Bangladesh University of Business and Technology (BUBT)

Course Title: Compiler Design Lab

Course Code: CSE 324

Lab work: Assignment

SUBMITTED BY	**SUBMITTED TO**
Name: Mehedi Hasan	Name: Md. Saddam Hossain
Id: 18192203022	Dept. of CSE
Intake: 32	
Section: 1	
Program: B.Sc Engg in CSE	

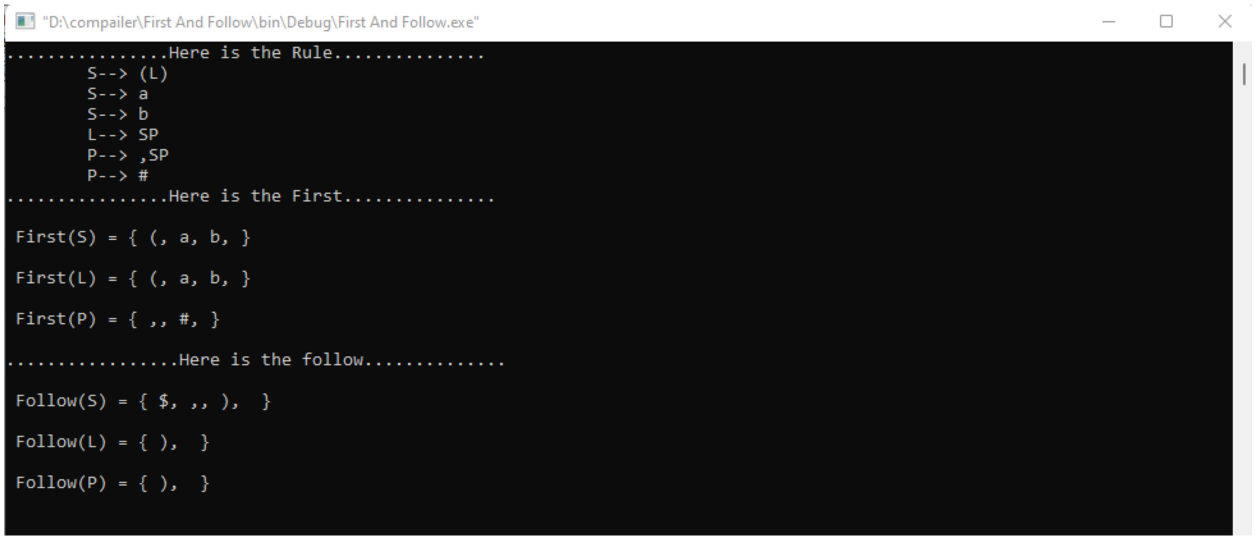
1. First and Follow Calculation

Code:

```
#include<stdio.h>
#include<ctype.h>
#include<string.h>

void followfirst(char, int, int);
void follow(char c);

void findfirst(char, int, int);

int count, n = 0;


char calc_first[10][100];

char calc_follow[10][100];
int m = 0;

char production[10][10];
char f[10], first[10];
int k;
char ck;
int e;

int main(int argc, char **argv)
{
    int jm = 0;
    int km = 0;
    int i, choice;
    char c, ch;
    count = 6;

    strcpy(production[0], "S=(L)");
    strcpy(production[1], "S=a");
    strcpy(production[2], "S=b");
    strcpy(production[3], "L=SP");
    strcpy(production[4], "P=,SP");
    strcpy(production[5], "P=#");


    int k2;
    char done[count];
    int ptr = -1;

    for(k = 0; k < count; k++) {
        for(k2 = 0; k2 < 100; k2++) {
```

```
        calc_first[k][k2] = '!';
    }
}
int point1 = 0, point2, x2;
printf("...............Here is the Rule..............\n");
printf("\tS--> (L)\n\tS--> a\n\tS--> b\n\tL--> SP\n\tP--> ,SP \n\tP--> #\n");


 printf("...............Here is the First..............\n");

for(k = 0; k < count; k++)
{
    c = production[k][0];
    point2 = 0;
    x2 = 0;

    for(k2 = 0; k2 <= ptr; k2++)
        if(c == done[k2])
            x2 = 1;

    if (x2 == 1)
        continue;

    findfirst(c, 0, 0);
    ptr += 1;

    // Adding c to the calculated list
    done[ptr] = c;
    printf("\n First(%c) = { ", c);
    calc_first[point1][point2++] = c;

    for(i = 0 + jm; i < n; i++) {
        int l2 = 0, chk = 0;

        for(l2 = 0; l2 < point2; l2++) {

            if (first[i] == calc_first[point1][l2])
            {
                chk = 1;
                break;
            }
        }
        if(chk == 0)
        {
            printf("%c, ", first[i]);
            calc_first[point1][point2++] = first[i];
        }
    }
    printf("}\n");
    jm = n;
    point1++;
}
printf("\n");
printf("................Here is the follow.............\n\n");
char donee[count];
ptr = -1;

for(k = 0; k < count; k++) {
    for(k2 = 0; k2 < 100; k2++) {
        calc_follow[k][k2] = '!';
    }
}
point1 = 0;
int land = 0;
for(e = 0; e < count; e++)
{
    ck = production[e][0];
    point2 = 0;
    x2 = 0;

    for(k2 = 0; k2 <= ptr; k2++)
        if(ck == donee[k2])
            x2 = 1;

    if (x2 == 1)
        continue;
    land += 1;
    follow(ck);
    ptr += 1;


    donee[ptr] = ck;
    printf(" Follow(%c) = { ", ck);
    calc_follow[point1][point2++] = ck;

    for(i = 0 + km; i < m; i++) {
        int l2 = 0, chk = 0;
        for(l2 = 0; l2 < point2; l2++)
        {
            if (f[i] == calc_follow[point1][l2])
            {
                chk = 1;
                break;
            }
        }
        if(chk == 0)
```

```
            {
                printf("%c, ", f[i]);
                calc_follow[point1][point2++] = f[i];
            }
        }
        printf(" }\n\n");
        km = m;
        point1++;
    }

    getch();
}

void follow(char c)
{
    int i, j;

    if(production[0][0] == c) {
        f[m++] = '$';
    }
    for(i = 0; i < 10; i++)
    {
        for(j = 2;j < 10; j++)
        {
            if(production[i][j] == c)
            {
                if(production[i][j+1] != '\0')
                {
                    followfirst(production[i][j+1], i, (j+2));
                }

                if(production[i][j+1]=='\0' && c!=production[i][0])
                {
                    follow(production[i][0]);
                }
            }
        }
    }
}

void findfirst(char c, int q1, int q2)
{
    int j;

    if(!(isupper(c))) {
        first[n++] = c;
    }
    for(j = 0; j < count; j++)
    {
        if(production[j][0] == c)
        {
            if(production[j][2] == '#')
            {
                if(production[q1][q2] == '\0')
                    first[n++] = '#';
                else if(production[q1][q2] != '\0'
                         && (q1 != 0 || q2 != 0))
                {
                    findfirst(production[q1][q2], q1, (q2+1));
                }
                else
                    first[n++] = '#';
            }
            else if(!isupper(production[j][2]))
            {
                first[n++] = production[j][2];
            }
            else
            {
                findfirst(production[j][2], j, 3);
            }
        }
    }
}

void followfirst(char c, int c1, int c2)
{
    int k;

    if(!(isupper(c)))
        f[m++] = c;
    else
    {
        int i = 0, j = 1;
        for(i = 0; i < count; i++)
        {
            if(calc_first[i][0] == c)
                break;
        }

        while(calc_first[i][j] != '!')
        {
            if(calc_first[i][j] != '#')
            {
                f[m++] = calc_first[i][j];
            }
            else
            {
               if(production[c1][c2] == '\0')
               {
                  follow(production[c1][0]);
               }
               else
               {
                  followfirst(production[c1][c2], c1, c2+1);
               }
            }
            j++;
         }
      }
   }
```
Output:



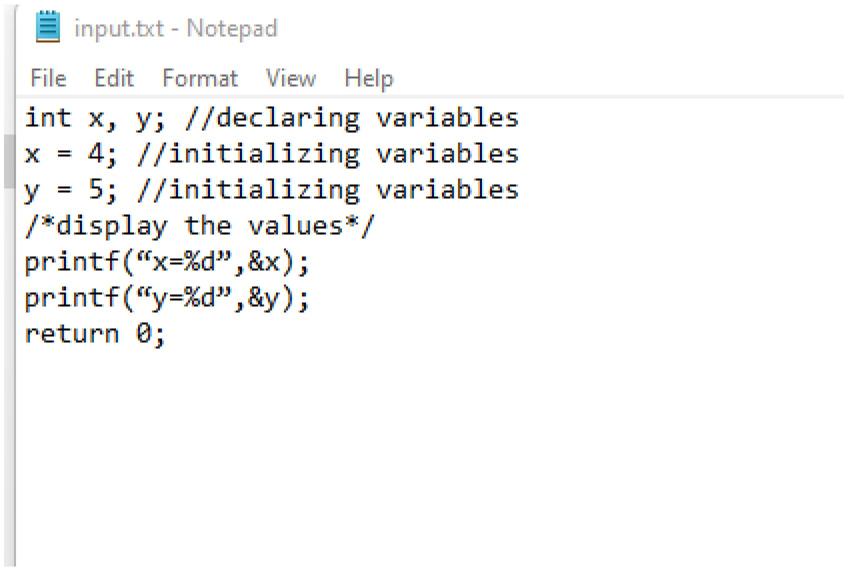
```
"D:\compailer\First And Follow\bin\Debug\First And Follow.exe"
...............Here is the Rule..............
        S--> (L)
        S--> a
        S--> b
        L--> SP
        P--> ,SP
        P--> #
...............Here is the First..............

 First(S) = { (, a, b, }

 First(L) = { (, a, b, }

 First(P) = { ,, #, }

................Here is the follow.............

 Follow(S) = { $, ,, ), }

 Follow(L) = { ), }

 Follow(P) = { ), }
```

2.Comment Detection and remove comment.

Code:

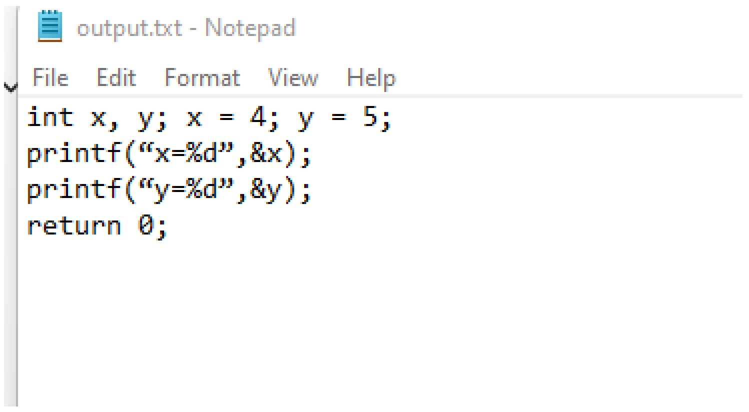
```
#include <stdio.h>
#include <stdlib.h>
void check_comment (char) ;
void block_comment () ;
void single_comment () ;
FILE *fp , *fp2;
int main(void)
{
   char c;
   fp = fopen ("input.txt","r") ;
   fp2 = fopen ("output.txt","w") ;
   while((c=fgetc(fp))!=EOF)
   check_comment(c);
```

```
    fclose(fp);
    fclose(fp2);
return 0;
}
void check_comment(char c)
{
    char d;
    if( c == '/')
    {
        if((d=fgetc(fp))=='*')
        block_comment();
        else if( d == '/')
        {
        single_comment();
        }
        else
        {
            fputc(c,fp2);
            fputc(d,fp2);
        }
    }
    else
    fputc(c,fp2);
}
void block_comment()
{
    char d,e;
    while((d=fgetc(fp))!=EOF)
    {
        if(d=='*')
        {
        e=fgetc(fp);
        if(e=='/')
        return;
        }
    }
}
void single_comment()
{
    char d,e;
    while((d=fgetc(fp))!=EOF)
    {
    if(d=='\n')
    return;
    }
}
```
Input File:



```
int x, y; //declaring variables
x = 4; //initializing variables
y = 5; //initializing variables
/*display the values*/
printf(“x=%d”,&x);
printf(“y=%d”,&y);
return 0;
```

Output File:



```
int x, y; x = 4; y = 5;
printf(“x=%d”,&x);
printf(“y=%d”,&y);
return 0;
```