Practical – 1

Aim:

Output:

1. Input String : bbbb

```
PS E:\Sem-6\Compiler Construction\Practical_Lab\Practical_1> & .\"Practical_1.exe"
Enter a string: bbbb
Validation result: Invalid
```

2. Input String : aaa

```
PS E:\Sem-6\Compiler Construction\Practical_Lab\Practical_1> & .\"Practical_1.exe"
Enter a string: aaa
Validation result: Invalid
```

3. Input String : baaabb

```
PS E:\Sem-6\Compiler Construction\Practical_Lab\Practical_1> & .\"Practical_1.exe"
Enter a string: baaabb
Validation result: Invalid
```

4. Input String : aaaaabb

```
PS E:\Sem-6\Compiler Construction\Practical_Lab\Practical_1> & .\"Practical_1.exe"
Enter a string: aaaaabb
Validation result: Valid
```

5. Input String : abb

```
PS E:\Sem-6\Compiler Construction\Practical_Lab\Practical_1> & .\"Practical_1.exe"
Enter a string: abb
Validation result: Valid
```

Conclusion:

Practical – 2

Aim:

Output:

```
PS E:\Sem-6\Compiler Construction\Practical_Lab\Practical_2> & .\"Practicacl_2.exe"
Number of input symbols : 2
Input Symbols : a b
Enter number of states : 4
Initial state : 1
number of accepting states : 1
Accepting states : 2
Transition Table :
1 to 0 -> 2
1 to 1 -> 3

2 to 0 -> 1
2 to 1 -> 4

3 to 0 -> 4
3 to 1 -> 1

4 to 0 -> 3
4 to 1 -> 2

Enter String : abbabab
Valid String
PS E:\Sem-6\Compiler Construction\Practical_Lab\Practical_2>
```

Conclusion:

Practical – 3

Aim:

Output:

```
Identifier: stdio.h
Operator: >
Identifier: int
Identifier: main
Punctuation: (
Punctuation: )
Punctuation: {
Identifier: int
Identifier: number1
Operator: =
Constant: 42
Punctuation: ;
Identifier: float
Identifier: pi
Operator: =
Constant: 3.14159
Punctuation: ;
Identifier: char
Identifier: message
Punctuation: [
Punctuation: ]
Operator: =
String: "Hello, World!"
Punctuation: ;
Operator: /
Operator: /
Identifier: Testing
Identifier: operators
Identifier: int
Identifier: sum
Operator: =
Identifier: number1
Operator: +
Constant: 10
Punctuation: ;
Identifier: int
Identifier: difference
Operator: =
Identifier: sum
Operator: -
Constant: 5
Punctuation: ;
Identifier: float
Identifier: product
Operator: =
Identifier: pi
Operator: *
Constant: 2
Punctuation: ;
Operator: /
Operator: /
Identifier: Testing
Identifier: invalid
Identifier: identifiers
Identifier: and
Identifier: constants
Identifier: int
Operator: =
Constant: 456
Punctuation: ;
Identifier: float
Operator: =
Punctuation: ;
Identifier: float
Operator: =
Constant: 789
Punctuation: ;
Identifier: if
Punctuation: (
Identifier: sum
Operator: >
Constant: 50
Punctuation: )
Punctuation: {
Identifier: printf
Punctuation: (
String: "Sum is greater than 50\n"
Punctuation: )
Punctuation: ;
Punctuation: }
Identifier: else
Punctuation: {
Identifier: printf
Punctuation: (
String: "Sum is less than or equal to 50\n"
Punctuation: )
Punctuation: ;
Punctuation: }
Identifier: return
Constant: 0
Punctuation: ;
Punctuation: }

Total Tokens: 90
```

Conclusion:

Practical – 4

Aim:

Output:

1. Objective 1:

- **Input String: a1b22c3**

```
PS E:\Sem-6\Compiler Construction\Practical_Lab\Practical_4\file1> flex file.l
PS E:\Sem-6\Compiler Construction\Practical_Lab\Practical_4\file1> gcc lex.yy.c -o file
PS E:\Sem-6\Compiler Construction\Practical_Lab\Practical_4\file1> ./file
Enter String: a1b22c3
1
22
3
```

- **Input String: power operation -> 12 ** 3 = 1728**

```
PS E:\Sem-6\Compiler Construction\Practical_Lab\Practical_4\file1> ./file
Enter String: power operation -> 12 ** 3 = 1728
12
3
1728
```

- **Input String: You multiply 804569 with 1 then will be :**

```
PS E:\Sem-6\Compiler Construction\Practical_Lab\Practical_4\file1> ./file
Enter String: You multiply 804569 with 1 then will be :
804569
1
```

2. Objective 2:

- **Input String: This is charusat.**

```
PS E:\Sem-6\Compiler Construction\Practical_Lab\Practical_4\file2> ./file
Enter String: This is charusat.
This is University.
```

- **Input String: Charusat is in Anand district.**

```
PS E:\Sem-6\Compiler Construction\Practical_Lab\Practical_4\file2> ./file
Enter String: Charusat is in Anand district.
University is in Anand district.
```

- **Input String: Charusat , What is charusat?**

```
PS E:\Sem-6\Compiler Construction\Practical_Lab\Practical_4\file2> ./file
Enter String: Charusat , What is charusat?
University , What is University?
```

3. Objective 3:

- **Input String: This is charusat.**

```
PS E:\Sem-6\Compiler Construction\Practical_Lab\Practical_4\file3> ./file
Enter String: The 45 is odd number.

Characters : 21
Words : 5
Line : 1
```

- **Input String: This is charusat.**

```
PS E:\Sem-6\Compiler Construction\Practical_Lab\Practical_4\file3> ./file
Enter String: I want to calculate a number. The number of characters, words and lines.

All know that \n is ending character of line.
no of line : 3
no of words : 22
no of char : 119
```

- **Input String: This is charusat.**

```
PS E:\Sem-6\Compiler Construction\Practical_Lab\Practical_4\file3> ./file
Enter String: 45 + 89 =40
no of line : 1
no of words : 4
no of char : 11
```

4. Objective 4:

- **Input String: a@IT**

```
PS E:\Sem-6\Compiler Construction\Practical_Lab\Practical_4\file4> ./file
Enter Password : a@1T
Invalid Password.
```

- **Input String: Charusat@2024**

```
PS E:\Sem-6\Compiler Construction\Practical_Lab\Practical_4\file4> ./file
Enter Password : Charusat@2024
Invalid Password.
```

- **Input String: Charu$at@20#24**

```
PS E:\Sem-6\Compiler Construction\Practical_Lab\Practical_4\file4> ./file
Enter Password : Charu$at@20#24
Valid Password.
```

- **Input String: charu*sAT;22**

```
PS E:\Sem-6\Compiler Construction\Practical_Lab\Practical_4\file4> ./file
Enter Password : charu*sAT;22
Valid Password.
```

Conclusion:

Practical – 5

Aim:

Output:

```
PS E:\Sem-6\Compiler Construction\Practical_Lab\Practical_5> ./file .\temp.c
Keyword: int
Identifier : main
Punctuation : (
Punctuation : )

Punctuation : {

Keyword: int
Identifier : a
Operator: =
Integer Constant : 5
Punctuation : ,
Integer Constant : 7
Identifier : H
Punctuation : ;


Keyword: char
Identifier : b
Operator: =
Char Constant : 'x'
Punctuation : ;

Operator: /
Operator: *
Keyword: return

Identifier : value
Operator: *
Operator: /
```

```
Keyword: return
Identifier : a
Operator: +
Identifier : b
Punctuation : ;

Punctuation : }
PS E:\Sem-6\Compiler Construction\Practical_Lab\Practical_5>
```

Conclusion: