Node.js Streams Assignment: "File Magic Stream"

Objective: Introduce students to Node.js streams by creating a program that reads a large text file, processes its content, and writes the modified content into another file using streams.

Scenario:

You're working as a junior developer at a tech company that processes huge amounts of text data. Your manager asks you to build a small tool that reads data from a text file, processes it (e.g., converts text to uppercase), and writes the result to a new file—all while ensuring that the memory usage stays efficient. Streams to the rescue!

Steps:

1. Setup:

- Create a new project folder: file-magic-stream.
- Inside the folder, run npm init -y to initialize a new Node.js project.
- Install dependencies (if any): None needed for this task.

2. Assignment Tasks:

Task 1: Create a Large Input File

- Use Node.js to create a script (createLargeFile.js) that generates a file named input.txt with 1,000,000 lines of random text. Example content:

```
Line 1: Lorem ipsum...
Line 2: Node.js is awesome!
```

> Hint: Use fs.writeFileSync or fs.createWriteStream with a loop to generate content.

Task 2: Process File with Streams

- Create a new script (processFile.js) that:

1. Reads the input.txt file using a readable stream.
2. Transforms the data by converting all text to uppercase (use a transform stream).
3. Writes the transformed data to output.txt using a writable stream.

Task 3: Add Logging

- Enhance your program to log:
  - When the process starts and ends.
  - The size of the input and output files in bytes.

Bonus Challenges (Optional):

- Add another transform stream that filters out lines containing specific words (e.g., "random").
- Measure the processing time and log it.
- Handle errors in the stream pipeline gracefully.