Write an algorithm to determine if a number is "happy".

A happy number is a number defined by the following process: Starting with any positive integer, replace the number by the sum of the squares of its digits, and repeat the process until the number equals 1 (where it will stay), or it loops endlessly in a cycle which does not include 1. Those numbers for which this process ends in 1 are happy numbers.

**Example:**19 is a happy number

* 12 + 92 = 82
* 82 + 22 = 68
* 62 + 82 = 100
* 12 + 02 + 02 = 1

提示

以十进位为例：

2 8 → 2²+8²=68 → 6²+8²=100 → 1²+0²+0²=1

3 2 → 3²+2²=13 → 1²+3²=10 → 1²+0²=1

3 7 → 3²+7²=58 → 5²+8²=89 → 8²+9²=145 → 1²+4²+5²=42 → 4²+2²=20 → 2²+0²=4 → 4²=16 → 1²+6²=37……

因此28和32是快乐数，而在37的计算过程中，37重覆出现，继续计算的结果只会是上述数字的循环，不会出现1，因此37不是快乐数。

不是快乐数的数称为不快乐数（unhappy number），所有不快乐数的数位平方和计算，最後都会进入 4 → 16 → 37 → 58 → 89 → 145 → 42 → 20 → 4 的循环中。

崔崔蠢萌自测版：

**public** **class** HappyNumber {

**public** **boolean** isHappy(**int** n) {

**int** res = flagHappy(n);

**if** (res == 1) **return** **true**;

**else** **return** **false**;

}

**public** **int** flagHappy(**int** n) {

**int** flag = 0;

**if**(n == 1||n ==10 ||n == 100||n == 1000||n == 10000||n ==100000||n == 1000000||n ==10000000||n == 100000000|| n == 1000000000)

{ flag = 1;

**return** flag;

}

**else** {

**if**(n == 4||n ==16 ||n == 37||n == 58||n == 89||n ==145||n == 42||n ==20 ) **return** flag;

**else** {

**if** (n < 10) **return** flagHappy((**int**)Math.*pow*(n,2));

**else** {

**if**(n < 100) **return** flagHappy((**int**)Math.*pow*(n%10,2)+(**int**)Math.*pow*(n/10,2));

**else** {

**if** (n < 1000) **return** flagHappy((**int**)Math.*pow*(n%10,2)+(**int**)Math.*pow*(n/100,2)+(**int**)Math.*pow*((n%100)/10,2));

**else** {

**if**(n < 10000) **return** flagHappy((**int**)Math.*pow*(n/1000,2)+(**int**)Math.*pow*((n%1000)/100,2)+(**int**)Math.*pow*((n%100)/10,2)+(**int**)Math.*pow*(n%10,2));

**else** {

**if**(n < 100000) **return** flagHappy((**int**)Math.*pow*(n/10000,2)+(**int**)Math.*pow*((n%10000)/1000,2)+(**int**)Math.*pow*((n%1000)/100,2)+(**int**)Math.*pow*((n%100)/10,2)+(**int**)Math.*pow*(n%10,2));

**else** {

**if**(n < 1000000) **return** flagHappy((**int**)Math.*pow*(n/100000,2)+(**int**)Math.*pow*((n%100000)/10000,2)+(**int**)Math.*pow*((n%10000)/1000,2)+(**int**)Math.*pow*((n%1000)/100,2)+(**int**)Math.*pow*((n%100)/10,2)+(**int**)Math.*pow*(n%10,2));

**else** {

**if**(n < 10000000) **return** flagHappy((**int**)Math.*pow*(n/1000000,2)+(**int**)Math.*pow*((n%1000000)/100000,2)+(**int**)Math.*pow*((n%100000)/10000,2)+(**int**)Math.*pow*((n%10000)/1000,2)+(**int**)Math.*pow*((n%1000)/100,2)+(**int**)Math.*pow*((n%100)/10,2)+(**int**)Math.*pow*(n%10,2));

**else** {

**if**( n < 100000000) **return** flagHappy((**int**)Math.*pow*(n/10000000,2)+(**int**)Math.*pow*((n%10000000)/1000000,2)+(**int**)Math.*pow*((n%1000000)/100000,2)+(**int**)Math.*pow*((n%100000)/10000,2)+(**int**)Math.*pow*((n%10000)/1000,2)+(**int**)Math.*pow*((n%1000)/100,2)+(**int**)Math.*pow*((n%100)/10,2)+(**int**)Math.*pow*(n%10,2));

**else** {

**if** (n < 1000000000) **return** flagHappy((**int**)Math.*pow*(n/100000000,2)+(**int**)Math.*pow*((n%100000000)/10000000,2)+(**int**)Math.*pow*((n%10000000)/1000000,2)+(**int**)Math.*pow*((n%1000000)/100000,2)+(**int**)Math.*pow*((n%100000)/10000,2)+(**int**)Math.*pow*((n%10000)/1000,2)+(**int**)Math.*pow*((n%1000)/100,2)+(**int**)Math.*pow*((n%100)/10,2)+(**int**)Math.*pow*(n%10,2));

**else** **return** flagHappy((**int**)Math.*pow*(n/1000000000,2)+(**int**)Math.*pow*((n%1000000000)/100000000,2)+(**int**)Math.*pow*((n%100000000)/10000000,2)+(**int**)Math.*pow*((n%10000000)/1000000,2)+(**int**)Math.*pow*((n%1000000)/100000,2)+(**int**)Math.*pow*((n%100000)/10000,2)+(**int**)Math.*pow*((n%10000)/1000,2)+(**int**)Math.*pow*((n%1000)/100,2)+(**int**)Math.*pow*((n%100)/10,2)+(**int**)Math.*pow*(n%10,2));

}

}

}

}

}

}

}

}

}

}

}

**public** **static** **void** main(String args[]){

**int** a = 19;

**int** b = 20;

System.***out***.println(**new** HappyNumber().isHappy(a));

System.***out***.println(**new** HappyNumber().isHappy(b));

}

}

大神未解版，然而我并不懂这是为什么：

**public** boolean isHappy(**int** n) {

**if** (n <= 0) **return** **false**;

HashSet<Integer> **set** = **new** HashSet<Integer>();

**while**(n !=1 && !**set**.contains(n)) {

**set**.add(n);

**int** m = n;

n = 0;

**while**(m != 0) {

n += (m%10)\*(m%10);

m = m/10;

}

}

**if** (n == 1) **return** **true**;

**else** **return** **false**;

}