

# 中山大学数据科学与计算机学院本科生实验报告

## (2016 学年春季学期)

课程名称: Algorithm design

任课教师: 张子臻

年级	1501	专业 (方向)	移动信息工程
学号	15352015	姓名	曹广杰
电话	13727022190	Email	<a href="mailto:1553118845@qq.com">1553118845@qq.com</a>
开始日期	2017/5/21	完成日期	2017/5/21

### 1. 实验题目



### 2. 实验目的

- 1) 有个损坏的里程表, 不能显示数字 4, 会从数字 3 直接跳到数字 5
- 2) 给出里程表的读数, 求出实际里程;

### 3. 程序设计

思路如下:



具体解释如下:

1. 首先, 输入的数据需要储存。这里使用字符串进行储存, 方便此后对于每一位的提取分析;
2. 输入数据储存之后, 分析题意。由于输入的数据是凡是数字是 4 的就舍去, 那么整体进制则成为 9, 故而所有数字都是 9 进制的, 此后通过对每一个位数的数字的筛选即可得出当前数字表示的正确数据;
3. 在分析数字的时候, 一个数字是否表示了该数字的所有信息, 或者该数字是否正确地表示了这个数字应该代表的所有信息是非常重要的, 最起码是非常基础的。这时候除了使用重新计数的原始方法进行

模拟计算之外，进制的操作方式应该被重视。这是非常行之有效的简单的方法；

4. 在表示原有数字的数值的时候，模拟是一种非常暴力的方法。但是，4 的频繁不能出现总是在暗示着什么。一开始笔者想到的除法，但是数字的分配在连续的数字段并不均匀——因为前一位的数字的缺失并不是下一位数字引起的，各个位置是相互独立的。各个位置有自己的递增规则，这时候，进制转化的思想就昭然若揭了。

## 4. 程序运行与测试

```
2
2: 2
3
3: 3
17
17: 15
41
41: 37
20
20: 18
666
666: 455
12
12: 11
53
53: 39
0
-----
Process exited after 14.47 seconds with return value 0
请按任意键继续. . .
```

## 5. 实验总结与心得

1. 本次实验中需要强调的位置是对于数字误表示的处理方式。  
误码数字的处理，如果直接使用数组进行储存，原有的信息确实可以被完整地保存。但是，在输入的时候会很不方便，string 已经在检索上提供了很便捷的方式。在分析数字的时候，一个数字是否表示了该数字的所有信息，或者该数字是否正确地表示了这个数字应该代表的所有信息是非常重要的，最起码是非常基础的。这时候除了使用重新计数的原始方法进行模拟计算之外，进制的操作方式应该被重视。这是非常行之有效的简单的方法。
2. 进制转化的思维方式。  
笔者使用的是 9 进制转化。首先声明，在表示原有数字的数值的时候，（按题意中表示，从零公里模拟）模拟是一种非常暴力的有效的方法。但是，4 的频繁不能出现总是在暗示着什么——各个位置的进制结构都是完全一样的，这种完全一致的进制结构表示的模型，说没有算法都不相信。一开始笔者想到的除法，但是数字的分配在连续的数字段并不均匀——因为前一位的数字的缺失并不是下一位数字引起的，各个位置是相互独立的。各个位置有自己的递增规则，这时候，进制转化的思想就昭然若揭了。
3. 配图：

 

smie15352015

where there is will, there is a way.  
[Logout](#)[Home](#) [Problems](#) [Contests](#) [Courses](#) [Ranklist](#) [Submit](#) [Setting](#) [Status](#) [Discuss](#) [<->](#)[Problem](#)  
[Description](#)**Solved****Number**

2270

**Submit****Number**

3838

**Problem**  
*solved*[Statistics](#)[Source code](#)[Discuss](#)[Submit](#)

### [1240. Faulty Odometer](#)

#### [Constraints](#)

Time Limit: 1 secs, Memory Limit: 32 MB

#### [Description](#)

You are given a car odometer which displays the miles traveled as an integer. The odometer has a defect, however: it proceeds from the digit 3 to the digit 5, always skipping over the digit 4. This defect shows up in all positions (the one's, the ten's, the

## 附录、提交文件清单

15352015-caogj-1240-v0

1240.cpp;