**人民币的构造**

我们都知道人民币的面值是1、2、5、101、2、5、10，为什么是这个数值呢，我们分析了下发现，从1−101−10的每个数字都可以由每种面值选出至多一张通过加法和减法（找钱）来构成，（比如：1+2=3，5−1=4，5+1=6，5+2=7，1+2+5=8，10−1=91+2=3，5−1=4，5+1=6，5+2=7，1+2+5=8，10−1=9）

但是实际上，我们只需要1、2、71、2、7三种面值就可以组成1−101−10的每一个数字了

（1+2=3，7−1−2=4，7−2=5，7−1=6，7+1=8，7+2=9，7+1+2=101+2=3，7−1−2=4，7−2=5，7−1=6，7+1=8，7+2=9，7+1+2=10）

那么现在问题来了，给一个数nn，请问最少需要多少种不同的面值就可以构成从1−n1−n的所有数字，注意在构成每一个数字时同种面值不能超过11张。

**Input**

一个数字nn（1<=nn<=100000）

**Output**

一个数字，代表最少需要多少种不同的面值可以构成从1−n1−n的所有数字。

**Sample Input**

10

**Sample Output**

3

从1、2、7来找规律就可以看出（但实际这个题的前三个数不是127而是139）

1、2、7可以表示1+2+7=10以下的所有的数，而要表示一个数，既可以加和又可以作差

，所以在两倍的1+2+7也就是20，再加个1=21以内的数，超过了10的可以作差来减得到，

21-10=11,21-9=12······所以，当你目前的数不能表示n了的时候，加的新数的值是当前所有数之和+1

#include<iostream>

#include<stdio.h>

using namespace std;

int main()

{

int n,t(0),sum(0),cnt(0);

cin>>n;

while(sum<n)

{

t=sum\*2+1;

sum+=t;

cnt++;

}

cout<<cnt;

return 0;

}