

## 为什么刷题？

---

刷题主要是算法学习者需要做的事情，很多为了快速实现项目的初学者一般都学习完语言的基础语法就要学习相关领域主流框架。



一般来说刷题主要语言都是C++、Python、Java，很少有人用js刷题。一方面，当具备算法的基础后，可以快速掌握js语言并且开发项目。另一方面使用js刷题流程比较繁琐，但是，js做题的好处是在初学期间，不需要考虑繁多的API接口，以及与html、js的交互设计，可以集中精力在js基础语法的灵活掌握上。

## js刷题输入输出模板

---



比如，给出n个数字，用空格分隔，然后输出原始数据。

```
1 let buf = "";
2
3 process.stdin.on("readable", function(){
4     let chunk = process.stdin.read();
5     if(chunk) buf += chunk.toString();
6 });
7
8 process.stdin.on("end", function(){
9     console.log(buf);
10 });
```

```

1 let buf = "";
2
3 process.stdin.on("readable", function(){
4     let chunk = process.stdin.read();
5     if(chunk) buf += chunk.toString();
6 });
7
8 process.stdin.on("end", function(){
9     console.log(buf);
10 });

```

运行

测试输入

1 2 3

输出:

1 2 3

Time: 35ms

Memory: 29MB

let buf = "";定义一个字符串，首先把输入的内容都转化成字符串，输入使用process.stdin.read()读取所有内容赋值给chunk变量，再用buf += chunk.toString()将chunk转化为字符串拼接到buf上。

会自动识别结束符，当输入完毕，读取过程结束。

然而，输入的数据是整数，输出的内容是字符串类型，那么如何分离出每个整数？

```

1 let [a, b, c] = buf.split(" ").map(x => {return parseInt(x)});

```

```

1 let buf = "";
2
3 process.stdin.on("readable", function(){
4     let chunk = process.stdin.read();
5     if(chunk) buf += chunk.toString();
6 });
7
8 process.stdin.on("end", function(){
9     let arr = buf.split(" ");
10    console.log(arr);
11 });

```

运行

测试输入

1 2 3

输出:

['1', '2', '3']

Time: 36ms

Memory: 29MB

用buf.split(" ")以空格分隔开每个原始数字，返回值是数组类型，但是数组里面是字符串类型的数字，因此，还要进一步将每个字符串类型的数字映射为整数类型。

公众号：黑猫编程

网址：<https://noi.hioier.co>

```
1 let arr = buf.split(" ").map(x => {return parseInt(x)});
```

```
1 let buf = "";
2
3 process.stdin.on("readable", function(){
4     let chunk = process.stdin.read();
5     if(chunk) buf += chunk.toString();
6 });
7
8 process.stdin.on("end", function(){
9     let arr = buf.split(" ").map(x => {return parseInt(x)});
10    console.log(arr);
11 });
```

运行

测试输入

1 2 3

输出:

[ 1, 2, 3 ]

Time: 36ms  
Memory: 30MB

## 计算线段长度

### 计算线段长度

#### 描述

已知线段的两个端点的坐标 $A(X_a, Y_a)$ ,  $B(X_b, Y_b)$ , 求线段AB的长度, 保留到小数点后3位。

#### 输入

第一行是两个实数 $X_a$ ,  $Y_a$ , 即A的坐标。

第二行是两个实数 $X_b$ ,  $Y_b$ , 即B的坐标。

输入中所有实数的绝对值均不超过10000。

#### 输出

一个实数, 即线段AB的长度, 保留到小数点后3位。

#### 输入样例 1

```
1 1
2 2
```

#### 输出样例 1

1.414

#### 提交信息

##### 题目信息

ID	206
时间限制	1000MS
内存限制	256MB
IO 类型	Standard IO
出题人	blackcat
难度	低
分数	100
标签	<a href="#">显示</a>

##### 统计

Details

AC WA

```
1 let buf = "";
2
3 process.stdin.on("readable", function(){
4     let chunk = process.stdin.read();
5     if(chunk) buf += chunk.toString();
6 });
7
8 process.stdin.on("end", function(){
9     let lines = buf.split("\n");
10    let [x1, y1] = lines[0].split(" ").map(x => {return parseFloat(x)});
11    let [x2, y2] = lines[1].split(" ").map(x => {return parseFloat(x)});
12
```

公众号: 黑猫编程

网址: <https://noi.hioqier.co>

```
13 let res = Math.sqrt((x1-x2)*(x1-x2) + (y1-y2)*(y1-y2));
14 console.log(res.toFixed(3));
15 });
```

因为输入内容为两行，且以换行符“\n”分隔，所以第一次使用split("\n")，得到一个二维数组。再对二维数组的0行、1行进行split和map转化成浮点型。

最后用勾股定理计算斜边长度，toFixed保留结果的小数位数。

## 小玉买文具

### 描述

班主任给小玉一个任务，到文具店里买尽量多的签字笔。已知一只签字笔的价格是1元9角，而班主任给小玉的钱是a元b角，小玉想知道，她最多能买多少只签字笔呢。

### 输入

输入只有一行两个整数，分别表示a和b。

### 输出

输出一行一个整数，表示小玉最多能买多少只签字笔。

### 输入样例 1

10 3

### 输出样例 1

5

### 提示

#### 数据范围

对于全部的测试点，保证 $0 \leq a \leq 10^4$ ， $0 \leq b \leq 9$ 。

### 来源

洛谷

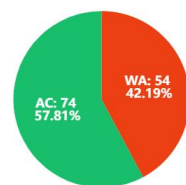
#### 题目信息

ID	34
时间限制	1000MS
内存限制	256MB
IO 类型	Standard IO
出题人	blackcat
难度	低
分数	100
标签	<a href="#">显示</a>

#### 统计

[Details](#)

AC WA



```
1 let buf = "";
2
3 process.stdin.on("readable", function(){
4     let chunk = process.stdin.read();
5     if(chunk) buf += chunk.toString();
6 });
7
8 process.stdin.on("end", function(){
9     let [a, b] = buf.split(" ").map(x => {return parseInt(x)});
10    console.log(parseInt((a*10+b)/19));
11 });
```

## 格式化输入输出

```
1 let name = 'cat', age = 28;
2 let info = `My name is ${name}, I\'m ${age} years old.`;
3
4 console.log(info);
5 console.log(`My name is ${name}, I\'m ${age} years old.`);
```

公众号：黑猫编程

网址：<https://noi.hioier.co>



## 条件判断语句

语法格式：



```
if (判断条件1){  
    执行语句1  
}else if(判断条件2){  
    执行语句2  
}else{  
    执行语句3  
}
```

逻辑运算符：



易混淆运算符:

**小tip:**

**=== 与 !==**

**===**绝对等于值和数据类型均相同才会为true

**!==**不绝对等于值或类型只要有一个不相同就为true

## 计算邮资

## 描述

根据邮件的重量和用户是否选择加急计算邮费。计算规则：重量在1000克以内(包括1000克), 基本费8元。超过1000克的部分, 每500克加收超重费4元, 不足500克部分按500克计算; 如果用户选择加急, 多收5元。

## 输入

输入一行, 包含整数和一个字符, 以一个空格分开, 分别表示重量(单位为克)和是否加急。如果字符是y, 说明选择加急; 如果字符是n, 说明不加急。

## 输出

输出一行, 包含一个整数, 表示邮费。

### 输入样例 1

1200 y

### 输出样例 1

17

#### 题目信息

ID	211
时间限制	1000MS
内存限制	256MB
IO 类型	Standard IO
出题人	blackcat
难度	低
分数	100
标签	<a href="#">显示</a>

#### 统计

Details

■ AC ■ WA

```
1 let buf = "";
2
3 process.stdin.on("readable", function(){
4     let chunk = process.stdin.read();
5     if(chunk) buf += chunk.toString();
6 });
7
8 process.stdin.on("end", function(){
9     let a = buf.split();
10    let w = parseInt(a[0]);
11    let flag = a[1];
12    console.log(a);
13    console.log(a[0], a[1]);
14    console.log(w, flag);
15    console.log(typeof(w), typeof(flag));
16
17    let res = 0;
18    if(w <= 1000) res = 8;
19    else{
20        res = 8 + Math.floor((w - 1000) / 500) * 4;
21        console.log(res);
22        if((w - 1000) % 500) res += 4;
23        console.log(res);
24    }
25
26    if(flag === "y") res += 5;
27
28    console.log(res);
29 });
```

## 循环结构

for循环语法:

```
1 for(初始化变量; 条件表达式; 变量更新 ){
2     //循环体
3 }
```



名称	作用
初始化变量	通常被用于初始化一个计数器，该表达式可以使用 let 关键字声明新的变量，这个变量帮我们来记录次数。
条件表达式	用于确定每一次循环是否能被执行。如果结果是 true 就继续循环，否则退出循环。
变量更新	循环体结束后，进行变量更新操作。

双重for循环：

```
1  for（外循环的初始；外循环的条件；外循环的操作表达式） {  
2      for（内循环的初始；内循环的条件；内循环的操作表达式） {  
3          需执行的代码；  
4      }  
5  }
```

while循环语法：

```
1  while（条件表达式） {  
2      // 循环体代码  
3  }
```

do.....while循环：

```
1  do {  
2      // 循环体代码 - 条件表达式为 true 时重复执行循环体代码  
3  } while(条件表达式);
```

循环跳出：

```
1  break: 跳出当前循环，多重循环，只跳出最内层循环。  
2  continue: 跳出本次循环，继续进行下一次循环。
```

## 循环求和

## 描述

计算输出 $1+2+3+\dots+n$ 的和。

## 输入

输入 $n$ 。

## 输出

如题所述之和。

### 输入样例 1

10

### 输出样例 1

55

## 提示

对于100%的数据， $1 \leq n \leq 100$ 。

```
1 let buf = "";  
2  
3 process.stdin.on("readable", function(){  
4     let chunk = process.stdin.read();  
5     if(chunk) buf += chunk.toString();  
6 });  
7  
8 process.stdin.on("end", function(){  
9     let n = parseInt(buf);  
10    let res = 0;  
11    for(let i = 1; i <= n; i++)  
12        res += i;  
13  
14    console.log(res);  
15 });
```

## 求整数的和与均值

## 描述

读入 $n(1 \leq n \leq 10000)$ 个整数，求它们的和与均值。

## 输入

输入第一行是一个整数 $n$ ，表示有 $n$ 个整数。

第2~ $n+1$ 行每行包含1个整数。每个整数的绝对值均不超过10000。

## 输出

输出一行，先输出和，再输出平均值(保留到小数点后5位)，两个数间用单个空格分隔。

### 输入样例 1

```
4
344
222
343
222
```

### 输出样例 1

```
1131 282.75000
```

```
1 let buf = "";
2
3 process.stdin.on("readable", function(){
4     let chunk = process.stdin.read();
5     if(chunk) buf += chunk.toString();
6 });
7
8 process.stdin.on("end", function(){
9
10    let a = buf.split("\n").map(x => {return parseInt(x)});
11    // console.log(a);
12    let n = a[0];
13
14
15    let sum = 0;
16
17    let i = 1;
18    while(i <= n){
19        sum += a[i];
20        i++;
21    }
22
23    console.log(sum + ' ' + (sum/n).toFixed(5));
24
25 });
```

## 描述

监护室每小时测量一次病人的血压，若收缩压在90-140之间并且舒张压在60-90之间(包含端点值)则称之为正常，现给出某病人若干次测量的血压值，计算病人保持正常血压的最长小时数。

## 输入

第一行为一个正整数 $n$ ( $n \leq 100$ )，其后有 $n$ 行，每行2个正整数，分别为一次测量的收缩压和舒张压。

## 输出

输出仅一行，血压连续正常的最长小时数。

### 输入样例 1

```
4
100 80
90 50
120 60
140 90
```

### 输出样例 1

```
2
```

```
1 let buf = "";
2
3 process.stdin.on("readable", function(){
4     let chunk = process.stdin.read();
5     if(chunk) buf += chunk.toString();
6 });
7
8 process.stdin.on("end", function(){
9
10     let line = buf.split("\n");
11     let n = parseInt(line[0]);
12     // console.log(n);
13
14     let cnt = 0, maxv = 0;
15     for(let i = 1; i <= n; i++){
16         let [a, b] = line[i].split(" ").map(x => {return parseInt(x)});
17         // console.log(a, b);
18         if(a >= 90 && a <= 140 && b >= 60 && b <= 90) cnt++;
19         else{
20             maxv = Math.max(maxv, cnt);
21             cnt = 0;
22         }
23     }
24     console.log(Math.max(maxv, cnt));
25 });
```

