# JS Advanced: Exam Preparation

Problems for exam preparation for the [“JavaScript Advanced” course @ SoftUni](https://softuni.bg/courses/javascript-advanced). Submit your solutions in the SoftUni judge system at <https://judge.softuni.bg/Contests/352/>.

## 03.Storm Watcher

Write a JS class that represents a meteorological station reading. Each reading has an **id**, **temperature**, **humidity**, **pressure** and **windSpeed** properties which are all numbers. The **ID** is auto assigned and **autoincremented** sequentially for each instance, while the rest of the properties are set trough the **constructor**.

In addition, the class must include a toString() method that **returns** a formatted string with a summary of the information kept inside the record and a **weather status**. The status is either **'Not stormy'** or **'Stormy'**, depending on the readings. For the weather to be **stormy**, **all** of these conditions must be met:

* **temperature** bellow 20
* **pressure** bellow 700 **OR** above 900
* **windSpeed** above 25

For **any other** conditions, the weather is **not stormy**. See the examples for more formatting details. Note each property is on a new line.

### Input / Output

Only valid data will be passed to the constructor. The **output** is expected as a **string**, **returned** by the toString() method of your class.

Depending on how you structure your code, submit just the **class** definition **as is**, **or** wrapped in an **IIFE** that **returns** the **class** definition.

### Examples

|  |  |
| --- | --- |
| Sample Input | Output |
| let record1 = new Record(32, 66, 760, 12);  console.log(record1.toString()); | Reading ID: 0  Temperature: 32\*C  Relative Humidity: 66%  Pressure: 760hpa  Wind Speed: 12m/s  Weather: Not stormy |
| let record2 = new Record(10, 40, 680, 30);  console.log(record2.toString()); | Reading ID: 1  Temperature: 10\*C  Relative Humidity: 40%  Pressure: 680hpa  Wind Speed: 30m/s  Weather: Stormy |