

Task 3 [15 points]

Alice is another student who wants to learn prototype chain and function redefinition in JavaScript. She finds that JavaScript is very flexible. For example, she can redefine the `sin` functions under the system-provided `Math` object below so that `sin(x)` becomes `2 * sin (x)`:

```
let oldSin = Math.sin;
Math.sin = function (x) {
  return 2 * oldSin(x);
}
```

Now, she wrote the following code and tested it on the empty tab of a web browser console.

```
let myObj = {prop1: 3, prop2: 5, prop3: 4};
console.log("Results:" + myObj);
```

She finds that the browser outputs the following text on the console:

Results:[object Object]

However, she visited another website and typed the same code into the console. She found the following output:

Results:{prop1: 3, prop3: 4, prop2: 5}

So the same code outputs each key-value pair of the object and **sort** them based on the value.

Question Can you help Alice explain why this happens by completing the code in the `index.js` file inside this folder?

Note that you cannot hard-code any strings in your implementation. The output format should follow the example.

Please complete this task in the `index.js` file inside this folder. Do **not** write the answer in this README file.