Happy Number

由题意，如果给定的正数的各个位的平方和相加不等于1，把得到的平方和作为新的数进行重复操作判断，直到出现1为止，或者进入无限循环。这里给出了能得到1的例子

那么我们分析一个不能得到1的例子：如11

1^2 + 1^2 = 2  
2^2 = 4  
4^2 = 16  
1^2 + 6^2 = 37  
3^2 + 7^2 = 58  
5^2 + 8^2 = 89  
8^2 + 9^2 = 145  
1^2 + 4^2 + 5^2 = 42  
4^2 + 2^2 = 20  
2^2 + 0^2 =4

注意到这里4进入了循环，这样计算下去便是一个死循环。

所以我们需要对每次计算得到的结果与之前得到的结果集比较，若存在则判断是不是1，不存在就加到结果集里，这里采用哈希表来记录所有出现过的数字，然后每出现一个新数字，在哈希表中查找看是否存在，若不存在则加入表中，若存在则跳出循环，并且判断此数是否为1，若为1返回true，不为1返回false，代码如下：

class Solution {

public:

bool isHappy(int n) {

if(n<=0)

return false;

map<int,int> map; //这里map的key能有效就行,值随便填什么，是判断key

while(n>1) //是否存在

{

int sum=0;

while(n!=0)

{

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

n=n/10;

}

n=sum;

if(map.find(n)!=map.end()) //找到了

break;

else

map[n]=1;

}

if(n==1)

return true;

else

return false;

}

};