

# 中山大学数据科学与计算机学院本科生实验报告

课程名称：算法设计与分析      任课教师：张子臻

年级	2017级	专业（方向）	软件工程
学号	17343100	姓名	仕润昊
电话	13280152626	Email	<a href="mailto:1056627011@qq.com">1056627011@qq.com</a>
开始日期	2019/4/10	完成日期	2019/4/12

## 完成情况——九道题

Problems only solved by r17343100

1011	1012	1014	1020	1021	1027	1028	1029	1034	1035	1046
1051	1093	1099	1119	1121	1134	1140	1148	1152	1153	1154
1163	1176	1198	1214	1203	1206	1231	1240	1259	1264	1345
1351	1381	1438	1500	1527	1628	1641	1775	1783	1828	1939
13062										

## 1.实验题目

### 1014

题意：

求出所有在十进制，十二进制，十六进制下各位数字之和相等的四位十进制数。

### 1099

题意：

有两种飞机A和B，花费分别为costA和costB，容量分别为passengersA和passengersB，现在要N个乘客，每个飞机都要完全装满乘客才能起飞，求最小费用

约束：

$$1 \leq n \leq 2,000,000,000$$

costA, passengersA, costB, and passengersB are all nonnegative integers having values less than 2,000,000,001

## 1119

题意：

1960年发行了4位计算机，从此以后美国十年，计算机的位数变成之前的两倍。输入某一年份，求出这个年份的最大整数 $n$ 使得 $n!$ 能被一个 $k$ 表示。

即算出满足  $n! < 2^k$  式子的最大整数 $n$ ，其中 $k = 2^{2+(year-1960)/10}$ 。

约束：

$1960 \leq year \leq 2160$

## 1203

题意：

题目 给出了一个有趣的现象：如果一个数字串，以1,3,7,9结尾，则会有一个数，他的三次方以这个数字串结尾，且长度不会超过这个数字串。

约束：

数字串长度  $1 \leq n \leq 10$

## 1214

题意：

给出数列 $a_1=1$ ， $a_3=3$ ， $a_{2n}=a_n$ ， $a_{4n+1} = 2*a_{2n+1} - a_n$ ， $a_{4n+3} = 3*a_{2n+1} - 2*a_n$

现在给定 $L$ ，求在 $[1,L]$ 中 $a_n=n$ 的个数

约束：

$1 \leq L \leq 2^{32}$

## 1231

题意：

给出两个正整数 $k$ ， $l$ ，问 $k$ 是否存在小于 $l$ 的质因数，有的话则给出最小的质因数。

约束：

$4 \leq k \leq 10^{100}$ ， $2 \leq l \leq 10^6$

## 1240

题意：

有一个损坏的里程表，不能显示数字4，会从数字3直接跳到数字5，给出里程表的读书，求出实际里程。

约束：

里程表的读数为1..999999999且不包含数字4

## 1259

题意：

给出一个正整数，求出他有多少种方法可以表示成连续的素数的和。

约束：

$2 \leq n \leq 10000$

## 1500

题意：

给出一个正整数 $k$ ，找到与之相邻的两个素数，并求出两个素数只差，如果 $k$ 是素数则输出0。

约束：

$1 \leq k \leq 1299709$

## 2.实验目的

---

练习高精度算法、贪心算法等，熟练掌握相关的思路。

## 3.程序设计

---

### 1014

暴力搜索即可，从 $n=2992$ 一直搜索到9999，如果 $n$ 的12进制表示16进制表示和10进制表示各个数字的和相同就输出。

```
for (int i = 2992; i <= 9999; i++) {
    int a = get_dec(i);
    int b = get_hex(i);
    int c = get_duo(i);
    if (a == b && b == c) {
        cout << i << endl;
    }
}
```

其中int get\_dec(int n) 如下

```
int get_dec(int n) {
    return n % 10 + n / 10 % 10 + n / 100 % 10 + n / 1000 % 10;
}
```

### 1099

贪心方法。

如果airplane A 和airplane B的数量为0，则需要做特殊判断。

如果airplane A 和airplane B的数量均不为0。我们先计算费用位置比，即 $value = cost/passeger$ ，value越小代表越划算。所以我们比较飞机A和B的value，如果A的value小于 B，那么我们先假设所有的乘客都乘坐A。

用乘客数除以载客量得到飞机的数量 $n$ ，然后计算剩下的乘客数量，如果这些乘客恰好能坐满飞机B的整数倍，如果不能就减少飞机A的数量，以此类推。当飞机A的数量为0时仍不能满足条件，则输出 `cannot be flown`

```
if (valueA <= valueB) {
    bool flag = false;
    long long countA = n / passengerA;
    long long countB;
    for (countA; countA >= 0; countA--) {
        if ((n-countA*passengerA) >= 0 && (n - countA * passengerA) % passengerB == 0) {
            countB = (n - countA * passengerA) / passengerB;
            cout << countA << " aircraft A, " << countB << " aircraft B" << endl;
            flag = true;
            break;
        }
    }
    if (flag == false) {
        cout << "cannot be flown" << endl;
    }
}
```

## 1119

本题的题意就是算出满足  $n! < 2^k$  式子的最大整数 $n$ ，其中 $k = 2^{2+(year-1960)/10}$ ， $year$ 的最大取值可以达到2160，就算用long long也会超出范围，所以我们不妨对不等式两边取对数得到

$$\log n + \log(n-1) + \dots + 0 < \log(2) * 2^{(2+(year-1960)/10)}$$

等式右边是定值，左边只需要暴力搜索即可。

```
double right = pow(2.0, (year - 1960) / 10 + 2);
double sum = 0;
int n = 1;
do{
    sum += log(double(n))/log(double(2.0));
    n++;
} while (sum < right);
cout << n - 2 << endl;
```

## 1203

本题就是计算满足  $a^3 \% 10^{\text{len}(b)} == b$  的  $a$  的值

这里单纯的穷举肯定会超时。我们知道竖式乘法中，积的第 $n$ 位（从右往左数）取决于因子的第1- $n$ 位，所以我们可以从第1位开始匹配，一直匹配度第 $\text{len}(b)$ 位。

又因为我们知道乘积的第一位是1,3,7,9，而相应的因子的第一位则一定是1,7,3,9，根据这个规律我们可以先求出因子的第一位。然后从因子的第二位开始计算。

以123为例，下一个我们要计算的因子的积为23，remain=23，此时因子的三次方应该模power=100，如果余数为23则可以计算下一位。

```
for (int i = num.size() - 2; i >= 0; i--) {
    remain += power * (num[i] - '0');
    long long step = power;
    power *= 10;
    while (cube(result, power) != remain) {
        result += step;
    }
}
```

然后就是计算因子的三次方了cube(result,power)。我们通过long long mul(long long a, long long b, long long mod)来实现，mul函数用来计算a\*b%mod的大小。

```
long long cube(long long num, long long mod) {
    return mul(mul(num, num, mod), num, mod);
}
```

因为a\*b可能会超出范围，所以我们对其进行特殊处理，我们知道

$$a * b \% mod = (b_n * 2^n + \dots + b_1 * 2^1 + \dots + b_0) * a \% mod$$

```
//c = a*b%mod = (bn*2^n+...+b1*2^1+...+b0)*a%mod
long long mul(long long a, long long b, long long mod) {
    long long c = 0;
    const int base = 2;
    for (; b != 0; b /= base) {
        c += (b % base) * a;
        c %= mod;
        a = (a * base) % mod;
    }
    return c;
}
```

最后输出结果即可。

## 1214

本题有一个规律，就是 $an=n$ 的充要条件是 $n$ 的二进制数为回文数，所以我们只要计算在 $n$ 之前回文串的个数即可。

因为回文串是左右对称的，任何一个数都可以通过我们定义的一种构造构造回文串。其中奇数构造为将这个数左右对掉后添加到这个数的右边，并且将右边的最高位与左边的最低位重合，比如说10的奇数构造为101。偶数构造为将这个数左右对调后添加到这个数的右边，比如说10的偶数构造为1001。每个数的奇数构造和偶数构造有且仅有一个，是一一映射的关系。

如果 $n$ 的二进制串长度为偶数，例如 $n=8$ ， $n = (1000)_2$ ，那么我们可以把它拆成一半，只保留左边部分，即10。这样小于或等于10的所有二进制串的奇数构造都小于1000，小于10的所有二进制串的偶数构造都小于1000，我们最后只需要判断10的偶数构造是否等于1000即可，在本例中是不可以的。

```

int half = num_str.length() / 2;
long long temp = num >> half;
long long temp_left = num - (temp << half);
if (half * 2 == num_str.length()) { // 偶
    int count = temp;
    count += pow(2, half) - 1;
    string reverse_left = num_str.substr(half, half);
    long long right_sum = 0;
    for (int k = 0; k < reverse_left.size(); k++) {
        right_sum = right_sum * 2 + reverse_left[k] - '0';
    }
    if (right_sum > temp_left) {
        count--;
    }
    cout << count << endl;
}

```

如果n的二进制串长度为奇数，讨论跟上面类似。

最后输出结果即可。

## 1231

本题的题意是判断k的素数因子是否比l小。

如果小就输出bad并输出这个因子，反之输出good。

本题k最大值可以到 $10^{100}$ 肯定要用高精度算法，而l的范围小于1000000，我们只需要从i=2开始循环到l（i为素数），判断k是否能整除i即可，所以就是一道高精度取模题。

先用埃氏筛法晒出1000000的素数。

```

memset(num, 1, sizeof(num));
for (int i = 2; i <= 1000000; i++) {
    if (num[i]) {
        for (int j = 2 * i; j <= 1000000; j += i) {
            num[j] = 0;
        }
    }
}

```

然后循环取模

```

for (int i = 2; i < l; i++) {
    if (num[i] && mod(k, i)) {
        flag = false;
        index = i;
        break;
    }
}

```

mod(k,i)的实现如下，利用的是

$$\begin{aligned} a \% m &= (a_n * 10^n + a_{n-1} * 10^{n-1} + \dots + a_1) \% m \\ &= (((a_n * 10 + a_{n-1}) * 10 + \dots) * 10 + a_1) \% m \\ &= (((a_n * 10 + a_{n-1}) \% m * 10 + \dots) * 10 + a_1) \% m \end{aligned}$$

```
bool mod(string k, int l) {
    long long m = 0;
    for (int i = 0; i < k.length(); i++) {
        m = (m * 10 + (k[i] - '0')) % l;
    }
    if (m == 0) {
        return true;
    }
    else return false;
}
```

## 1240

实际是进制转换，九进制转换为十进制，如果输入的数的某一位大于4，则该减一，然后将处理好的数字转换为十进制即可。

```
for (int i = 0; i < temp.length(); i++) {
    if (temp[i] - '0' > 4) {
        temp[i]--;
    }
}
long long sum = 0;
int base = 9;
for (int i = 0; i < temp.length(); i++) {
    sum = sum * base + (temp[i] - '0');
}
```

## 1259

用埃氏筛法筛选素数，对输入判断是否为素数，如果是和数就输出0，是素数就从2到n循环计算，暴力搜索即可。

```
for (int i = 2; i <= n; i++) {
    int sum = 0;
    for (int k = i; k <= n; k++) {
        if (num[k]) {
            if (sum == 0) {
                start = k;
            }
            sum += k;
        }
        if (sum == n) {
            count++;
        }
    }
}
```

```

        i = start;
        break;
    }
    if (sum > n) {
        break;
    }
}
}

```

## 1500

计算与输入相连的和数的数目，先用埃氏筛法筛选素数，然后枚举即可。

```

int count = 1;
for (int i = n - 1; i >= 0; i--) {
    if (!num[i]) count++;
    else break;
}
for (int i = n+1; i <= 1299709; i++) {
    if (!num[i]) count++;
    else break;
}

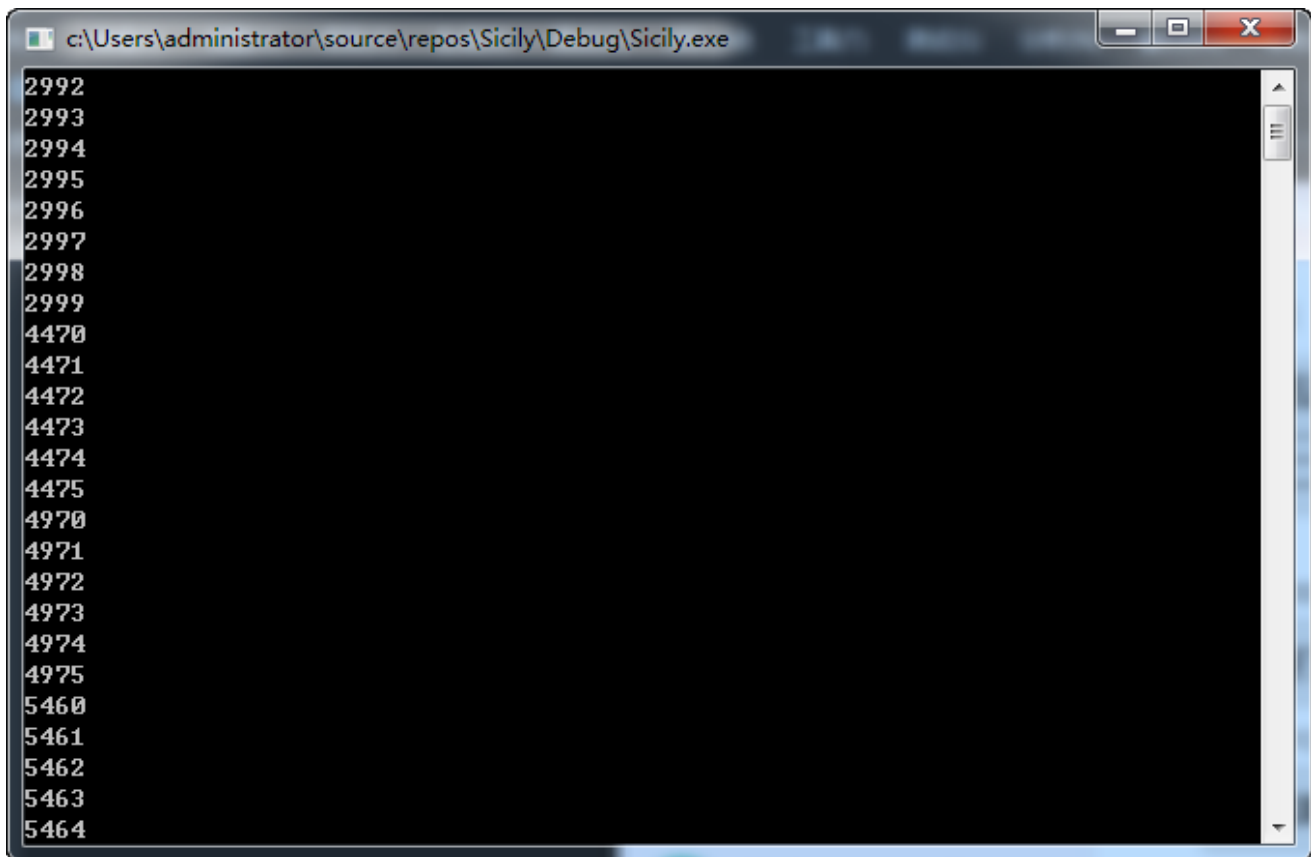
```

## 4.程序运行与测试

---

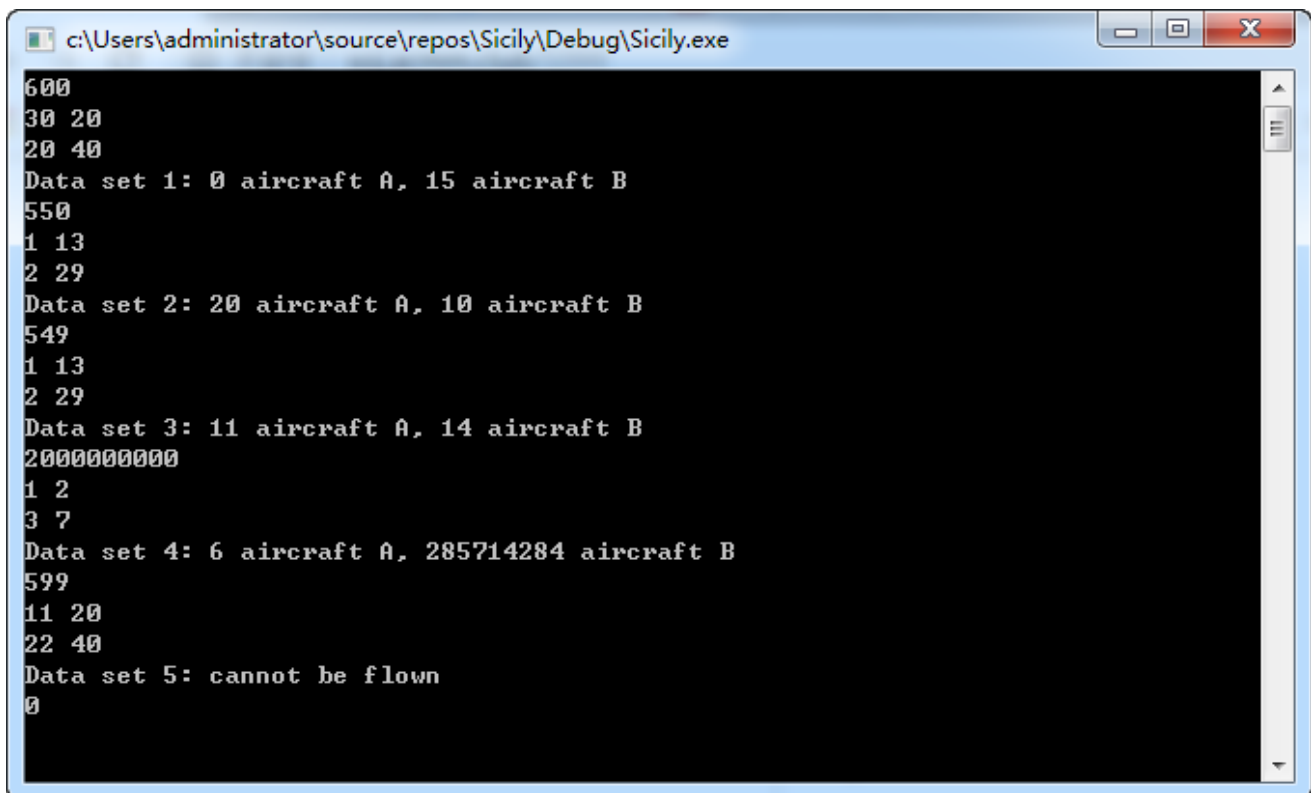
### 1014





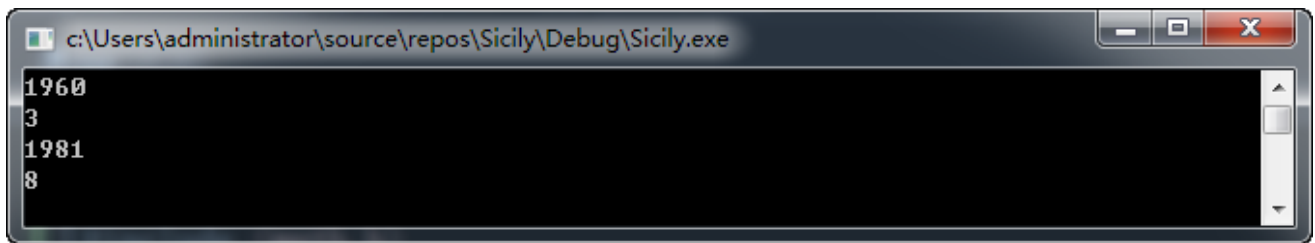
```
c:\Users\administrator\source\repos\Sicily\Debug\Sicily.exe
2992
2993
2994
2995
2996
2997
2998
2999
4470
4471
4472
4473
4474
4475
4970
4971
4972
4973
4974
4975
5460
5461
5462
5463
5464
```

1099



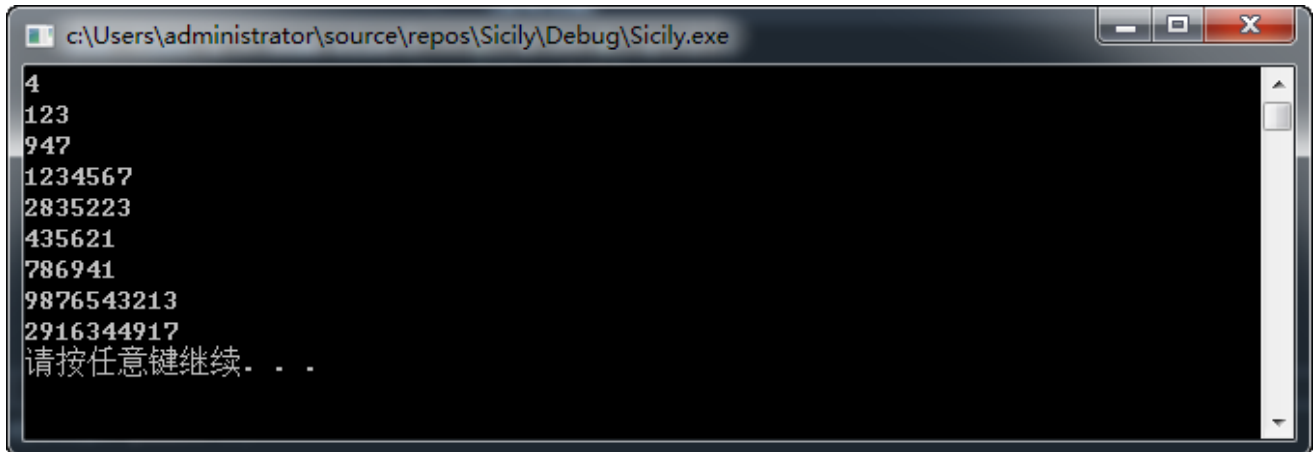
```
c:\Users\administrator\source\repos\Sicily\Debug\Sicily.exe
600
30 20
20 40
Data set 1: 0 aircraft A, 15 aircraft B
550
1 13
2 29
Data set 2: 20 aircraft A, 10 aircraft B
549
1 13
2 29
Data set 3: 11 aircraft A, 14 aircraft B
2000000000
1 2
3 7
Data set 4: 6 aircraft A, 285714284 aircraft B
599
11 20
22 40
Data set 5: cannot be flown
0
```

1119



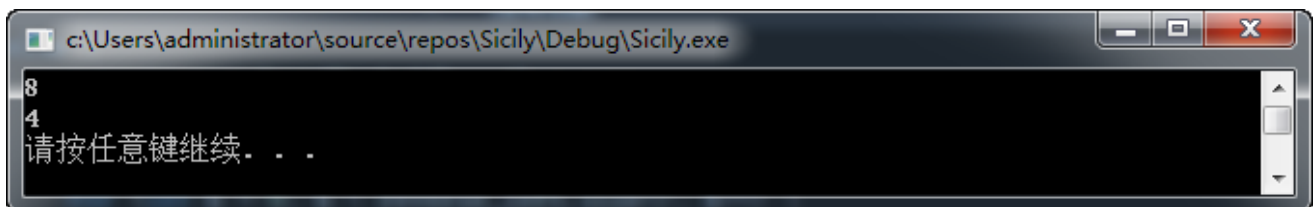
```
c:\Users\administrator\source\repos\Sicily\Debug\Sicily.exe
1960
3
1981
8
```

1203



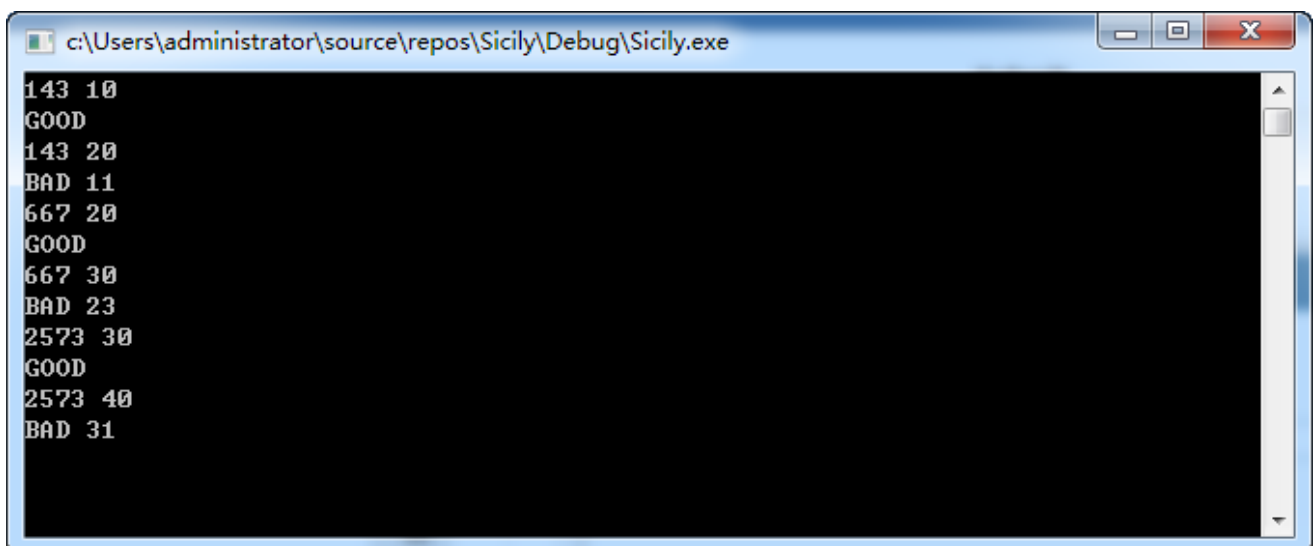
```
c:\Users\administrator\source\repos\Sicily\Debug\Sicily.exe
4
123
947
1234567
2835223
435621
786941
9876543213
2916344917
请按任意键继续...
```

1214



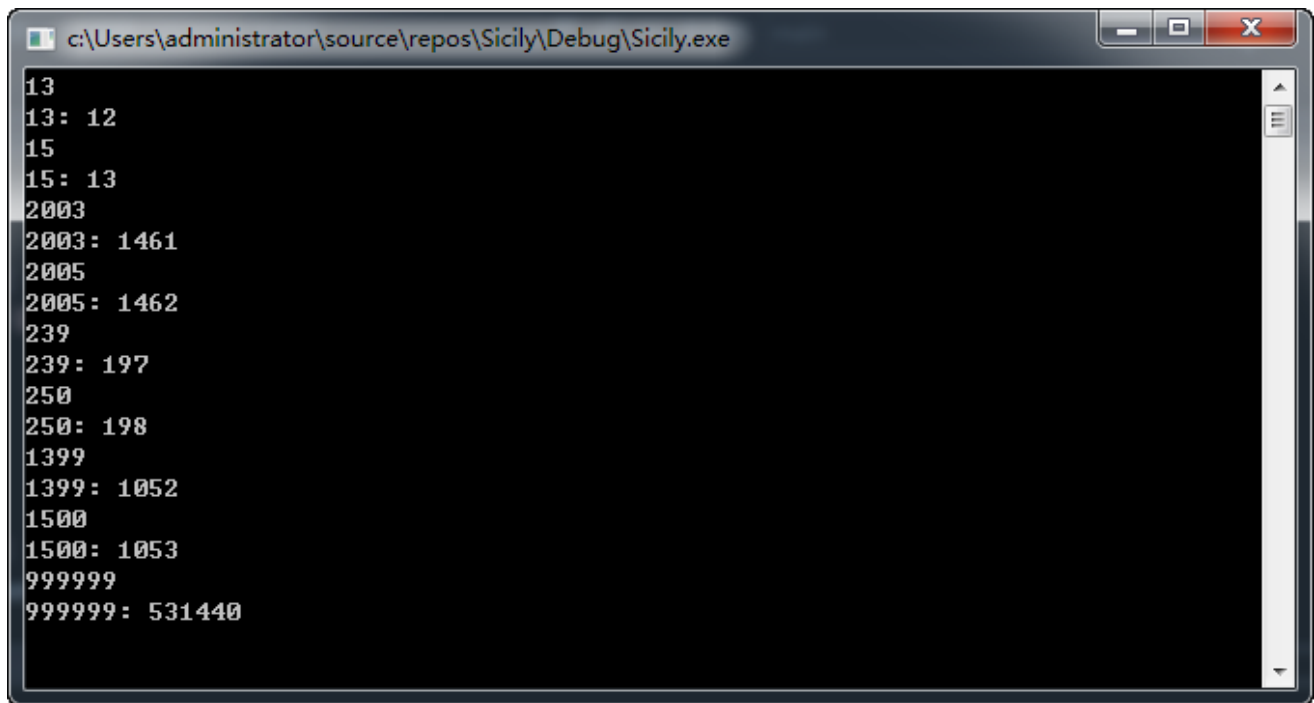
```
c:\Users\administrator\source\repos\Sicily\Debug\Sicily.exe
8
4
请按任意键继续...
```

1231



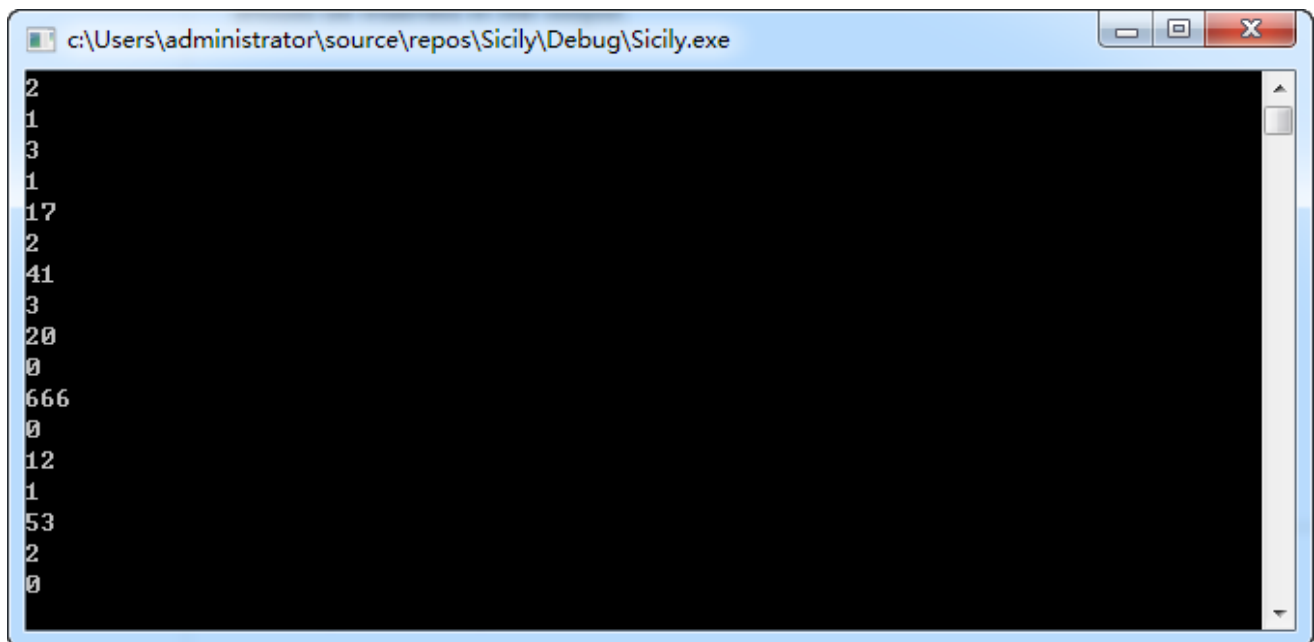
```
c:\Users\administrator\source\repos\Sicily\Debug\Sicily.exe
143 10
GOOD
143 20
BAD 11
667 20
GOOD
667 30
BAD 23
2573 30
GOOD
2573 40
BAD 31
```

1240



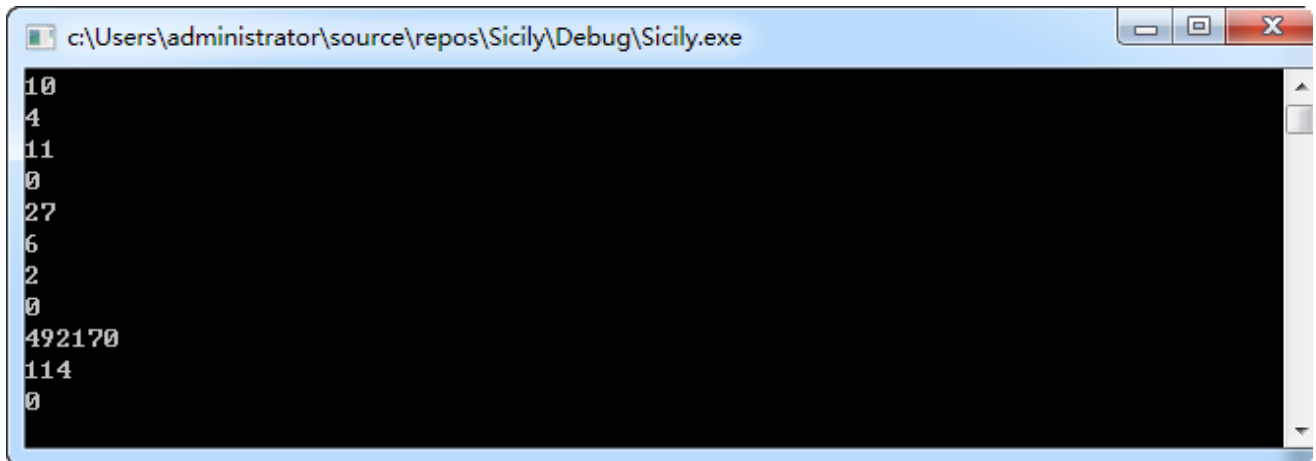
```
c:\Users\administrator\source\repos\Sicily\Debug\Sicily.exe
13
13: 12
15
15: 13
2003
2003: 1461
2005
2005: 1462
239
239: 197
250
250: 198
1399
1399: 1052
1500
1500: 1053
999999
999999: 531440
```

1259



```
c:\Users\administrator\source\repos\Sicily\Debug\Sicily.exe
2
1
3
1
17
2
41
3
20
0
666
0
12
1
53
2
0
```

1500



## 5.实验总结与心得

耗费时间最长的是回文串那道题，太容易算错，这次的题目基本比较简单，大部分都是高精度算法、贪心算法、筛法等等，主要还是要找到规律才能算对。

思路转换代码能力还是比较差，还要多加练习才行。

## 附录、提交文件清单

### 1014

```
// 1014
#include <iostream>
#include <string>
using namespace std;

int get_dec(int n) {
    return n % 10 + n / 10 % 10 + n / 100 % 10 + n / 1000 % 10;
}

int get_hex(int n) {
    return n % 16 + n / 16 % 16 + n / (16 * 16) % 16 + n / (16 * 16 * 16) % 16;
}

int get_duo(int n) {
    return n % 12 + n / 12 % 12 + n / (12 * 12) % 12 + n / (12 * 12 * 12) % 12;
}

int main() {
    for (int i = 2992; i <= 9999; i++) {
        int a = get_dec(i);
        int b = get_hex(i);
        int c = get_duo(i);
        if (a == b && b == c) {
            cout << i << endl;
        }
    }

    system("pause");
}
```

```
}
```

## 1099

```
// 1099
#include <iostream>
#include <string>
using namespace std;

int main() {
    int count = 0;
    while (true) {
        count++;
        long long n;
        cin >> n;
        if (n == 0) {
            return 0;
        }
        long long costA, passengerA;
        long long costB, passengerB;
        cin >> costA >> passengerA >> costB >> passengerB;
        cout << "Data set " << count << ": ";
        if (passengerA == 0) {
            if (passengerB == 0 || n%passengerB != 0) {
                cout << "cannot be flown" << endl;
            }
            else {
                cout << "0 aircraft A, " << n / passengerB << " aircraft B" << endl;
            }
            continue;
        }
        if (passengerB == 0) {
            if (passengerA == 0 || n%passengerA != 0) {
                cout << "cannot be flown" << endl;
            }
            else {
                cout << n/passengerA << " aircraft A, 0 aircraft B" << endl;
            }
            continue;
        }
        double valueA = double(costA) / passengerA;
        double valueB = double(costB) / passengerB;
        if (valueA <= valueB) {
            bool flag = false;
            long long countA = n / passengerA;
            long long countB;
            for (countA; countA >= 0; countA--) {
                if ((n-countA*passengerA) >= 0 && (n - countA * passengerA) % passengerB == 0) {
                    countB = (n - countA * passengerA) / passengerB;
                    cout << countA << " aircraft A, " << countB << " aircraft B" << endl;
                    flag = true;
                    break;
                }
            }
        }
    }
}
```

```

    }
}
if (flag == false) {
    cout << "cannot be flown" << endl;
}
}
else {
    bool flag = false;
    long countB = n / passengerB;
    long countA;
    for (countB; countB >= 0; countB--) {
        if ((n - countB * passengerB) >= 0 && (n - countB * passengerB) % passengerA ==
0) {
            countA = (n - countB * passengerB) / passengerA;
            cout << countA << " aircraft A, " << countB << " aircraft B" << endl;
            flag = true;
            break;
        }
    }
    if (flag == false) {
        cout << "cannot be flown" << endl;
    }
}
}
}
}

```

## 1119

```

// 1119
#include <iostream>
#include <math.h>
#include <string>
#include <vector>
using namespace std;
int main() {
    while (true) {
        int year;
        cin >> year;
        if (year == 0) {
            return 0;
        }
        // get the max n that satisfies  $n! < 2^k \rightarrow n! < 2^{\{2^{(2+(n-1960)/10)}\}}$   $\rightarrow \log n + \log n - 1 + \dots + 0 < \log(2) * (2 + (year - 1960) / 10)$ 
        double right = pow(2.0, (year - 1960) / 10 + 2);
        double sum = 0;
        int n = 1;
        do{
            sum += log(double(n))/log(double(2.0));
            n++;
        }
    }
}

```

```

    } while (sum < right);
    cout << n - 2 << endl;
}
}

```

## 1203

```

// 1203
#include <iostream>
#include <math.h>
#include <string>
#include <vector>
#include <sstream>
using namespace std;

int a[10] = { 0,1,0,7,0,0,0,3,0,9 };

// a^3%1000 = (a*a*a)%1000 = ((a*a)%1000)*a%1000

long long mul(long long a, long long b, long long mod) {
    long long c = 0;
    const int base = 2;
    for (; b != 0; b /= base) {
        c += (b % base) * a;
        c %= mod;
        a = (a * base) % mod;
    }
    return c;
}

long long cube(long long num, long long mod) {
    return mul(mul(num, num, mod), num, mod);
}

int main() {
    int T;
    cin >> T;
    while (T--) {
        string num;
        cin >> num;
        long long result = a[num[num.size() - 1] - '0'];
        long long remain = num[num.size()-1] - '0';
        long long power = 10;
        for (int i = num.size() - 2; i >= 0; i--) {
            remain += power * (num[i] - '0');
            long long step = power;
            power *= 10;
            while (cube(result, power) != remain) {
                result += step;
            }
        }
    }
}

```

```

    }
    cout << result << endl;
}
}

```

## 1214

```

// 1214
#include <iostream>
#include <math.h>
#include <string>
#include <vector>
#include <memory.h>
#include <sstream>
#include <queue>
#include <cstring>
#include <algorithm>
using namespace std;

// 700000
// 1, 2, 3,4,

int main() {

    long long num;
    cin >> num;
    if (num == 1) {
        cout << 1 << endl;
        return 0;
    }
    string num_str = "";
    long long num_cal = num;
    while (num_cal > 0) {
        int a = num_cal % 2;
        num_cal /= 2;
        num_str += a + '0';
    }
    int half = num_str.length() / 2;
    long long temp = num >> half;
    long long temp_left = num - (temp << half);
    if (half * 2 == num_str.length()) { // 偶
        int count = temp;
        count += pow(2, half) - 1;
        string reverse_left = num_str.substr(half, half);
        long long right_sum = 0;
        for (int k = 0; k < reverse_left.size(); k++) {
            right_sum = right_sum * 2 + reverse_left[k] - '0';
        }
        if (right_sum > temp_left) {
            count--;
        }
    }
}

```



```

    }
    cout << count << endl;
}
else {
    temp_left = num - ((temp>>1) << (half+1));
    int count = (pow(2, half)-1) + temp;
    string reverse_left = num_str.substr(half, half + 1);
    long long right_sum = 0;
    for (int k = 0; k < reverse_left.size(); k++) {
        right_sum = right_sum * 2 + reverse_left[k] - '0';
    }
    if (right_sum > temp_left) {
        count--;
    }
    cout << count << endl;
}
}
// system("pause");
}

```

## 1231

```

#include <iostream>
#include <math.h>
#include <string>
#include <vector>
#include <memory.h>
#include <sstream>
using namespace std;

int num[2000000];

bool mod(string k, int l) {
    long long m = 0;
    for (int i = 0; i < k.length(); i++) {
        m = (m * 10 + (k[i] - '0')) % l;
    }
    if (m == 0) {
        return true;
    }
    else return false;
}

int main() {
    memset(num, 1, sizeof(num));
    for (int i = 2; i <= 1000000; i++) {
        if (num[i]) {
            for (int j = 2 * i; j <= 1000000; j += i) {
                num[j] = 0;
            }
        }
    }
}

```

```

    }
    while (true) {
        string k;
        int l;
        cin >> k >> l;
        if (l == 0) {
            return 0;
        }
        bool flag = true;
        int index;
        for (int i = 2; i < l; i++) {
            if (num[i] && mod(k, i)) {
                flag = false;
                index = i;
                break;
            }
        }
        if (flag) {
            cout << "GOOD" << endl;
        }
        else {
            cout << "BAD " << index << endl;
        }
    }
}

```

## 1240

```

// 1240
#include <iostream>
#include <math.h>
#include <string>
#include <vector>
#include <memory.h>
#include <sstream>
using namespace std;

int main() {
    while (true) {
        long long n;
        cin >> n;
        if (n == 0) {
            return 0;
        }
        string temp;
        stringstream ss;
        ss << n;
        ss >> temp;

        for (int i = 0; i < temp.length(); i++) {

```

```

        if (temp[i] - '0' > 4) {
            temp[i]--;
        }
    }
    long long sum = 0;
    int base = 9;
    for (int i = 0; i < temp.length(); i++) {
        sum = sum * base + (temp[i] - '0');
    }
    cout << n << ": " << sum << endl;
}
}

```

## 1259

```

// 1259
#include <iostream>
#include <math.h>
#include <string>
#include <vector>
#include <memory.h>
#include <sstream>
using namespace std;

int num[20010];
int main() {
    memset(num, 1, sizeof(num));
    for (int i = 2; i <= 20000; i++) {
        if (num[i]) {
            for (int j = 2 * i; j <= 20000; j += i) {
                num[j] = 0;
            }
        }
    }
    while (true) {
        int n;
        cin >> n;
        if (n == 0) {
            return 0;
        }
        int count = 0;
        int start = 0;
        for (int i = 2; i <= n; i++) {
            int sum = 0;
            for (int k = i; k <= n; k++) {
                if (num[k]) {
                    if (sum == 0) {
                        start = k;
                    }
                    sum += k;
                }
            }
        }
    }
}

```

```

        }
        if (sum == n) {
            count++;
            i = start;
            break;
        }
        if (sum > n) {
            break;
        }
    }
}
cout << count << endl;
}
}

```

## 1500

```

// 1500
#include <iostream>
#include <math.h>
#include <string>
#include <vector>
#include <memory.h>
#include <sstream>
using namespace std;

int num[2000000];
int main() {
    memset(num, 1, sizeof(num));
    for (int i = 2; i <= 1299709; i++) {
        if (num[i]) {
            for (int j = 2 * i; j <= 1299709; j+=i) {
                num[j] = 0;
            }
        }
    }
    // for (int i = 2; i <= 1000; i++) {
    //     if (num[i]) cout << i << " ";
    // }
    while (true) {
        int n;
        cin >> n;
        if (n == 0) {
            return 0;
        }
        if (num[n]) {
            cout << 0 << endl;
            continue;
        }
    }
    int count = 1;
}

```

```
    for (int i = n - 1; i >= 0; i--) {  
        if (!num[i]) count++;  
        else break;  
    }  
    for (int i = n+1; i <= 1299709; i++) {  
        if (!num[i]) count++;  
        else break;  
    }  
    cout << count+1 << endl;  
}  
}
```