HOMEWORK 1

BASICS. NODEJS FUNDAMENTAL THEORY

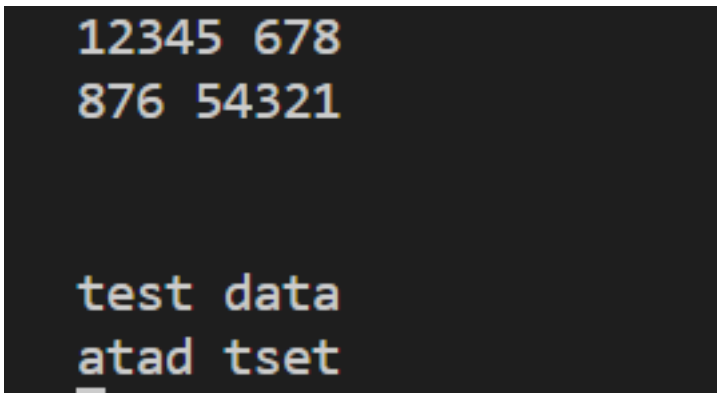
PREREQUISITES:

1. Install the latest **LTS** (Long Term Support) version of **Node.js** (https://nodejs.org/en/), by any available means (.exe, nvm, brew, etc.).
2. Check in the console (terminal) that the Node.js installation was done properly by running the following commands `node -v` or `node -version`.
3. Create a repo for your homework tasks on **Github** (https://github.com/) or **git.epam.com**.
4. Provide your mentor with the link to the repo and add read access permissions.
5. Create package.json by running the following commands `npm init` or `npm init -y`.
6. Install globally or locally **npm** package **nodemon** (https://github.com/remy/nodemon) to **dev dependency**.
7. Get ready to watch the lectures and do the homework tasks to study the basic principles and approaches of development server-side applications with **Node.js**.

TASK 1.1

Write a program which reads a string from the standard input stdin, reverses it and then writes it to the standard output stdout.

- The program should be started from **npm** script via **nodemon** (i.e. `npm run task1`).
- The program should be running in a stand-by mode and should not be terminated after the first-string processing.
- For example:



TASK 1.2

Write a program which should do the following:

- Read the content of **csv** file from `./csv` directory. Example: https://epa.ms/nodejs19-hw1-ex1
- Use the **csvtojson** package (https://github.com/Keyang/node-csvtojson) to convert **csv** file to **json** object.

- Write the **csv** file content to a new **txt** file.

  Use the following format: https://epa.ms/nodejs19-hw1-ex2.
- Do not load all the content of the **csv** file into RAM via stream (read/write file content line by line).
- In case of read/write errors, log them in the console.
- The program should be started via **npm script** using **nodemon** (i.e. `npm run task2`).

TASK 1.3

Rewrite the above-mentioned programs to use **babel** (https://babeljs.io/) and **ES6** modules.

EVALUATION CRITERIA

2. **Task 1.1** is fulfilled to the full extent.
3. **Task 1.2** is fulfilled to the full extent; the file is loaded fully into the RAM.
4. **Task 1.2** is fulfilled to the full extent; the file is not loaded fully in the RAM (pipeline method https://nodejs.org/api/stream.html#stream_stream_pipeline_streams_callback).
5. All the tasks are fulfilled to the full extent (**Task 1.1**, **Task 1.2**, **Task 1.3**).