1069. The Black Hole of Numbers (20)

时间限制

100 ms

内存限制

65536 kB

代码长度限制

16000 B

判题程序

Standard

作者

CHEN, Yue

For any 4-digit integer except the ones with all the digits being the same, if we sort the digits in non-increasing order first, and then in non-decreasing order, a new number can be obtained by taking the second number from the first one. Repeat in this manner we will soon end up at the number 6174 -- the "black hole" of 4-digit numbers. This number is named Kaprekar Constant.

For example, start from 6767, we'll get:

7766 - 6677 = 1089  
9810 - 0189 = 9621  
9621 - 1269 = 8352  
8532 - 2358 = 6174  
7641 - 1467 = 6174  
... ...

Given any 4-digit number, you are supposed to illustrate the way it gets into the black hole.

**Input Specification:**

Each input file contains one test case which gives a positive integer N in the range (0, 10000).

**Output Specification:**

If all the 4 digits of N are the same, print in one line the equation "N - N = 0000". Else print each step of calculation in a line until 6174 comes out as the difference. All the numbers must be printed as 4-digit numbers.

**Sample Input 1:**

6767

**Sample Output 1:**

7766 - 6677 = 1089

9810 - 0189 = 9621

9621 - 1269 = 8352

8532 - 2358 = 6174

**Sample Input 2:**

2222

**Sample Output 2:**

2222 - 2222 = 0000

[提交代码](https://www.patest.cn/contests/pat-a-practise/1069)

犯了一个不该犯的错误，在输入6174时判断条件会被直接跳过，但是当时若已经知道有数据出错应该从代码里找出可能出错的bug，而不是自己瞎凑数据，仔细看的话能看得出来的，应以为鉴。

#include<iostream>

#include<string>

#include<algorithm>

#include<queue>

#include<vector>

#include<sstream>

#include<stack>

#include<map>

#define MAX 505

using namespace std;

int face\_coin[MAX] = { 0 };

vector<int>increase;

vector<int>decrease;

int comp(int a, int b)

{

return a > b;

}

int trans\_d(int num)

{

decrease.clear();

int time = 4;

while (time--)

{

int current = num - num / 10\*10;

decrease.push\_back(current);

num /= 10;

}

sort(decrease.begin(), decrease.end());

int total = 0;

for (int i = 0; i < decrease.size(); i++)

{

total += decrease[i] \* pow(10, i);

}

return total;

}

int trans\_i(int num)

{

decrease.clear();

int time = 4;

while (time--)

{

int current = num - num / 10 \* 10;

decrease.push\_back(current);

num /= 10;

}

sort(decrease.begin(), decrease.end(),comp);

int total = 0;

for (int i = 0; i < decrease.size(); i++)

{

total += decrease[i] \* pow(10, i);

}

return total;

}

int main()

{

int num;

cin >> num;

if (num == 6174)

{

cout << "7641 - 1467 = 6174\n";

return 0;

}

while (num != 6174)

{

int decrease = trans\_d(num);

int increase = trans\_i(num);

num = decrease - increase;

printf("%04d - %04d = %04d\n", decrease, increase, num);

if (!num)

break;

}

}