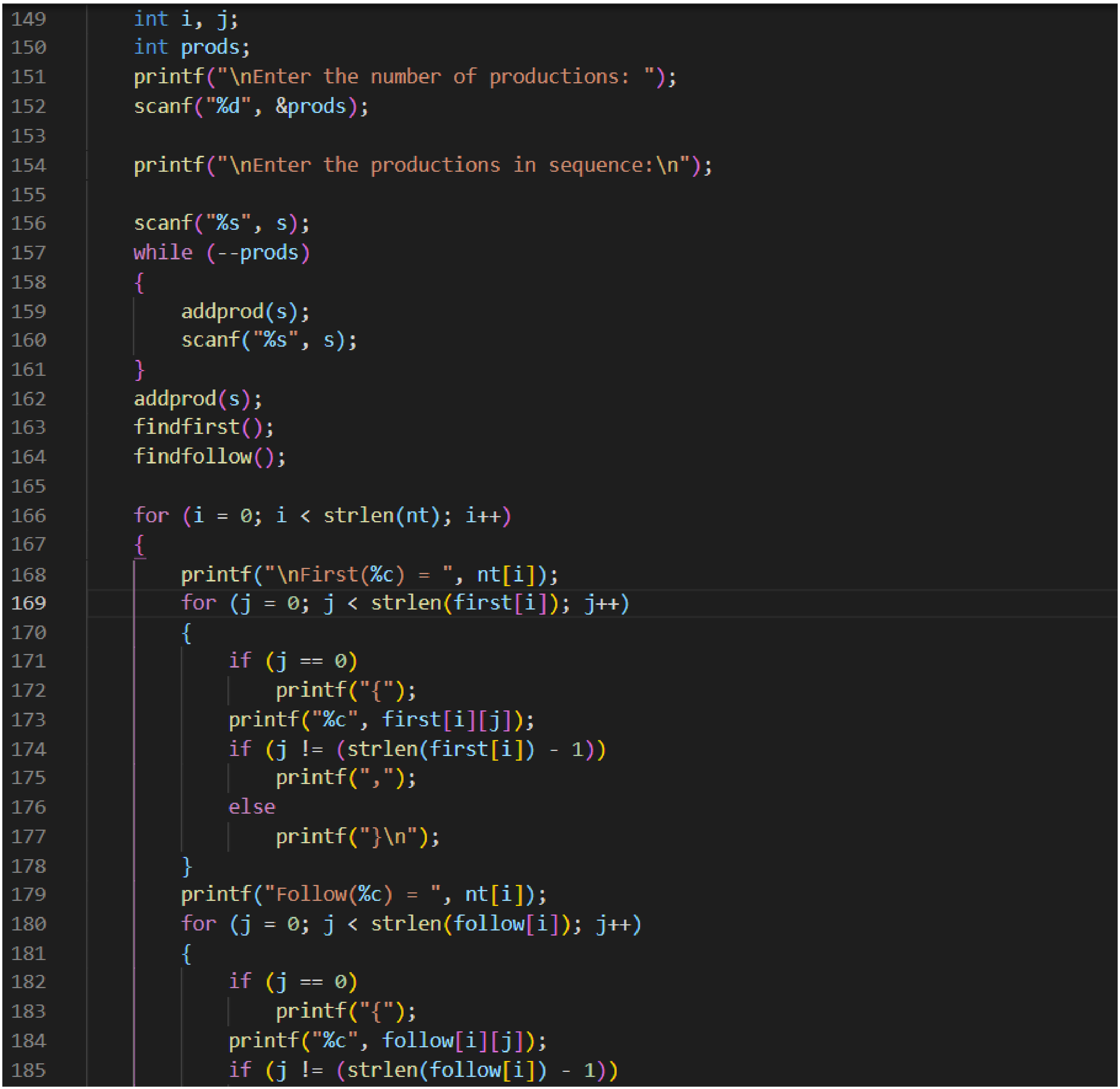
```
    if (flag != 1)
        arr[strlen(arr)] = c;
}
void addarr(char *s1, char *s2)
{
    int i, j, flag = 0;
    for (i = 0; s2[i] != '\0'; i++)
    {
        flag = 0;
        for (j = 0;; j++)
        {
            if (s2[i] == s1[j])
            {
                flag = 1;
                break;
            }
            if (j == strlen(s1) && flag != 1)
            {
                s1[strlen(s1)] = s2[i];
                break;
            }
        }
    }
}
void addprod(char *s)
{
    int i;
    prod[count][0] = s[0];
    for (i = 3; s[i] != '\0'; i++)
    {
        if (!IsCap(s[i]))
            add(ter, s[i]);
        prod[count][i - 2] = s[i];
    }
    prod[count][i - 2] = '\0';
    add(nt, s[0]);
    count++;
```



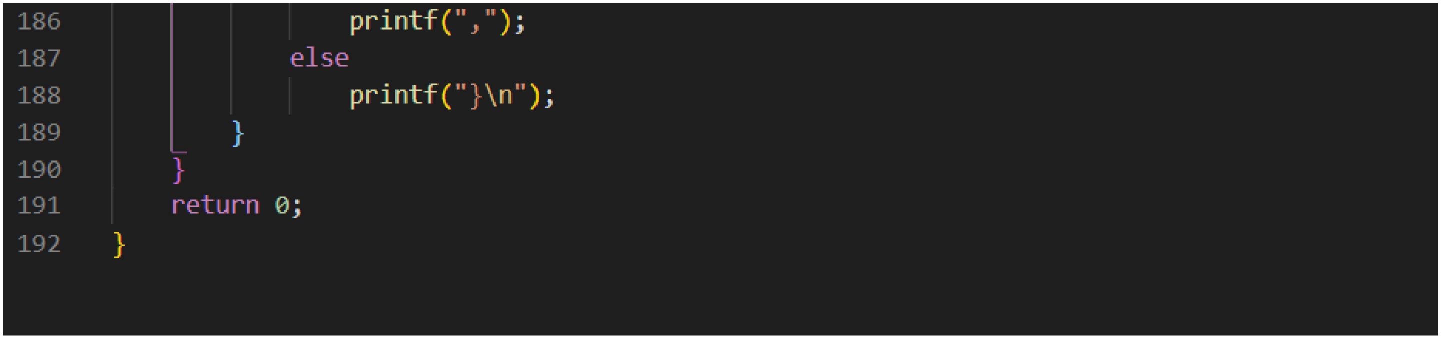
```
}
void findfirst()
{
    int i, j, n, k, e, n1;
    for (i = 0; i < count; i++)
    {
        for (j = 0; j < count; j++)
        {
            n = findpos(prod[j][0]);
            if (prod[j][1] == (char)238)
                eps[n] = 1;
            else
            {
                for (k = 1, e = 1; prod[j][k] != '\0' && e == 1; k++)
                {
                    if (!IsCap(prod[j][k]))
                    {
                        e = 0;
                        add(first[n], prod[j][k]);
                    }
                    else
                    {
                        n1 = findpos(prod[j][k]);
                        addarr(first[n], first[n1]);
                        if (eps[n1] == 0)
                            e = 0;
                    }
                }
                if (e == 1)
                    eps[n] = 1;
            }
        }
    }
}
void findfollow()
{
    int i, j, k, n, e, n1;
```



```c
    n = findpos(prod[0][0]);
    add(follow[n], '$');
    for (i = 0; i < count; i++)
    {
        for (j = 0; j < count; j++)
        {
            k = strlen(prod[j]) - 1;
            for (; k > 0; k--)
            {
                if (IsCap(prod[j][k]))
                {
                    n = findpos(prod[j][k]);
                    if (prod[j][k + 1] == '\0')
                    {
                        n1 = findpos(prod[j][0]);
                        addarr(follow[n], follow[n1]);
                    }
                    if (IsCap(prod[j][k + 1]))
                    {
                        n1 = findpos(prod[j][k + 1]);
                        addarr(follow[n], first[n1]);
                        if (eps[n1] == 1)
                        {
                            n1 = findpos(prod[j][0]);
                            addarr(follow[n], follow[n1]);
                        }
                    }
                    else if (prod[j][k + 1] != '\0')
                        add(follow[n], prod[j][k + 1]);
                }
            }
        }
    }
}
int main()
{
    char s[max];
```

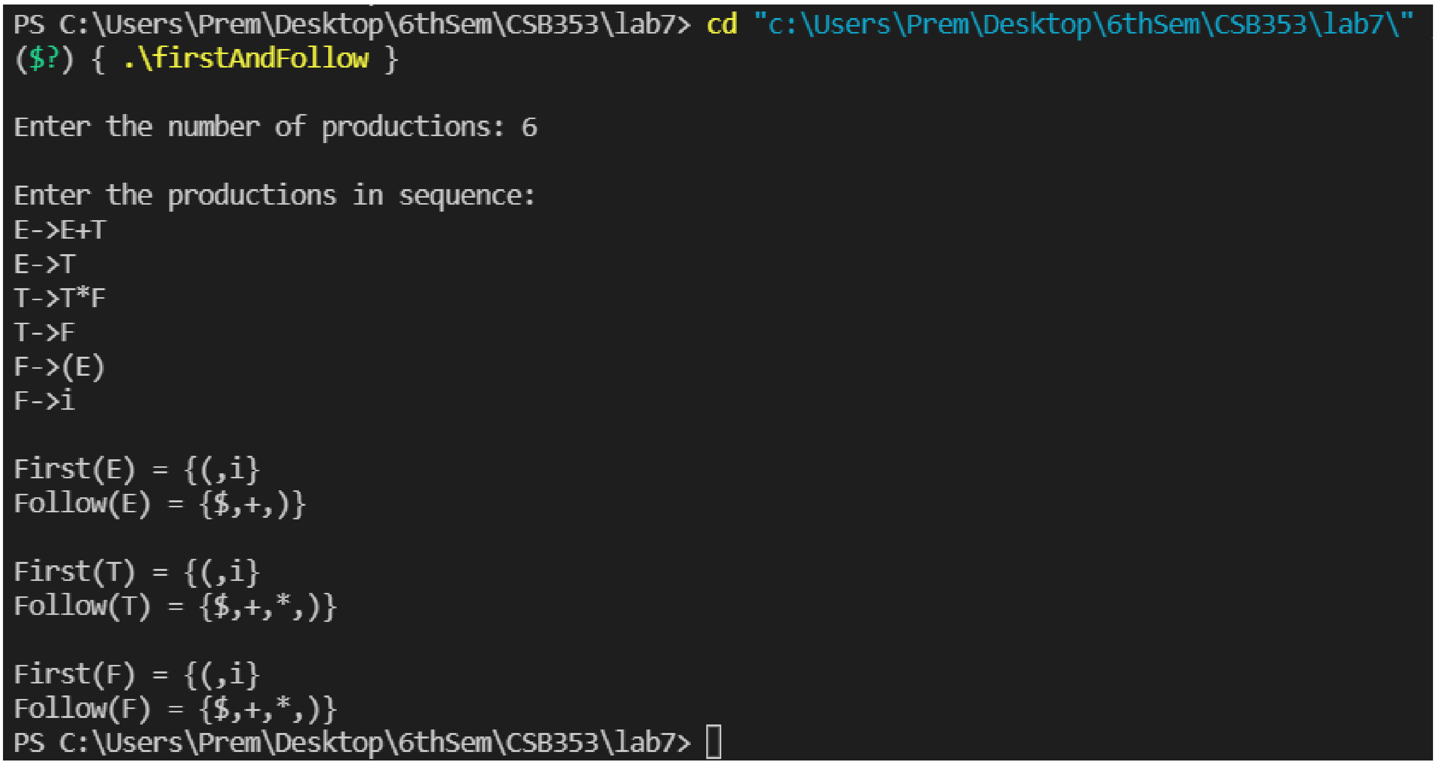


```c
    int i, j;
    int prods;
    printf("\nEnter the number of productions: ");
    scanf("%d", &prods);

    printf("\nEnter the productions in sequence:\n");

    scanf("%s", s);
    while (--prods)
    {
        addprod(s);
        scanf("%s", s);
    }
    addprod(s);
    findfirst();
    findfollow();

    for (i = 0; i < strlen(nt); i++)
    {
        printf("\nFirst(%c) = ", nt[i]);
        for (j = 0; j < strlen(first[i]); j++)
        {
            if (j == 0)
                printf("{");
            printf("%c", first[i][j]);
            if (j != (strlen(first[i]) - 1))
                printf(",");
            else
                printf("}\n");
        }
        printf("Follow(%c) = ", nt[i]);
        for (j = 0; j < strlen(follow[i]); j++)
        {
            if (j == 0)
                printf("{");
            printf("%c", follow[i][j]);
            if (j != (strlen(follow[i]) - 1))
```



```c
                printf(",");
            else
                printf("}\n");
        }
    }
    return 0;
}
```

Output:



```
PS C:\Users\Prem\Desktop\6thSem\CSB353\lab7> cd "c:\Users\Prem\Desktop\6thSem\CSB353\lab7\"
($?) { .\firstAndFollow }

Enter the number of productions: 6

Enter the productions in sequence:
E->E+T
E->T
T->T*F
T->F
F->(E)
F->i

First(E) = {(,i}
Follow(E) = {$,+,)}

First(T) = {(,i}
Follow(T) = {$,+,*,)}

First(F) = {(,i}
Follow(F) = {$,+,*,)}
PS C:\Users\Prem\Desktop\6thSem\CSB353\lab7>
```