

## JS 磁盘容量排序

题目描述：磁盘的容量单位常用的有M，G，T这三个等级，它们之间的换算关系为1T = 1024G，1G = 1024M，现在给定n块磁盘的容量，请对它们按从小到大的顺序进行稳定排序，例如给定5块盘的容量，1T，20M，3G，10G6T，3M12G9M排序后的结果为20M，3G，3M12G9M，1T，10G6T。注意单位可以重复出现，上述3M12G9M表示的容量即为3M+12G+9M，和12M12G相等。

输入描述：输入第一行包含一个整数n( $2 \leq n \leq 100$ )，表示磁盘的个数，接下的n行，每行一个字符串(长度大于2，小于30)，表示磁盘的容量，由一个或多个格式为mv的子串组成，其中m表示容量大小，v表示容量单位，例如20M，1T，30G，10G6T，3M12G9M。

磁盘容量m的范围为1到1024的正整数，容量单位v的范围只包含题目中提到的M，G，T三种，换算关系如题目描述。

输出描述：输出n行，表示n块磁盘容量排序后的结果。

### 示例

#### 示例1

输入：3

1G  
2G  
1024M

输出：1G

1024M  
2G

说明：1G和1024M容量相等，稳定排序要求保留它们原来的相对位置，故1G在1024M之前

#### 示例2

输入：3

2G4M  
3M2G  
1T

输出：3M2G

2G4M  
1T

说明：1T的容量大于2G4M，2G4M的容量大于3M2G

```
1  const rl = require("readline").createInterface({ input: process.stdin });
2  var iter = rl[Symbol.asyncIterator]();
3  const readline = async () => (await iter.next()).value;
4
5  void (async function () {
6    // Write your code here
7    const arr = [];
8    while ((line = await readline())) {
9      let tokens = line.split(" ");
10     arr.push(tokens[0]);
11   }
12   arr.shift();
13   sortCap(arr).forEach((cap) => console.log(cap));
14 })();
15
16 function sortCap(arr) {
```

```
17     arr.sort((a, b) => calcCap(a) - calcCap(b));
18     return arr;
19 }
20
21 function calcCap(cap) {
22     let res = 0;
23     let tmp = [];
24
25     for (let i = 0; i < cap.length; i++) {
26         let curr = cap[i];
27         if (parseInt(curr) >= 0) {
28             tmp.push(curr);
29         } else {
30             if (curr === "M") {
31                 res += parseInt(tmp.join(""));
32             } else if (curr === "G") {
33                 res += parseInt(tmp.join("")) * 1024;
34             } else if (curr === "T") {
35                 res += parseInt(tmp.join("")) * 1024 * 1024;
36             } else {
37                 res = res;
38             }
39             tmp = [];
40         }
41     }
42     return res;
43 }
```