**题目描述**

蓝桥历届试题  带分数      
时间限制：1.0s     内存限制：256.0MB   
        
问题描述   
100  可以表示为带分数的形式：100  =  3  +  69258  /  714。   
还可以表示为：100  =  82  +  3546  /  197。   
注意特征：带分数中，数字1~9分别出现且只出现一次（不包含0）。   
类似这样的带分数，100  有  11  种表示法。   
输入格式   
从标准输入读入一个正整数N  (N< 1000\*1000)   
输出格式   
程序输出该数字用数码1~9不重复不遗漏地组成带分数表示的全部种数。   
注意：不要求输出每个表示，只统计有多少表示法！   
样例输入1   
100    
样例输出1   
11    
样例输入2   
105    
样例输出2   
6

思路：  
首先，这个题用暴力枚举一定会超时的，所以我就没试。

为何暴力枚举会超时？

原因在于，暴力枚举会搜索到很庞大的没有用的数据，最后在十几万甚至几百万个组合中，也许仅仅只有十几种  
组合符合条件，这就大大的浪费了时间。想要避免这类事件的发生，就要有好的剪枝条件。

如何建造好的剪枝条件？

本题说的是，n=a+b/c；那么首先a一定是小于n的，又因为n为整数，所以a和b/c都是整数，这就要求

b/c一定可以整除，所以b%c=0，b/c还要满足可除条件，即b>=c。剪枝的三个条件已经确定  
(1).a<n；

(2).b%c=0；

(3).b>=c

再加上n=a+b/c就是四个条件了。只要在1至9的全排列中选取满足这四个条件的全排列就是所求的结果之一。

那么在1至9的全排列（9个数字）中如何确定a，b，c的取值范围呢？

a前面已经说过，而又知道，b一定大于或等于c，则b的取值范围一定在a选择过后去选择剩下的一半或一半以上的数据。举个例子，1至9的其中一个全排列--156987423，若a选择156，则b只能选择剩下的987423中的一半或  
一半以上，如987、9874、98742。如果b小于剩下的一半，那么一定不满足除法（如98/7432）。c  
的范围则是a和b选择剩下的所有了。这样我们就可以判定，假设num=9，a选择9位中的前n位，那  
么b的结尾选择范围为第n+（num-n）/2至num-1位数字（结尾为一半或一半以上，最多时到num-1  
，给c留一个数字）；  
那么利用深度优先搜索（用来得到一个9位的全排列）和适当的判断（剪枝，找出符合3个条件并  
且满足n=a+b/c的全排列）就可以解决。

先把9个数的排列搜出来

再搜哪些数字是作为m1，哪些是m2，哪些是m3

n=m1+m2/m3

#include <iostream>

#include <cstdio>

#include <cstdlib>

#include <cstring>

#include<vector>

#include<algorithm>

#define INF 1000000007

using namespace std;

typedef long long ll;

int n;

bool b[20];

int a[20];

int ans(0);

int sum(int x,int y)

{

int sum(0);

for(int i=x;i<=y;i++)

sum=sum\*10+a[i];

return sum;

}

void pd()

{

int m1,m2,m3;

for(int i=1;i<=9;i++)

{

m1=sum(1,i);

if(m1>n)return;

for(int j=i+(10-i)/2;j<9;j++)

{

m2=sum(i+1,j);

m3=sum(j+1,9);

if(m2>m3 && m2%m3==0 && n==m1+m2/m3)

{

ans++;

}

}

}

}

void dfs(int start)

{

if(start==10)

{

pd();

return;

}

for(int i=1;i<=9;i++)

if(b[i]==0)

{

a[start]=i;

b[i]=1;

dfs(start+1);

b[i]=0;

}

}

int main()

{

//freopen("input.txt","r",stdin);

cin>>n;

for(int i=0;i<20;i++)b[i]=0;

ans=0;

dfs(1);

cout<<ans<<"\n";

return 0;

}