

Вміст файлу Input.js

```
if (x == 10) { return x + 1; } // This is a comment
Лістинг коду CharScanner.java
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

public class CharScanner {
    public static void main(String[] args) {
        String fileName = "input.js"; // Файл з JavaScript
КОДОМ
        StringBuilder content = new StringBuilder();

        try (BufferedReader br = new BufferedReader(new
FileReader(fileName))) {
            int c;
            while ((c = br.read()) != -1) {
                char character = (char) c;
                content.append(character);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }

        // Записуємо текст у новий файл для аналізу JavaScript
        try (FileWriter fileWriter = new
FileWriter("scanned_output.txt")) {
            fileWriter.write(content.toString());
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Вміст файлу lexer.js

```
const fs = require('fs');

class Lexer {
    constructor(input) {
        this.input = input;
        this.position = 0;
        this.tokens = [];
    }
}
```

```

isWhitespace(char) {
    return /\s/.test(char);
}

isDigit(char) {
    return /\d/.test(char);
}

isLetter(char) {
    return /[a-zA-Z_]/.test(char);
}

getNextChar() {
    return this.input[this.position];
}

advance() {
    this.position++;
}

addToken(type, value) {
    this.tokens.push({ type, value });
}

lex() {
    while (this.position < this.input.length) {
        let currentChar = this.getNextChar();

        if (this.isWhitespace(currentChar)) {
            this.advance();
            continue;
        }

        if (this.isDigit(currentChar)) {
            let number = "";
            while (this.isDigit(currentChar)) {
                number += currentChar;
                this.advance();
                currentChar = this.getNextChar();
            }
            this.addToken("NUMBER", number);
            continue;
        }
    }
}

```

```

        if (this.isLetter(currentChar)) {
            let identifier = "";
            while (this.isLetter(currentChar) ||
this.isDigit(currentChar)) {
                identifier += currentChar;
                this.advance();
                currentChar = this.getNextChar();
            }
            if (["if", "else", "for", "while",
"return"].includes(identifier)) {
                this.addToken("KEYWORD", identifier);
            } else {
                this.addToken("IDENTIFIER", identifier);
            }
            continue;
        }

        if (currentChar === "'" || currentChar === '"') {
            let quoteType = currentChar;
            let stringLiteral = "";
            this.advance();
            currentChar = this.getNextChar();
            while (currentChar !== quoteType) {
                stringLiteral += currentChar;
                this.advance();
                currentChar = this.getNextChar();
            }
            this.advance(); // Skip the closing quote
            this.addToken("STRING", stringLiteral);
            continue;
        }

        if ("+-*/=<>!".includes(currentChar)) {
            this.addToken("OPERATOR", currentChar);
            this.advance();
            continue;
        }

        if ("{}()[];,".includes(currentChar)) {
            this.addToken("SYMBOL", currentChar);
            this.advance();
            continue;
        }

```

```

        this.addToken("UNKNOWN", currentChar);
        console.error(`Lexical error: unrecognized symbol
${currentChar}`);
        this.advance();
    }

    return this.tokens;
}
}

fs.readFile('scanned_output.txt', 'utf8', (err, data) => {
    if (err) {
        console.error(err);
        return;
    }
    const lexer = new Lexer(data);
    const tokens = lexer.lex();
    console.log(tokens);
});

```

Вивід програми:

```

[
  { type: 'KEYWORD', value: 'if' },
  { type: 'SYMBOL', value: '(' },
  { type: 'IDENTIFIER', value: 'x' },
  { type: 'OPERATOR', value: '==' },
  { type: 'NUMBER', value: '10' },
  { type: 'SYMBOL', value: ')' },
  { type: 'SYMBOL', value: '{' },
  { type: 'KEYWORD', value: 'return' },
  { type: 'IDENTIFIER', value: 'x' },
  { type: 'OPERATOR', value: '+' },
  { type: 'NUMBER', value: '1' },
  { type: 'SYMBOL', value: ';' },
  { type: 'SYMBOL', value: '}' }
]

```

Висновки:

Компіляція та запуск CharScanner.java: зчитує текст з файлу input.js і зберігає його у scanned_output.txt.

Запуск lexer.js: зчитує текст з scanned_output.txt, виконує лексичний аналіз і виводить токени.