

#1 数字根 解答

代码

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

int digit_sum(int n)
{
    if (n < 10)
        return n;

    return digit_sum(n / 10) + n % 10;
}

int digital_root(int n)
{
    if (n < 10) return n;

    return digital_root(digit_sum(n));
}

int main()
{
    int n;
    cin >> n;
    cout << digital_root(n);

    return 0;
}
```

解析

根据题意，解题的基本思路是，递归地对当前数字的各位数字求和，直到各位数字的和是一个个位数（也即 < 10 的自然数）。唯一的难点知识写出那个“对各位数字求和”的函数。

初学者写可能会写出一些比较繁冗的写法，这里的 `digit_sum` 使用递归思想，算得上简洁了。

小小的强调一下递归吧：递归思想其实和数列有很大渊源（也许以后大家会学到递推——递归的对应）。总的来说，写递归函数需要两个元素：

1. 递归关系
2. 边界条件

这里的 `digit_sum` 函数很明显的包含了这两个元素。

这一段代码： `if(n < 10) return n;` 对应了边界条件，也就是，递归要在哪里停止。这个元素是必须的，没有边界条件或者边界条件错误很多时候会导致无穷递归，你永远也得不到问题的答案。

而下面的 `return digit_sum(n / 10) + n % 10;` 对应了递归关系。递归关系的用途不必多言。大概相当于车之于开车，饭之于吃饭的关系吧。

啦啦啦啦啦啦啦。