

快乐数

编写一个算法来判断一个数 n 是不是快乐数。

「快乐数」定义为：

- 对于一个正整数，每一次将该数替换为它每个位置上的数字的平方和。
- 然后重复这个过程直到这个数变为 1，也可能是 **无限循环** 但始终变不到 1。
- 如果 **可以变为 1**，那么这个数就是快乐数。

如果 n 是快乐数就返回 `true`；不是，则返回 `false`。

示例 1：

输入：19

输出：true

解释：

$$1^2 + 9^2 = 82$$

$$8^2 + 2^2 = 68$$

$$6^2 + 8^2 = 100$$

$$1^2 + 0^2 + 0^2 = 1$$

示例 2：

输入：n = 2

输出：false

```
class Solution {
public:
    int getnext(int n)
    {
        int sum=0;
        while(n)
        {
            int r=n%10;
            n=n/10;
            sum+=r*r;
        }
        return sum;
    }
}
```

```
bool isHappy(int n) {  
    //快慢指针  
    //使用“快慢指针”思想，找出循环：“快指针”每次走两步，“慢指针”每次走一步，当二者相等时，即为一个循环周期。此时，判断是不是因为 1 引起的循环，是的话就是快乐数，否则不是快乐数  
    int slow=n;  
    int fast=getnext(n);  
    while(fast!=1&&slow!=fast)  
    {  
        slow=getnext(slow);  
        fast=getnext(getnext(fast));  
    }  
    return fast==1;  
}  
};
```