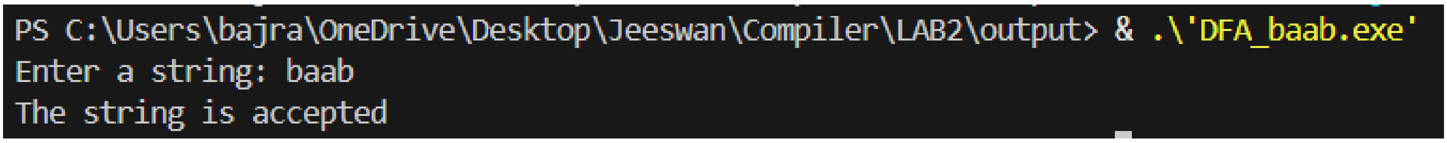
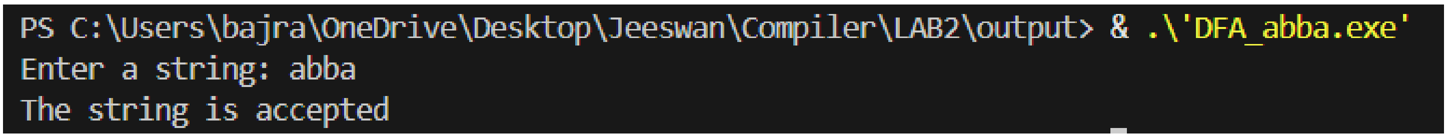
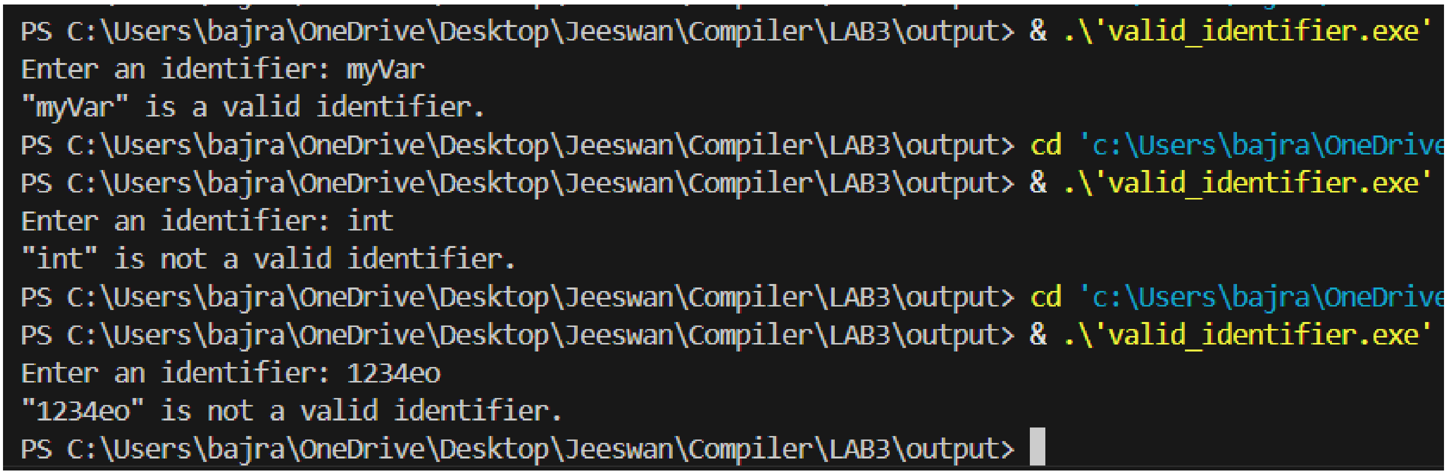
# Table of Content



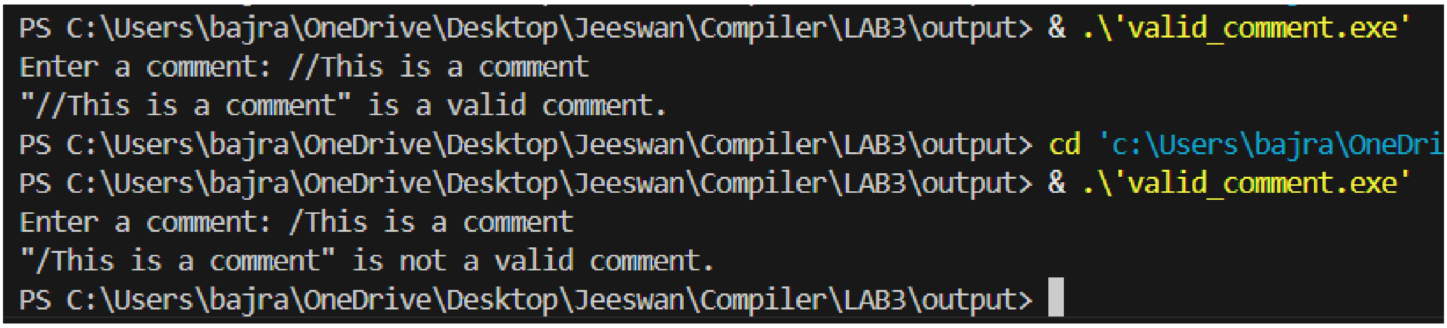
```
PS C:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler\LAB2\output> & .\'DFA_baab.exe'
Enter a string: baab
The string is accepted
```



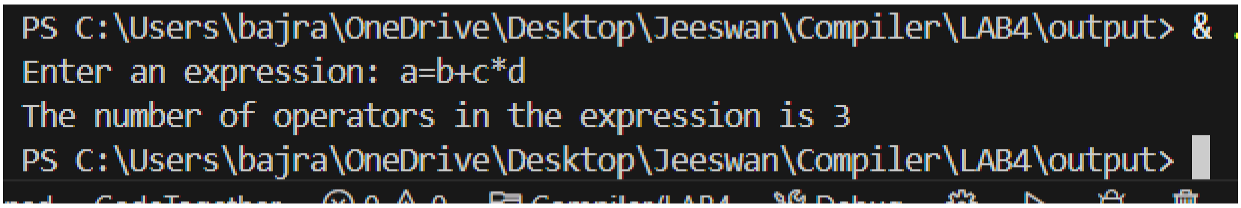
```
PS C:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler\LAB2\output> & .\'DFA_abba.exe'
Enter a string: abba
The string is accepted
```



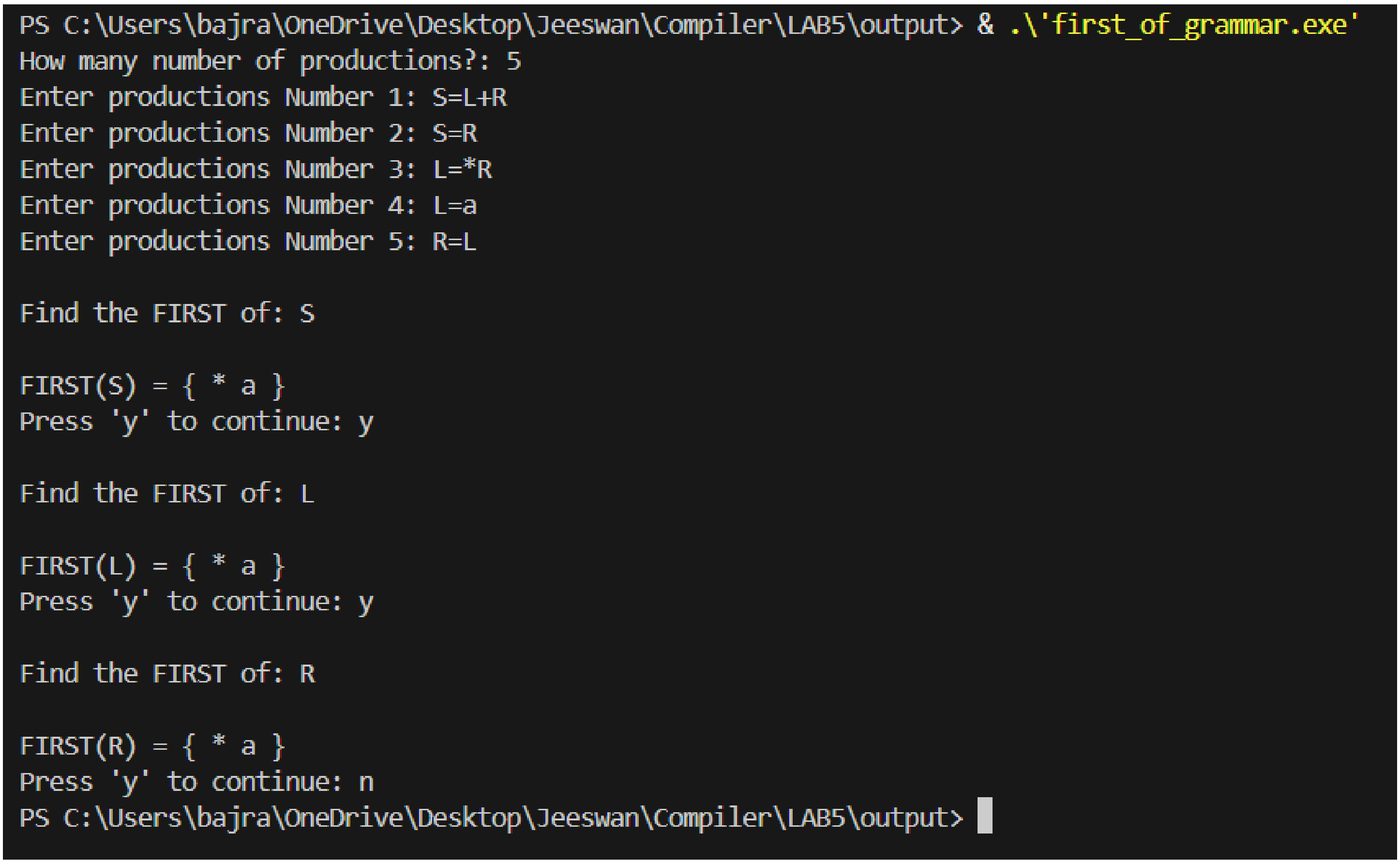
```
PS C:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler\LAB3\output> & .\'valid_identifier.exe'
Enter an identifier: myVar
"myVar" is a valid identifier.
PS C:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler\LAB3\output> cd 'c:\Users\bajra\OneDrive
PS C:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler\LAB3\output> & .\'valid_identifier.exe'
Enter an identifier: int
"int" is not a valid identifier.
PS C:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler\LAB3\output> cd 'c:\Users\bajra\OneDrive
PS C:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler\LAB3\output> & .\'valid_identifier.exe'
Enter an identifier: 1234eo
"1234eo" is not a valid identifier.
PS C:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler\LAB3\output>
```



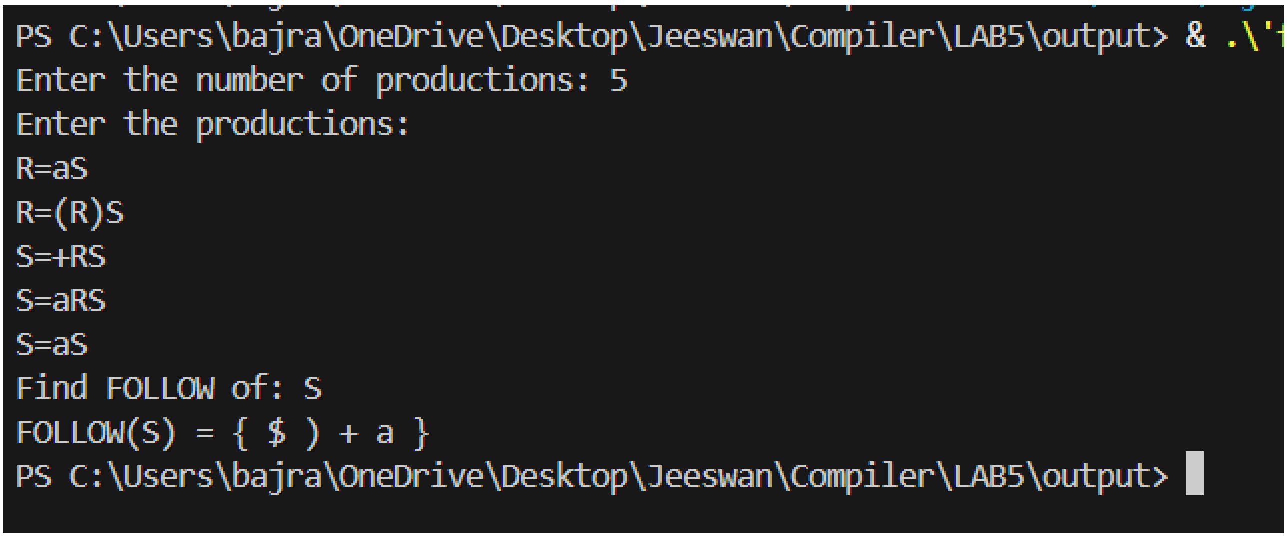
```
PS C:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler\LAB3\output> & .\'valid_comment.exe'
Enter a comment: //This is a comment
"//This is a comment" is a valid comment.
PS C:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler\LAB3\output> cd 'c:\Users\bajra\OneDri
PS C:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler\LAB3\output> & .\'valid_comment.exe'
Enter a comment: /This is a comment
"/This is a comment" is not a valid comment.
PS C:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler\LAB3\output>
```



```
PS C:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler\LAB4\output> &
Enter an expression: a=b+c*d
The number of operators in the expression is 3
PS C:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler\LAB4\output>
```

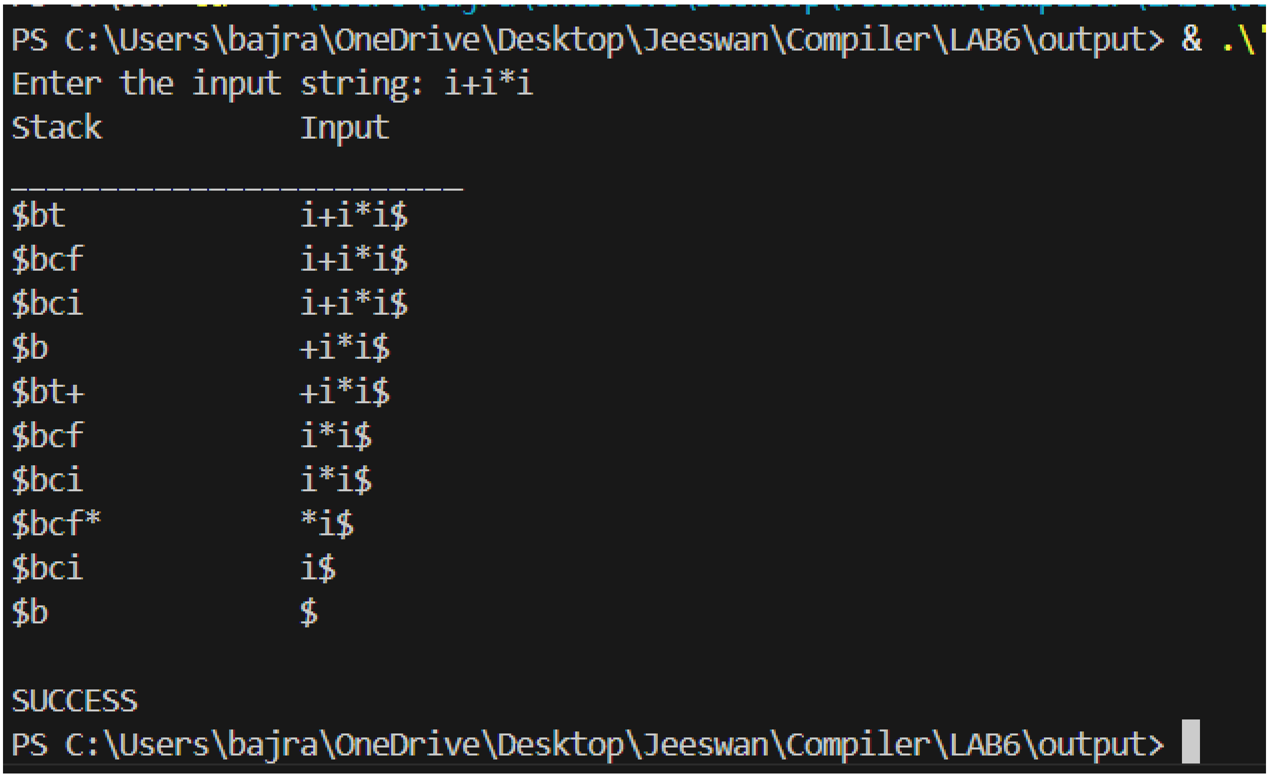


```
PS C:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler\LAB5\output> & .\'first_of_grammar.exe'
How many number of productions?: 5
Enter productions Number 1: S=L+R
Enter productions Number 2: S=R
Enter productions Number 3: L=*R
Enter productions Number 4: L=a
Enter productions Number 5: R=L

Find the FIRST of: S

FIRST(S) = { * a }
Press 'y' to continue: y

Find the FIRST of: L

FIRST(L) = { * a }
Press 'y' to continue: y

Find the FIRST of: R

FIRST(R) = { * a }
Press 'y' to continue: n
PS C:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler\LAB5\output>
```

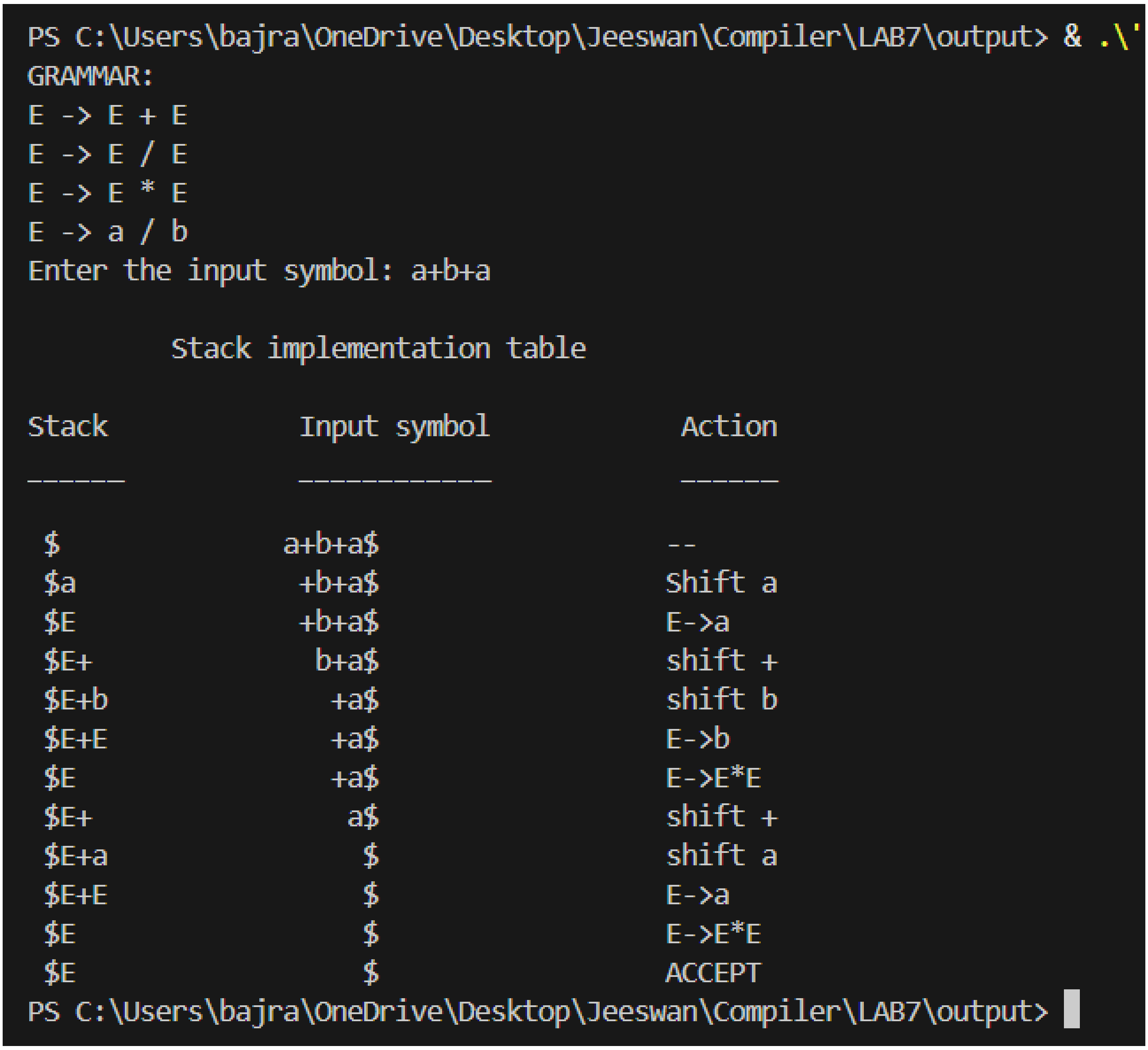


```
PS C:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler\LAB5\output> & .\'
Enter the number of productions: 5
Enter the productions:
R=aS
R=(R)S
S=+RS
S=aRS
S=aS
Find FOLLOW of: S
FOLLOW(S) = { $ ) + a }
PS C:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler\LAB5\output>
```

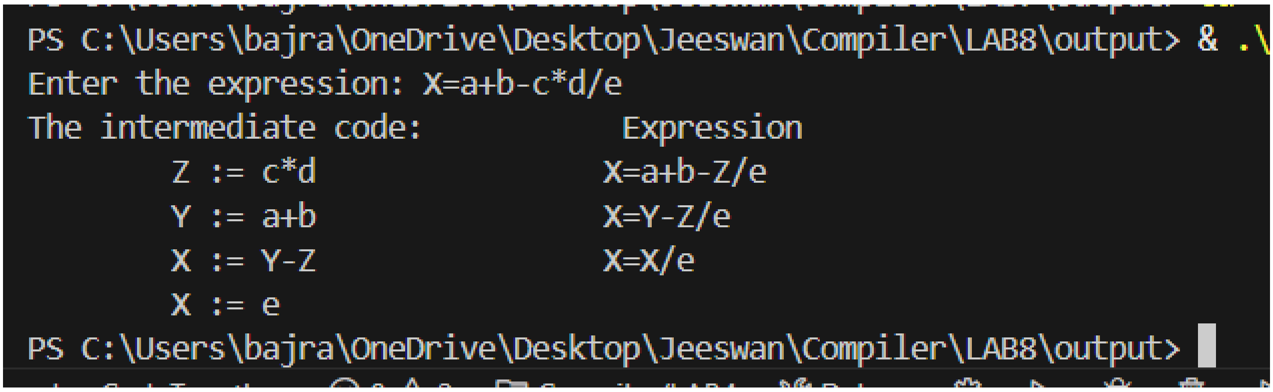
Output:



```
PS C:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler\LAB6\output> & .\
Enter the input string: i+i*i
Stack               Input

_________________________
$bt                 i+i*i$
$bcf                i+i*i$
$bci                i+i*i$
$b                  +i*i$
$bt+                +i*i$
$bcf                i*i$
$bci                i*i$
$bcf*               *i$
$bci                i$
$b                  $

SUCCESS
PS C:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler\LAB6\output>
```

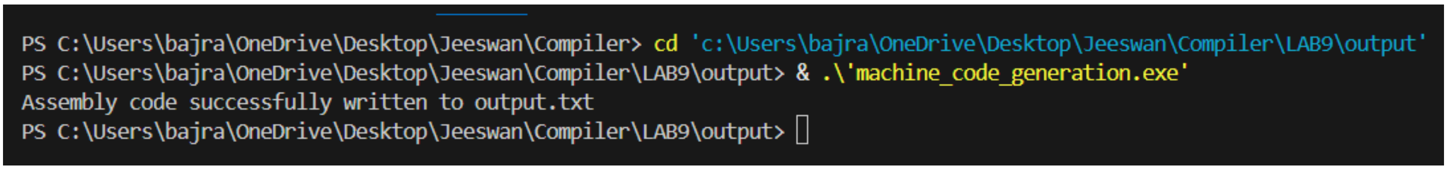


```
PS C:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler\LAB7\output> & .\'
GRAMMAR:
E -> E + E
E -> E / E
E -> E * E
E -> a / b
Enter the input symbol: a+b+a

         Stack implementation table

Stack                Input symbol            Action
------               ------------            ------

 $                  a+b+a$                  --
 $a                  +b+a$                  Shift a
 $E                  +b+a$                  E->a
 $E+                  b+a$                  shift +
 $E+b                  +a$                  shift b
 $E+E                  +a$                  E->b
 $E                    +a$                  E->E*E
 $E+                    a$                  shift +
 $E+a                    $                  shift a
 $E+E                    $                  E->a
 $E                      $                  E->E*E
 $E                      $                  ACCEPT
PS C:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler\LAB7\output> 
```

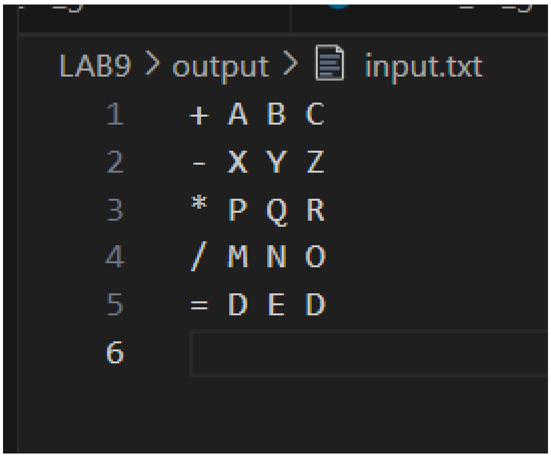


```
PS C:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler\LAB8\output> & .\
Enter the expression: X=a+b-c*d/e
The intermediate code:          Expression
        Z := c*d               X=a+b-Z/e
        Y := a+b               X=Y-Z/e
        X := Y-Z               X=X/e
        X := e
PS C:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler\LAB8\output> 
```

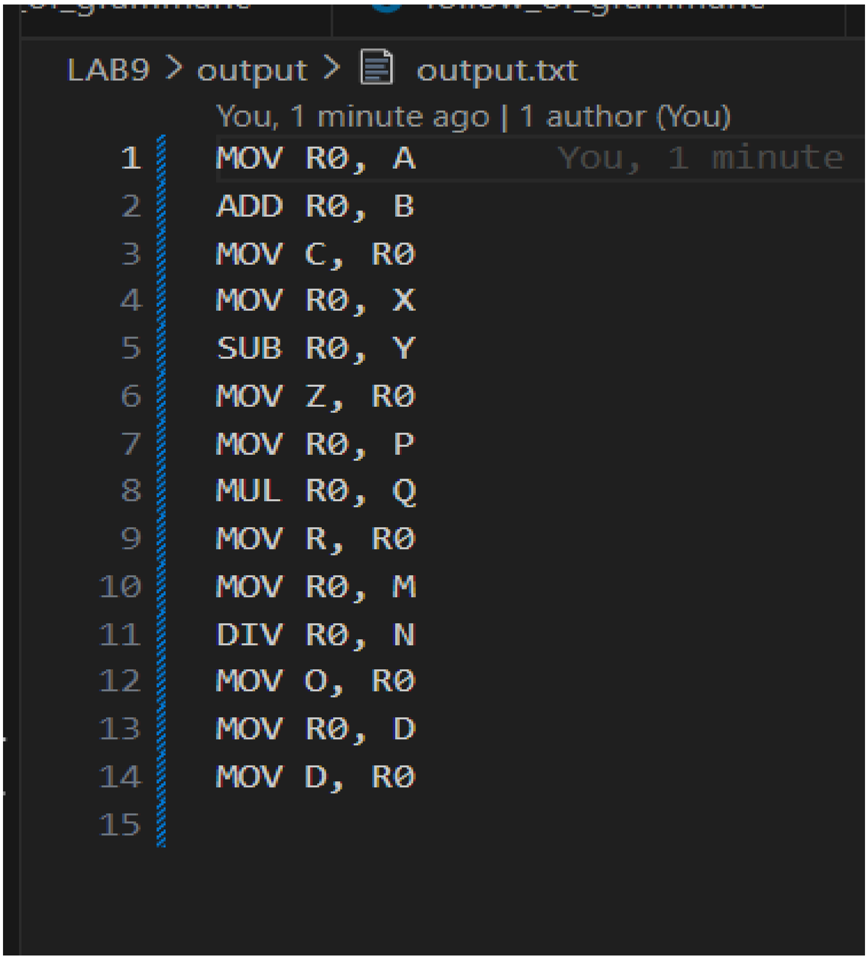
Output:



```
PS C:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler> cd 'c:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler\LAB9\output'
PS C:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler\LAB9\output> & .\'machine_code_generation.exe'
Assembly code successfully written to output.txt
PS C:\Users\bajra\OneDrive\Desktop\Jeeswan\Compiler\LAB9\output> 
```



```
+ A B C
- X Y Z
* P Q R
/ M N O
= D E D

```


LAB9 > output > output.txt
You, 1 minute ago | 1 author (You)
You, 1 minute

```
MOV R0, A
ADD R0, B
MOV C, R0
MOV R0, X
SUB R0, Y
MOV Z, R0
MOV R0, P
MUL R0, Q
MOV R, R0
MOV R0, M
DIV R0, N
MOV O, R0
MOV R0, D
MOV D, R0

```