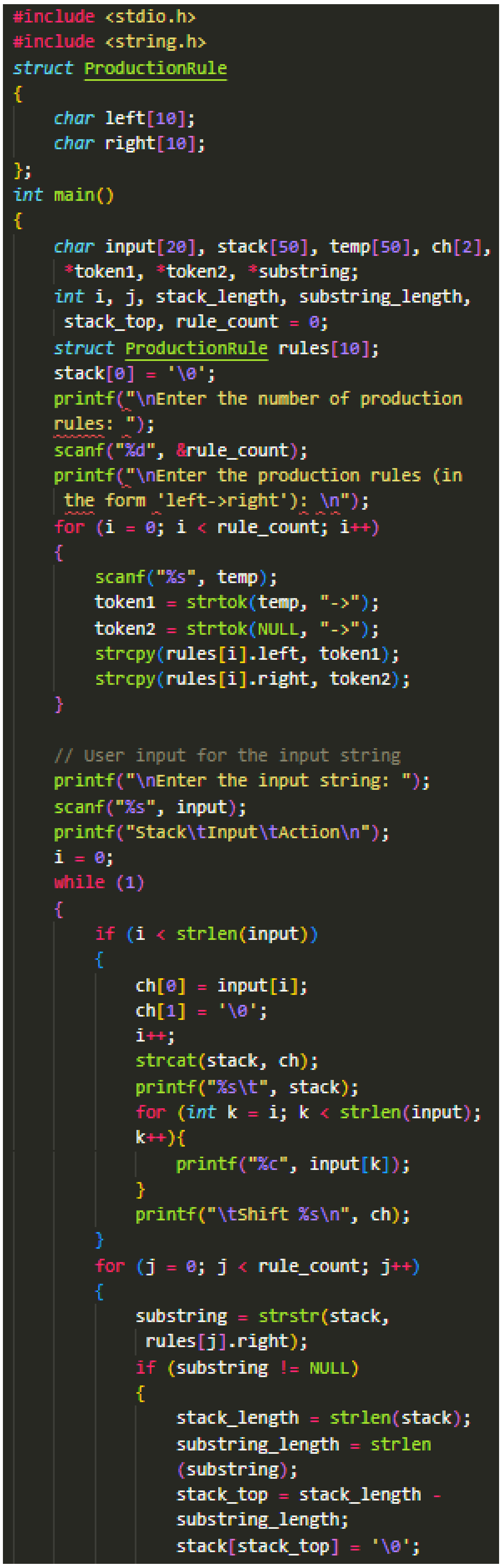
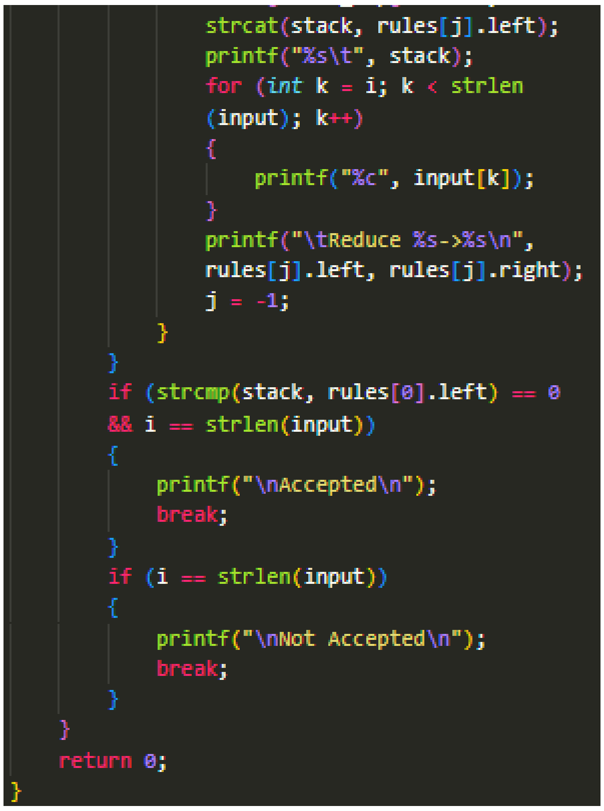
CODE (Expt9)

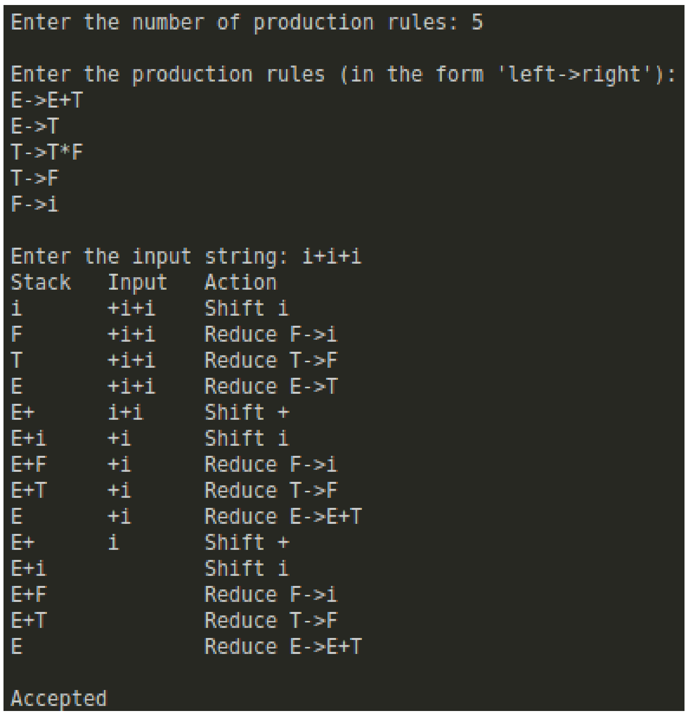


```
#include <stdio.h>
#include <string.h>
struct ProductionRule
{
    char left[10];
    char right[10];
};
int main()
{
    char input[20], stack[50], temp[50], ch[2],
     *token1, *token2, *substring;
    int i, j, stack_length, substring_length,
     stack_top, rule_count = 0;
    struct ProductionRule rules[10];
    stack[0] = '\0';
    printf("\nEnter the number of production
    rules: ");
    scanf("%d", &rule_count);
    printf("\nEnter the production rules (in
     the form 'left->right'): \n");
    for (i = 0; i < rule_count; i++)
    {
        scanf("%s", temp);
        token1 = strtok(temp, "->");
        token2 = strtok(NULL, "->");
        strcpy(rules[i].left, token1);
        strcpy(rules[i].right, token2);
    }

    // User input for the input string
    printf("\nEnter the input string: ");
    scanf("%s", input);
    printf("Stack\tInput\tAction\n");
    i = 0;
    while (1)
    {
        if (i < strlen(input))
        {
            ch[0] = input[i];
            ch[1] = '\0';
            i++;
            strcat(stack, ch);
            printf("%s\t", stack);
            for (int k = i; k < strlen(input);
            k++){
                printf("%c", input[k]);
            }
            printf("\tShift %s\n", ch);
        }
        for (j = 0; j < rule_count; j++)
        {
            substring = strstr(stack,
             rules[j].right);
            if (substring != NULL)
            {
                stack_length = strlen(stack);
                substring_length = strlen
                (substring);
                stack_top = stack_length -
                substring_length;
                stack[stack_top] = '\0';
```



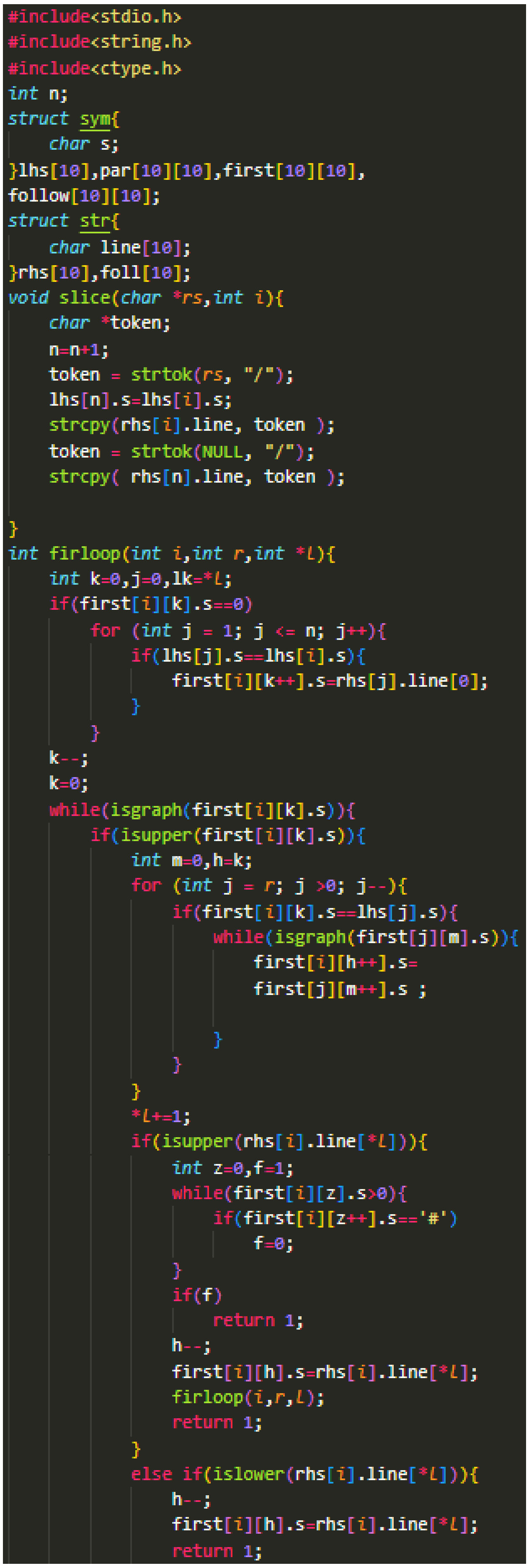
```
                strcat(stack, rules[j].left);
                printf("%s\t", stack);
                for (int k = i; k < strlen
                (input); k++)
                {
                    printf("%c", input[k]);
                }
                printf("\tReduce %s->%s\n",
                rules[j].left, rules[j].right);
                j = -1;
            }
        }
        if (strcmp(stack, rules[0].left) == 0
        && i == strlen(input))
        {
            printf("\nAccepted\n");
            break;
        }
        if (i == strlen(input))
        {
            printf("\nNot Accepted\n");
            break;
        }
    }
    return 0;
}
```

OUTPUT

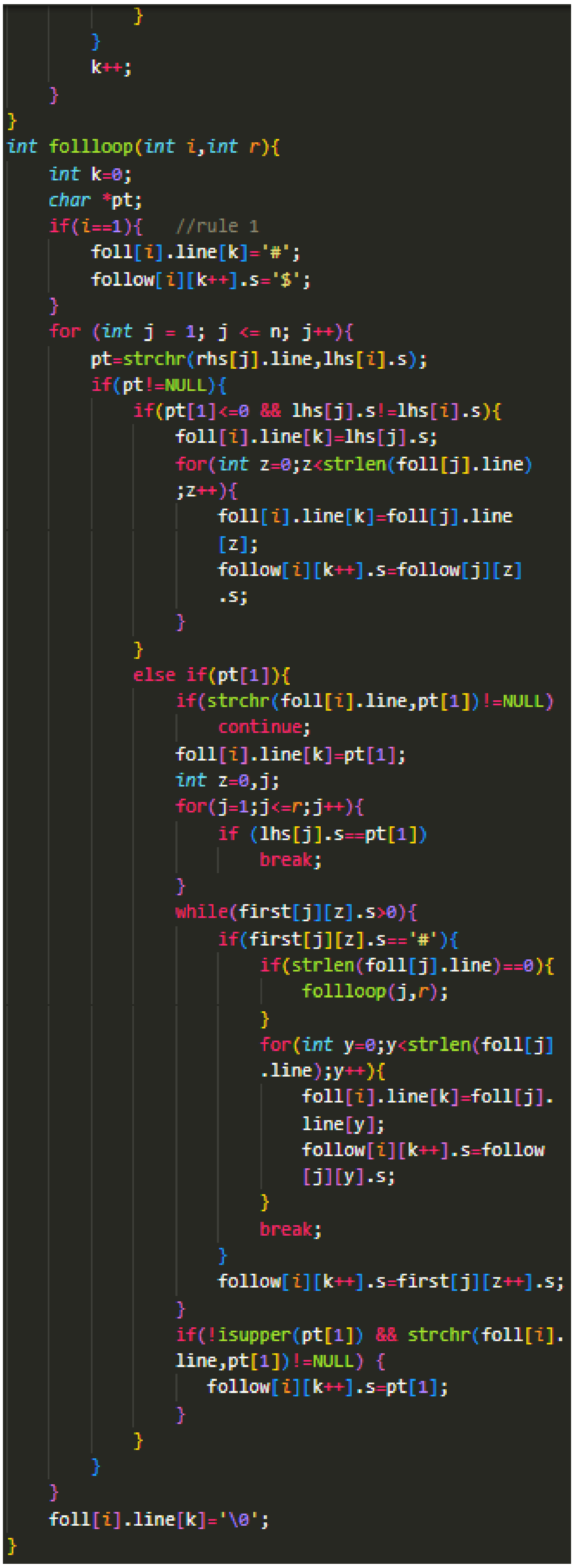


```
Enter the number of production rules: 5

Enter the production rules (in the form 'left->right'):
E->E+T
E->T
T->T*F
T->F
F->i

Enter the input string: i+i+i
Stack   Input   Action
i       +i+i    Shift i
F       +i+i    Reduce F->i
T       +i+i    Reduce T->F
E       +i+i    Reduce E->T
E+      i+i     Shift +
E+i     +i      Shift i
E+F     +i      Reduce F->i
E+T     +i      Reduce T->F
E       +i      Reduce E->E+T
E+      i       Shift +
E+i             Shift i
E+F             Reduce F->i
E+T             Reduce T->F
E               Reduce E->E+T

Accepted
```

CODE(Expt6)

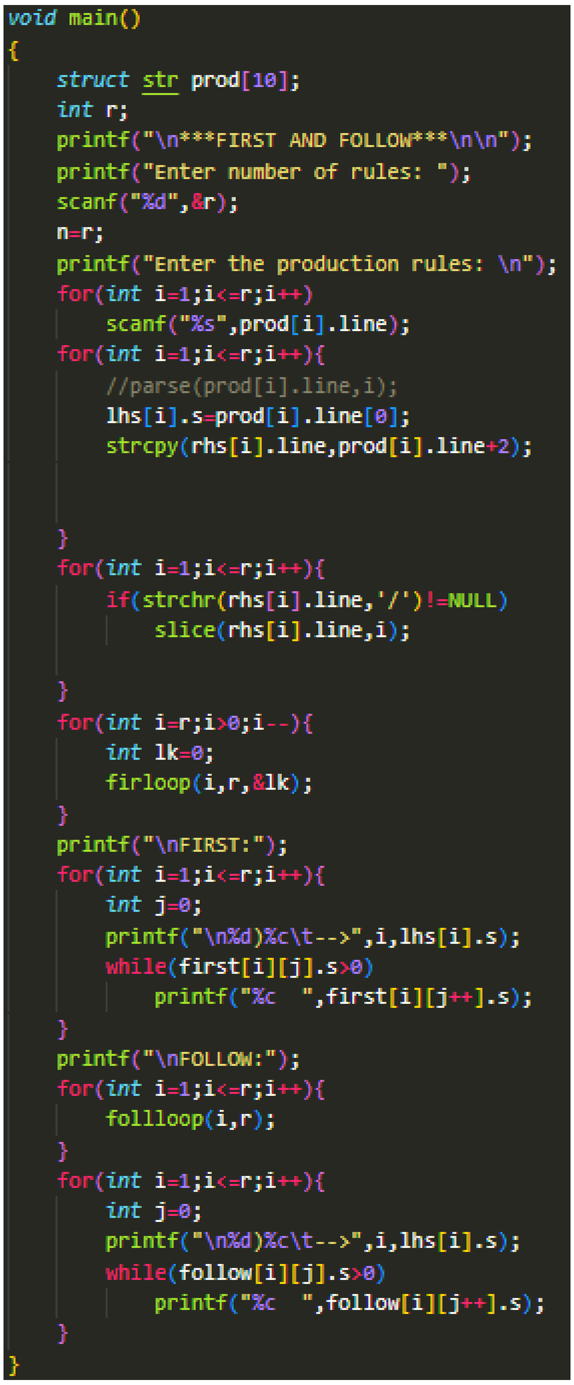


```c
#include<stdio.h>
#include<string.h>
#include<ctype.h>
int n;
struct sym{
    char s;
}lhs[10],par[10][10],first[10][10],
follow[10][10];
struct str{
    char line[10];
}rhs[10],foll[10];
void slice(char *rs,int i){
    char *token;
    n=n+1;
    token = strtok(rs, "/");
    lhs[n].s=lhs[i].s;
    strcpy(rhs[i].line, token );
    token = strtok(NULL, "/");
    strcpy( rhs[n].line, token );

}
int firloop(int i,int r,int *l){
    int k=0,j=0,lk=*l;
    if(first[i][k].s==0)
        for (int j = 1; j <= n; j++){
            if(lhs[j].s==lhs[i].s){
                first[i][k++].s=rhs[j].line[0];
            }
        }
    k--;
    k=0;
    while(isgraph(first[i][k].s)){
        if(isupper(first[i][k].s)){
            int m=0,h=k;
            for (int j = r; j >0; j--){
                if(first[i][k].s==lhs[j].s){
                    while(isgraph(first[j][m].s)){
                        first[i][h++].s=
                        first[j][m++].s ;

                    }
                }
            }
            *l+=1;
            if(isupper(rhs[i].line[*l])){
                int z=0,f=1;
                while(first[i][z].s>0){
                    if(first[i][z++].s=='#')
                        f=0;
                }
                if(f)
                    return 1;
                h--;
                first[i][h].s=rhs[i].line[*l];
                firloop(i,r,l);
                return 1;
            }
            else if(islower(rhs[i].line[*l])){
                h--;
                first[i][h].s=rhs[i].line[*l];
                return 1;
```

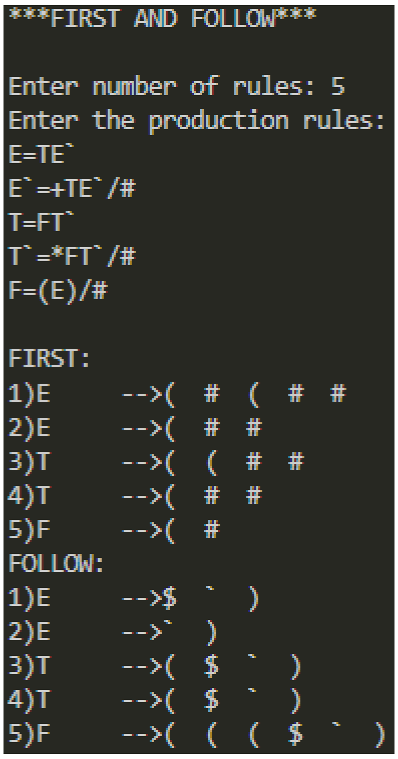


```c
                    }
                }
                k++;
            }
}
int follloop(int i,int r){
    int k=0;
    char *pt;
    if(i==1){   //rule 1
        foll[i].line[k]='#';
        follow[i][k++].s='$';
    }
    for (int j = 1; j <= n; j++){
        pt=strchr(rhs[j].line,lhs[i].s);
        if(pt!=NULL){
            if(pt[1]<=0 && lhs[j].s!=lhs[i].s){
                foll[i].line[k]=lhs[j].s;
                for(int z=0;z<strlen(foll[j].line)
                ;z++){
                    foll[i].line[k]=foll[j].line
                    [z];
                    follow[i][k++].s=follow[j][z]
                    .s;
                }
            }
            else if(pt[1]){
                if(strchr(foll[i].line,pt[1])!=NULL)
                    continue;
                foll[i].line[k]=pt[1];
                int z=0,j;
                for(j=1;j<=r;j++){
                    if (lhs[j].s==pt[1])
                        break;
                }
                while(first[j][z].s>0){
                    if(first[j][z].s=='#'){
                        if(strlen(foll[j].line)==0){
                            follloop(j,r);
                        }
                        for(int y=0;y<strlen(foll[j]
                        .line);y++){
                            foll[i].line[k]=foll[j].
                            line[y];
                            follow[i][k++].s=follow
                            [j][y].s;
                        }
                        break;
                    }
                    follow[i][k++].s=first[j][z++].s;
                }
                if(!isupper(pt[1]) && strchr(foll[i].
                line,pt[1])!=NULL) {
                   follow[i][k++].s=pt[1];
                }
            }
        }
    }
    foll[i].line[k]='\0';
}
```

OUTPUT(Expt6)



```c
void main()
{
    struct str prod[10];
    int r;
    printf("\n***FIRST AND FOLLOW***\n\n");
    printf("Enter number of rules: ");
    scanf("%d",&r);
    n=r;
    printf("Enter the production rules: \n");
    for(int i=1;i<=r;i++)
        scanf("%s",prod[i].line);
    for(int i=1;i<=r;i++){
        //parse(prod[i].line,i);
        lhs[i].s=prod[i].line[0];
        strcpy(rhs[i].line,prod[i].line+2);


    }
    for(int i=1;i<=r;i++){
        if(strchr(rhs[i].line,'/')!=NULL)
            slice(rhs[i].line,i);

    }
    for(int i=r;i>0;i--){
        int lk=0;
        firloop(i,r,&lk);
    }
    printf("\nFIRST:");
    for(int i=1;i<=r;i++){
        int j=0;
        printf("\n%d)%c\t-->",i,lhs[i].s);
        while(first[i][j].s>0)
            printf("%c  ",first[i][j++].s);
    }
    printf("\nFOLLOW:");
    for(int i=1;i<=r;i++){
        follloop(i,r);
    }
    for(int i=1;i<=r;i++){
        int j=0;
        printf("\n%d)%c\t-->",i,lhs[i].s);
        while(follow[i][j].s>0)
            printf("%c  ",follow[i][j++].s);
    }
}
```



```
***FIRST AND FOLLOW***

Enter number of rules: 5
Enter the production rules:
E=TE`
E`=+TE`/#
T=FT`
T`=*FT`/#
F=(E)/#

FIRST:
1)E     -->(  #  (  #  #
2)E     -->(  #  #
3)T     -->(  (  #  #
4)T     -->(  #  #
5)F     -->(  #
FOLLOW:
1)E     -->$  `  )
2)E     -->`  )
3)T     -->(  $  `  )
4)T     -->(  $  `  )
5)F     -->(  (  (  $  `  )
```