



链滴

PAT 甲级刷题实录——1023

作者: [aopstudio](#)

原文链接: <https://ld246.com/article/1581068665462>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

原题链接

<https://pintia.cn/problem-sets/994805342720868352/problems/994805478658260992>

思路

这题基本思路是统计从 0 到 9 的各个数字的出现个数，可以用 size 为 10 的 vector `<int>` 来存储每数字的出现次数。在统计输入的原数字时，进行 +1 操作，统计原数字的两倍时，进行减 1 操作，最看 vector 的每一位是否都为 0。本来想用 int 存储输入的数字，然后用 %10 来取每一位的数字。但发现题目中有一个要求，数字长度最多为 20 位，这已经超出了 int 类型所能存储的最大范围了。比 int 更大的还有 long long，但很可惜，还是不够，还有一个更大的 unsigned long long，因为题目中用的是 positive integer 也就是正整数，所以可以用 unsigned 存储，但很可惜，就算是 unsigned long long 也还是不够。当所有的整数类型都不够存储的时候，我们就要将整数转化为一个个的字符来处理了。`cin.get()` 方法能够读取单个字符，我们将读取到的字符与 '0' 相减，就能得出该字符对应的 int，本质上这个相减的过程是两个字符的 ASCII 码相减。我们将读取到的的每位数字动态插入 vector。之后计算两倍时，从原数字的个位开始分别对每一位进行计算，如果计算结果大于 9，则需要进行进位操作。具体操作见代码

完整代码

```
#include <iostream>
#include <vector>
#include <string>
using namespace std;

int main()
{
    vector<int> count(10,0);
    char c;
    vector<int> num;
    vector<int> dbNum;
    bool flag = true;
    while ((c = cin.get()) != '\n')
    {
        int digit = c - '0';
        count[digit]++; //统计各个数字出现次数
        num.push_back(digit);
    }
    dbNum.assign(num.size() + 1, 0);
    for (int i = num.size()-1; i >= 0; i--)
    {
        int digit = num[i];
        int index = num.size() - 1 - i;
        dbNum[index] += digit * 2 % 10; //当前位进行两倍操作
        dbNum[index + 1] += digit * 2 / 10; //进位
    }

    for (int i = dbNum.size() - 1; i >= 0; i--) //统计两倍后各个数字出现的次数
    {
        if (i == dbNum.size() - 1 && dbNum[i] == 0)
            continue;
        int digit = dbNum[i];
```

```
    count[digit]--;
}
for (int i = 0; i < 10; i++)
{
    if (count[i] != 0)
    {
        flag = false;
        break;
    }
}
if (flag)
{
    cout << "Yes" << endl;
}
else
{
    cout << "No" << endl;
}
for (int i = dbNum.size() - 1; i >= 0; i--)
{
    if (i == dbNum.size() - 1 && dbNum[i] == 0)
        continue;
    int digit = dbNum[i];
    cout << digit;
}
return 0;
}
```