

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	1553118845@qq.com
开始日期	2017/5/13	完成日期	2017/5/14

1. 实验题目



2. 实验目的

将大于 2992 小于 9999 的数字进行 12 进制, 16 进制转换, 分别将各个进制的各个位数相加。如果最后做和结果等于 10 进制的和, 则输出;

3. 程序设计

```
进制转化函数 (数字, 进制) {  
    While (n/进制) {  
        储存取余结构;  
    }  
    储存数据倒置;  
}
```

1. 本次数据没有输入, 故不必考虑储存的数据结构;
2. 其实按照要求作出的实现主要是对于进制转化函数的实现, 阶段性进制转化过程结束之后, 获得某一位的数字, 做和即可;
3. 进制转化函数。进制转化的过程就是对于一个数字不断进行取余操作, 并将取余的数字倒置即可获得进制转化后的数字。这里笔者使用 stack 储存, 并实现转化。由于该过程比较简单, 此处不予赘言;

4. 有一点，每次转化之后都需要考虑将原有的数字项转化为字母的表示形式。即大于 10 的余数转化为 A B C D……以此类推，转化之后返回；
5. 检查输出；

4. 程序运行与测试

```
2992
2993
2994
2995
2996
2997
2998
2999
4470
4471
4472
4473
4474
4475
4970
4971
4972
4973
4974
4975
5460
5461
5462
5463
5464
5465
5466
5467
5468
搜狗拼音输入法 全：
5960
5961
5962
5963
6456
6457
6458
6459
7440
7441
7442
7443
7444
7445
7446
7447
7448
7449
7940
7941
```

5. 实验总结与心得

1. 本题主要使用的取余函数也是比较常规的函数结构，整体的构造过程难度不大；
2. 判断结构的设计，这次判断结构的设计笔者使用了等式的传递性，表征所有的信息都与 10 进制的计算值相等；
3. 此处笔者尝试强调了记录的力量，在特定的位置直接修改数据，以期达到记录的效果，后来意识到该解决方式其实与动态规划有所关联；
4. 最后的 debug 问题。尽管题目是比较简单，笔者还是莫名地遇到了很奇怪的 BUG，输出与样例一致，没有后续的输出。这里笔者想要强调一点——永远不要小看一个局部变量的恶心程度。如果遇到了循环结构体，注定是局部变量的记录 flag 一定要放在循环体内部，尤其是要记录某一种状态的 variable，否则上一次的信息会对这一次测试造成影响，并且如果像本题一样，前 8 个数据都是连续的，削弱了这里的影响，debug 的时候会很冤；
5. 附图：



smie15352015
where there is will, there is a way.
[Logout](#)

[Home](#) [Problems](#) [Contests](#) [Courses](#) [Ranklist](#) [Submit](#) [Setting](#) [Status](#) [Discuss](#) [<->](#)

[Problem](#)
[Description](#)

Solved
Number
5295

Submit
Number
9387

✓ Problem
solved

[Statistics](#)
[Source code](#)
[Discuss](#)

1014. Specialized Four-Dig

Constraints

Time Limit: 1 secs, Memory Limit: 32 MB

Description

Find and list all four-digit numbers in decimal notation that have the property that the sum of its four digits equals the sum of its digits when represented in hexadecimal (base 16) notation and also equals the sum of its digits when represented in duodecimal (base 12) notation.

附录、提交文件清单

1014. cpp

15352015-caogj-1014-v0;