АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА
СТРАННИ ЧИСЛА

Разглеждаме $2n$ цифрените числа, които трябва да притежават описаното свойство. Да означим числото, съставено от левите n цифри (в същия ред), с a, а това, съставено от десните n цифри – с b. Сега можем да запишем някои математически преобразувания:

$$\overline{ab} = 10^n a + b = (a+b)^2$$

$$a^2 - (10^n - 2b)a + b^2 - b = 0$$

$$a^2 - 2(2^{n-1}5^n - b)a + b^2 - b = 0$$

$$D = (2^{n-1}5^n - b)^2 - b^2 + b = 2^{2n-2}5^{2n} - 2.2^{n-1}5^n b + \cancel{b^2} - \cancel{b^2} + b =$$

$$= \frac{10^{2n}}{4} - (10^n - 1)b = t^2 \ge 0$$

$$b \le \frac{10^{2n}}{4.(10^n-1)} = \frac{10^{2n}-1+1}{4.(10^n-1)} = \frac{\cancel{(10^n-1)}(10^n+1)}{4\cancel{(10^n-1)}} + \frac{1}{4(10^n-1)} \le \frac{10^n}{4}$$

Само полученото ограничение за b може да е достатъчно за намиране на всички до 12-цифрени търсени числа. За по-нататъшно ускоряване на изчислителния процес ще трябва да направим още наблюдения и да впрегнем резултатите от тях. Както се вижда от записа, b е цяло положително число с не повече от n цифри, представляващо „опашка” на някакъв квадрат. Различните такива „опашки” се получават чрез повдигане на квадрат на число с не повече от n цифри и запомняне на остатъка му по модул 10^n. Ние не можем да си позволим да създаваме и помним всички n-цифрени „опашки” за големи n. Но можем да съобразим, че ако b е n-цифрена „опашка” на някой квадрат, то последните му k цифри пък са k-цифрена „опашка” на квадрат за всяко k от 1 до n. Можем да разгледаме, например, k=5. Петцифрените опашки се пресмятат бързо и се оказват по-малко от 10000. Така „кандидатите” за b можем да получим чрез „дописване отляво” към всяка от „опашките” на всевъзможните допълнения. Ако и дискриминантата D е точен квадрат, то квадратното уравнение има целочислени решения (каквито ни трябват) относно a, а именно $a_1 = 2^{n-1}5^n - b + \sqrt{D}$ и $a_2 = 2^{n-1}5^n - b - \sqrt{D}$. Всяко положително n-цифрено от тях поражда различно $2n$ цифрено число $\overline{ab}$ (b се допълва, ако е нужно, до n цифри с помощта на водещи нули) с търсените свойства.

Задачата допуска разнообразни специални наблюдения и атака с по-слаби алгоритми за частични решения.

Генериране на странните числа

```
#include <iostream>
#include <cmath>
using namespace std;
const int Nsq=100000;
long sq5[10000];
long sqCnt=0;
bool isSq(long long &a)
{long long s=(long long)ceil(sqrt(a));
 if (a==s*s){a=s;return true;}
 return false;
}
void makeSq()
{long long i;
 bool sq[Nsq]={false};
 for (i=0;i<Nsq;i++)
 {long t=(i*i)%Nsq;
```

```
  if(!sq[t]) {sq[t]=true;sq5[sqCnt++]=t;}
 }
}
void gen(int n)
{long long a,b,d=1,d2,d5=100000,dd=10,c,t;
 int i;
 for (i=0;i<n;i++)
 {d*=10;
  if (i<n-6) dd*=10;
 }
 d2=(d*d)>>2;
 d--;
 for (c=0;c<=(dd>>2);c++)
  for (i=0;i<sqCnt;i++)
  {b=c*d5+sq5[i];
   t=d2-d*b;
   if (isSq(t))
   {a=((d+1)>>1)-b+t;
    if (a>=(d+1)/10 && a<=d)
    {cout<<a;
     cout.width(n);
     cout.fill('0');
     cout<<b<<endl;
    }
    a=((d+1)>>1)-b-t;
    if (a>=(d+1)/10 && a<=d)
    {cout<<a;
     cout.width(n);
     cout.fill('0');
     cout<<b<<endl;
    }
   }
  }
}
int main(void)
{ makeSq();
  for (int n=1;n<10;n++) gen(n);
  return 0;
}
```

Автор: Павлин Пеев