

PAT 甲级刷题实录——1005

作者: [aopstudio](#)

原文链接: <https://ld246.com/article/1578830336924>

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/994805519074574336>

思路

这题相对比较简单，就是逐个读入数字然后加起来，算出来的总和从最高位开始逐位用英文输出。需要注意的是，题中所给的数字是连在一起的，没有用空格隔开，而如果直接用cin会把一整串数字都读去，这不是我们想要的。逐个字符读入需要用到的是`cin.get()`方法。不过这样读进来的是char类型，如果用int输出相当于输出各个数字对应的ASCII码。我们需要将它转换成int类型，如果记得住数字对应ASCII值比数字多48，就可以直接减48，如果记不住也可以用`num = c - '0'`来转换，其中num是int型，c是char类型。关于什么时候停止读取，网上给的是`cin.get() != EOF`，不过我试了一下，发现PA的输入最后还有一位换行符，如果只有EOF来判断，那么最后还会读入换行符对应的ASCII值10。因我们需要再加一个判断即`(c = cin.get()) != EOF && c!=10`。

读入的同时进行累加运算，读完所有数字后就已经得到了它们的总和。题目中要求从最高位开始逐位英文输出。先解决英文输出的问题，这个简单，写一个方法专门进行转换即可。如何从最高位开始输出，我目前没有想到直接从最高位开始输出的方法，但我知道怎么从最低位开始输出，只要每次取模10的结果并且再将总和自身除以10即可。我可以把从最低位开始的每位数字存在vector中，之后倒序输出即可。代码如下：

代码

```
#include <iostream>
#include <vector>
#include <string>
using namespace std;
string transfer(int digit);
int main()
{
    char c;
    int num,sum=0;
    vector<int> result;
    while ((c = cin.get()) != EOF && c!=10)
    {
        num = c - '0';
        sum += num;    //累加计算总和
    }
    if (sum == 0)    //考虑总和为0的情况，此时直接输出zero
    {
        cout << "zero";
        return 0;
    }
    while (sum != 0)
    {
        result.push_back(sum % 10);    //从个位开始一位位往vector中添加数字
        sum /= 10;
    }
    for (int i = result.size() - 1; i >= 0; i--)    //倒序输出vector
    {
        if (i == result.size() - 1)
            cout << transfer(result[i]);
```

```

        else
            cout << ' ' << transfer(result[i]);
    }
    return 0;
}
string transfer(int digit)  //阿拉伯数字和英文转换器
{
    switch (digit)
    {
        case 0:
            return "zero";
            break;
        case 1:
            return "one";
            break;
        case 2:
            return "two";
            break;
        case 3:
            return "three";
            break;
        case 4:
            return "four";
            break;
        case 5:
            return "five";
            break;
        case 6:
            return "six";
            break;
        case 7:
            return "seven";
            break;
        case 8:
            return "eight";
            break;
        case 9:
            return "nine";
            break;
        default:
            break;
    }
    return "NaN";  //用不到，但函数必须得有返回值
}

```

一开始我没有考虑总和一开始就为0的情况，所以有一个测试点过不去。之后加上了一条专门判断总是否为0的语句。