

Table of Content

S. No.	Topic	Date	Remark
1.	Write a program to DFA that accept string. a. baab b. abba	February 16, 2025	
2.	2.1. Write a program to check valid identifier. 2.2. Write a program to check valid comment or not.	February 16, 2025	
3.	Write a program to count number of operators used in given input. $a=b+c*d$	February 16, 2025	
4.	3.1. Write a program to find the first of given grammar. $S \rightarrow L+R$ $S \rightarrow R$ $L \rightarrow *R$ $L \rightarrow a$ $R \rightarrow L$ 3.2. Write a program to find the follow of the given grammar. $R \rightarrow aS$ $R \rightarrow (R)S$ $S \rightarrow +RS$ $S \rightarrow aRS$ $S \rightarrow aS$	February 16, 2025	
5.	Write a program for construction of LL(1) Parser.	February 16, 2025	
6.	Write a program to implement shift reduce parsing. $E \rightarrow E+E$ $E \rightarrow E/E$ $E \rightarrow E * E$ $E \rightarrow a/b$ Input symbol $a+b+a$	February 16, 2025	
7.	Write a program to implement intermediate code generation. $X=a+b-c*d/e$	February 16, 2025	
8.	Write a program to implement machine code generation.	February 16, 2025	

```
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\OneDrive'  
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> & .\'t
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:

$E \rightarrow E + E$

$E \rightarrow E / E$

$E \rightarrow E * E$

$E \rightarrow a / b$

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\$	$E \rightarrow a$
\$E+	b+a\$	shift +
\$E+b	+a\$	shift b
\$E+E	+a\$	$E \rightarrow b$
\$E	+a\$	$E \rightarrow E * E$
\$E+	a\$	shift +
\$E+a	\$	shift a
\$E+E	\$	$E \rightarrow a$
\$E	\$	$E \rightarrow E * E$
\$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> █
```

LAB9 > output > input.txt

```
1  + A B C
2  - X Y Z
3  * P Q R
4  / M N O
5  = D E D
6  █
```

LAB9 > output > output.txt

```
You, 1 minute ago | 1 author (You)
1  MOV R0, A      You, 1 minute
2  ADD R0, B
3  MOV C, R0
4  MOV R0, X
5  SUB R0, Y
6  MOV Z, R0
7  MOV R0, P
8  MUL R0, Q
9  MOV R, R0
10 MOV R0, M
11 DIV R0, N
12 MOV O, R0
13 MOV R0, D
14 MOV D, R0
15 █
```

