C

#include <stdio.h>

#include <stdlib.h>

typedef struct Interval

{

int pre;

int rear;

int satisfy;

}Interval;

Interval interval[7][5];

int count=0;

//初始化

void Init()

{

int i,j;

int value;

for(i=1;i<7;i++)

{

value=i;

for(j=1;j<5;j++)

{

interval[i][j].pre=value++;

interval[i][j].rear=value;

}

}

}

//数组初始化为0

void InitZero(int \*sign)

{

int i;

sign[0]=1;

for(i=1;i<10;i++)

{

sign[i]=0;

}

}

//将一个数的各个位上拆分，并在相应的位上赋值1

int Split(int \*sign,int value)

{

int index;

while(value)

{

index=value%10;

if(sign[index]==0) sign[index]=1;

else return 1;

value/=10;

}

return 0;

}

//计算一个数的位数

int CountBit(int value)

{

int n=0;

while(value)

{

n++;

value/=10;

}

return n;

}

//将一个整型数组转换成一个整数

int CreateInteger(int \*data,int n)

{

int i;

int value=0;

for(i=0;i<n;i++)

{

value=value\*10+data[i];

}

return value;

}

//检查是否每个数都用到

int Check(int \*sign)

{

int i;

for(i=1;i<10;i++)

{

if(sign[i]==0) return 0;

}

return 1;

}

//复制

void Copy(int \*sign,int \*temp\_sign)

{

int i;

for(i=0;i<10;i++)

{

temp\_sign[i]=sign[i];

}

}

//创建一个n位数的整数

void CreateNBitNumber(int \*sign,int \*data,int n,int m,int value,int value3)

{

if(n==m)

{

int value1=CreateInteger(data,n);

int value2=value1\*value;

int temp\_sign[10];

Copy(sign,temp\_sign);

if(!Split(temp\_sign,value2) && Check(temp\_sign))

{

count++;

}

}

else

{

int i;

for(i=1;i<10;i++)

{

if(sign[i]==0)

{

sign[i]=1;

data[m]=i;

CreateNBitNumber(sign,data,n,m+1,value,value3);

sign[i]=0;

}

}

}

}

//求出解

void Create(int value)

{

int i,j;

int sign[10];

int result;

int result\_n;

int n;

for(i=3;i<value;i++)

{

InitZero(sign);

if(Split(sign,i)) continue;

result=value-i;

result\_n=CountBit(result);

n=CountBit(i);

for(j=1;j<5;j++)

{

if( ((interval[result\_n][j].pre+j)==(9-n)) || ((interval[result\_n][j].rear+j)==(9-n)))

{

int data[5];

CreateNBitNumber(sign,data,j,0,result,i);

}

}

}

}

int main()

{

int value;

scanf("%d",&value);

Init();

Create(value);

printf("%d\n",count);

return 0;

}

C++:

#include<iostream>

#include<cstring>

using namespace std;

int N,ans,digit,flag,full[9];

void div(int m)

{

while(m)

{

if(m%10!=0)

full[m%10-1]=1;

m/=10;

digit++;

}

}

bool check(int \*f)

{

int j;

for(j=0;j<9;j++)

if(!f[j])

return false;

return true;

}

void DFS(int a,int b,int c)

{

digit=0;

flag=0;

div(a);

div(b);

div(c);

if(digit>9){memset(full,0,sizeof(full));return ;}

if(check(full)){ans++;memset(full,0,sizeof(full));}

else memset(full,0,sizeof(full));

DFS(a,(c+1)\*b/c,c+1);

return ;

}

int main()

{

int i;

while(cin>>N)

{

ans=0;

memset(full,0,sizeof(full));

for(i=2;i<N-1;i++)

DFS(i,N-i,1);

cout<<ans<<endl;

}

return 0;

}

Java：

import java.util.Scanner;

/\*\*

\* http://lx.lanqiao.org/problem.page?gpid=T26

\* Created by revintec on 14-5-5.

\*/

public class Main{

static int getBits(int k){

int bits=0;

while(k>0){

int a=k%10;k/=10;

if(a==0) return -1;

int b=1<<a;

if((bits&b)!=0) return -1;

bits|=b;

}return bits;

}

static int getBits2(int b){

int lo=9;

for(int i=1;i<1<<10;i<<=1)

if((b&i)!=0) --lo;

return lo;

}

public static void main(String[] args){

// System.clearProperty("m.TEST");

Scanner sc=System.getProperty("m.TEST")==null?new Scanner(System.in):

new Scanner("999988");

int sx=sc.nextInt();

int[] ts={0,

10,

100,

1000,

10000,

100000,

1000000,

10000000,

100000000,

1000000000,

};

int so=0;

for(int a=1;a<sx;++a){

int bits=getBits(a);

if(bits==-1) continue;

for(int b=1;;++b){

int bts=getBits(b);

if((bits&bts)!=0) continue;

bts|=bits;

long c=b\*(sx-a);

if(c>=ts[getBits2(bts)]) break;

if((getBits((int)c)^bts)==1022)

++so;

}

}System.out.println(so);

}

}