MARMARA UNIVERSITY

FACULTY OF ENGINEERING

CSE2260 – Principles of Programming Languages

PROJECT 1

Department: CENG
Due Date: 14/04/2023
Used Language: Java

	Dept	Student ID	Name Surname
1	CENG	150121004	Ahmet Arda Nalbant
2	CENG	150121036	Hasan Özeren
3	CENG	150121991	Niyazi Ozan Ateş

Purpose

The purpose of this project is implementing a lexical analyzer/scanner for a given specific programming language.

Main

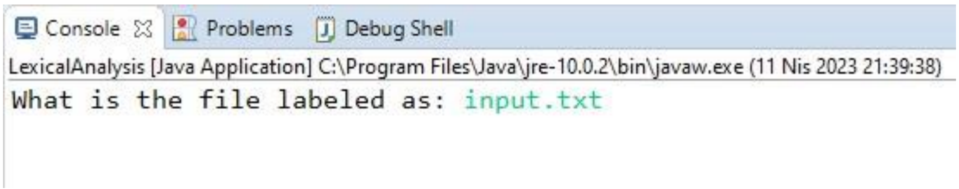
In the main function we read the file name and based on that we will given the right output. We do this by multiple if-else statements and a controlling part. First we check if the given input cosist of any errors. If there are no error, then we print the tokens based on their rules on both console and output text files. If a error is encountered then we print an error message at the first error that has been found. At last we close our files.

TokenizeInput

This function clears out the brackets, keywords and strings. Also, it helps to control our input files for any errors.

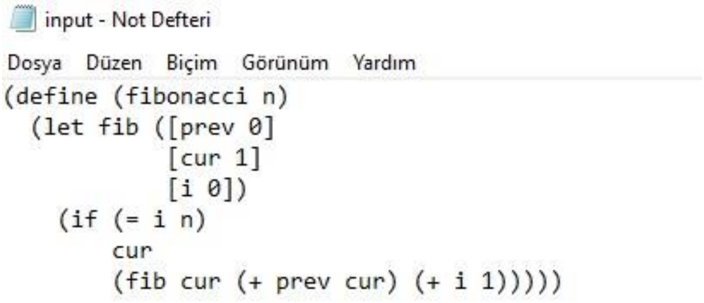
Usage of Program

When running the program it will ask you to put an input file name as follows:

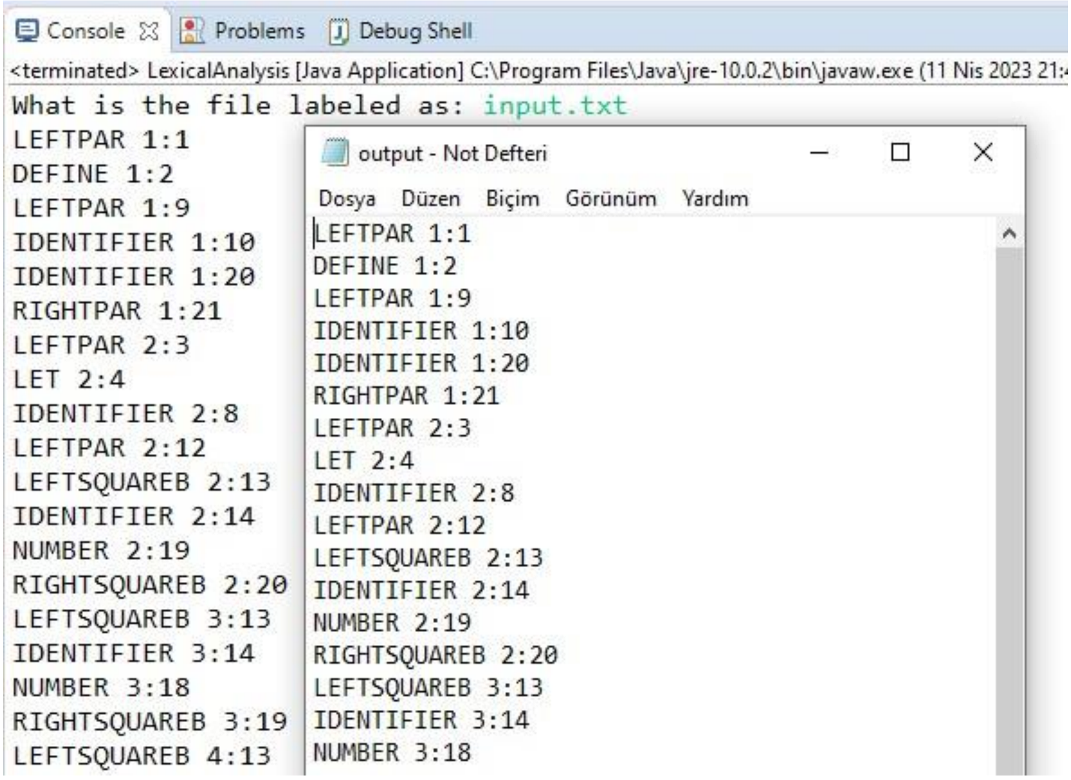


```
LexicalAnalysis [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (11 Nis 2023 21:39:38)
What is the file labeled as: input.txt
```

If it is the case that the user enters a valid input text file, then the program will run and gives you the correct output in both console and output text file. An example of the input and output is as follows:



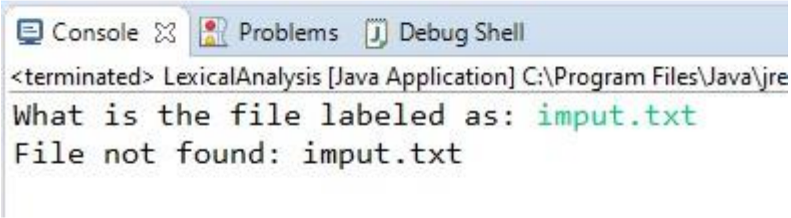
```
(define (fibonacci n)
  (let fib ([prev 0]
            [cur 1]
            [i 0])
    (if (= i n)
        cur
        (fib cur (+ prev cur) (+ i 1)))))
```



```
<terminated> LexicalAnalysis [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (11 Nis 2023 21:
What is the file labeled as: input.txt
LEFTPAR 1:1
DEFINE 1:2
LEFTPAR 1:9
IDENTIFIER 1:10
IDENTIFIER 1:20
RIGHTPAR 1:21
LEFTPAR 2:3
LET 2:4
IDENTIFIER 2:8
LEFTPAR 2:12
LEFTSQUAREB 2:13
IDENTIFIER 2:14
NUMBER 2:19
RIGHTSQUAREB 2:20
LEFTSQUAREB 3:13
IDENTIFIER 3:14
NUMBER 3:18
RIGHTSQUAREB 3:19
LEFTSQUAREB 4:13
```

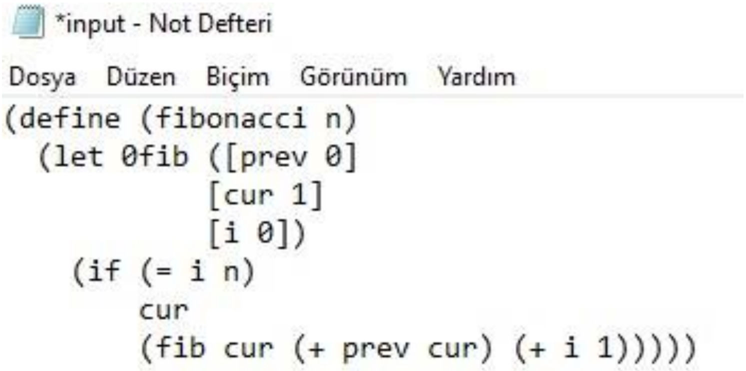
```
LEFTPAR 1:1
DEFINE 1:2
LEFTPAR 1:9
IDENTIFIER 1:10
IDENTIFIER 1:20
RIGHTPAR 1:21
LEFTPAR 2:3
LET 2:4
IDENTIFIER 2:8
LEFTPAR 2:12
LEFTSQUAREB 2:13
IDENTIFIER 2:14
NUMBER 2:19
RIGHTSQUAREB 2:20
LEFTSQUAREB 3:13
IDENTIFIER 3:14
NUMBER 3:18
```

If the file has not been found then you will get the output as follows:

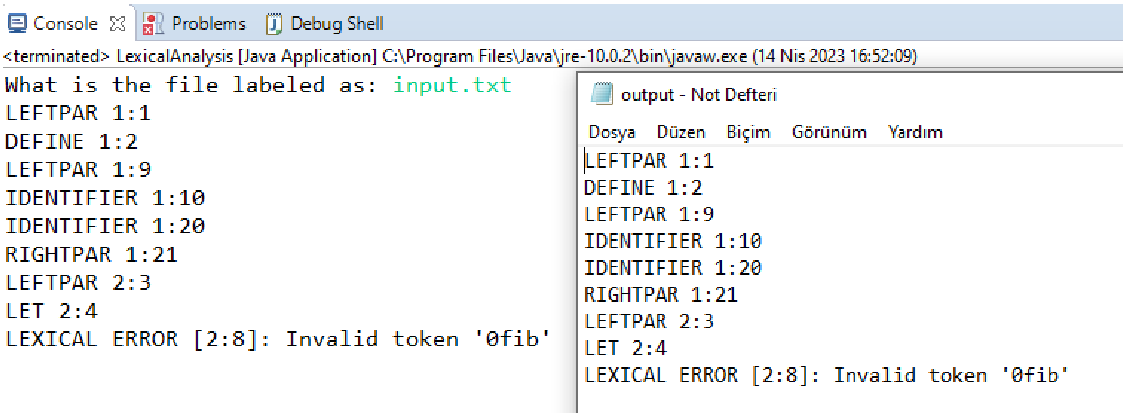


```
<terminated> LexicalAnalysis [Java Application] C:\Program Files\Java\jre
What is the file labeled as: imput.txt
File not found: imput.txt
```

If it is the case that the input has been found but there is an error then the input and output will be as follows:



```
(define (fibonacci n)
  (let 0fib ([prev 0]
             [cur 1]
             [i 0])
    (if (= i n)
        cur
        (fib cur (+ prev cur) (+ i 1)))))
```



```
<terminated> LexicalAnalysis [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (14 Nis 2023 16:52:09)
What is the file labeled as: input.txt
LEFTPAR 1:1
DEFINE 1:2
LEFTPAR 1:9
IDENTIFIER 1:10
IDENTIFIER 1:20
RIGHTPAR 1:21
LEFTPAR 2:3
LET 2:4
LEXICAL ERROR [2:8]: Invalid token '0fib'
```

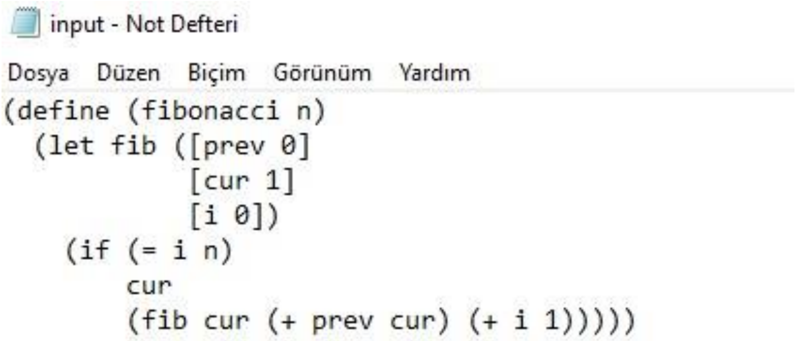
```
LEFTPAR 1:1
DEFINE 1:2
LEFTPAR 1:9
IDENTIFIER 1:10
IDENTIFIER 1:20
RIGHTPAR 1:21
LEFTPAR 2:3
LET 2:4
LEXICAL ERROR [2:8]: Invalid token '0fib'
```

Parts of The Assignment are Completed

Everything works as desired in all our input test cases. (comments, brackets, number literals, boolean literals, character literals, strings literals, keywords, identifiers)

Input/Output Examples

Example 1 (Correct Input):

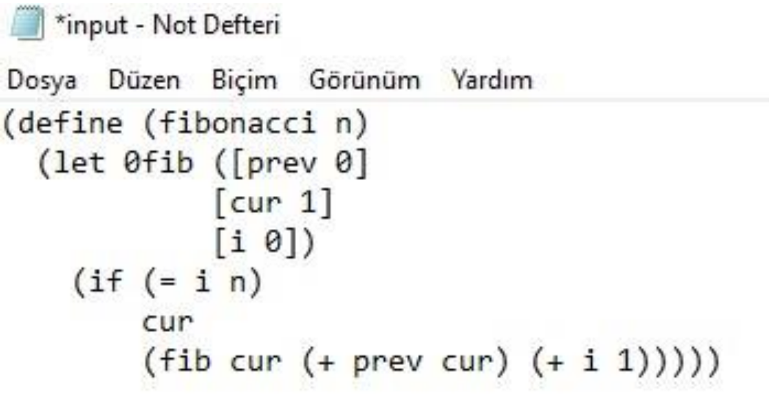


```
(define (fibonacci n)
  (let fib ([prev 0]
            [cur 1]
            [i 0])
    (if (= i n)
        cur
        (fib cur (+ prev cur) (+ i 1)))))
```

```
LEFTPAR 1:1
DEFINE 1:2
LEFTPAR 1:9
IDENTIFIER 1:10
IDENTIFIER 1:20
RIGHTPAR 1:21
LEFTPAR 2:3
LET 2:4
IDENTIFIER 2:8
LEFTPAR 2:12
LEFTSQUAREB 2:13
IDENTIFIER 2:14
NUMBER 2:19
RIGHTSQUAREB 2:20
LEFTSQUAREB 3:13
```

```
IDENTIFIER 3:14
NUMBER 3:18
RIGHTSQUAREB 3:19
LEFTSQUAREB 4:13
IDENTIFIER 4:14
NUMBER 4:16
RIGHTSQUAREB 4:17
RIGHTPAR 4:18
LEFTPAR 5:5
IF 5:6
LEFTPAR 5:9
IDENTIFIER 5:10
IDENTIFIER 5:12
IDENTIFIER 5:14
RIGHTPAR 5:15
IDENTIFIER 6:9
LEFTPAR 7:9
IDENTIFIER 7:10
IDENTIFIER 7:14
LEFTPAR 7:18
IDENTIFIER 7:19
IDENTIFIER 7:21
IDENTIFIER 7:26
RIGHTPAR 7:29
LEFTPAR 7:31
IDENTIFIER 7:32
IDENTIFIER 7:34
NUMBER 7:36
RIGHTPAR 7:37
RIGHTPAR 7:38
RIGHTPAR 7:39
RIGHTPAR 7:40
RIGHTPAR 7:41
```

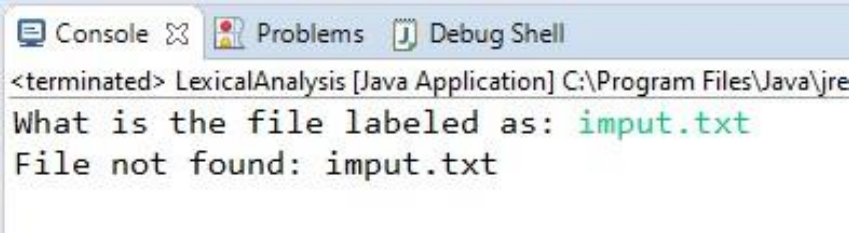
Example 2 (Incorrect Input):



```
(define (fibonacci n)
  (let 0fib ([prev 0]
             [cur 1]
             [i 0])
    (if (= i n)
        cur
        (fib cur (+ prev cur) (+ i 1)))))
```

```
LEFTPAR 1:1
DEFINE 1:2
LEFTPAR 1:9
IDENTIFIER 1:10
IDENTIFIER 1:20
RIGHTPAR 1:21
LEFTPAR 2:3
LET 2:4
LEXICAL ERROR [2:8]: Invalid token '0fib'
```

Example 3 (File Not Found):



<terminated> LexicalAnalysis [Java Application] C:\Program Files\Java\jre

```
What is the file labeled as: imput.txt
File not found: imput.txt
```