

ЕСЕНЕН ТУРНИР ПО ИНФОРМАТИКА

Шумен, 10–12 ноември 2006 г.

Тема за група С (8-9 клас)

Задача за С1. ПРОСТОТА ОТДЯСНО

Десетичният запис на естествените числа крие много любопитни свойства. Да наречем „просто отдясно” такова просто число, в десетичния запис на което:

- ако има повече от една цифра и изтрием цифрата на единиците – пак се получава просто число;
- ако полученото има повече от една цифра и направим същото – пак остава просто;
- и все така, докато остане само една цифра (и тя пак е просто число)!

Има ли „прости отдясно” числа? Да, вижте, например, числото 7193. Самото то е просто. Но прости са и всички числа, които се получават от него чрез описания по-горе процес – това са 719, 71 и 7. За да няма недоразумения, ще припомним, че числото 1 не е просто. За числото 5, например, също ще считаме, че е „просто отдясно” – то е просто и едноцифрено и за него описаният процес не започва (има нула стъпки).

Задачата Ви, която ще реализирате с програмата **rprimes**, се състои в преброяване на „простите отдясно” числа в зададен интервал $[a, b]$.

От стандартния вход се въвежда един ред с естествените числа a и b , разделени с интервал, като $1 \leq a \leq b \leq 2000000000$. Изведете на стандартния изход един ред с едно неотрицателно цяло число, равно на броя на простите числа с търсеното свойство, не по-малки от a и не по-големи от b .

ПРИМЕР

Вход

7 300

Изход

13

Обяснение на изхода: В интервала $[7, 300]$ „простите отдясно” числа са: 7, 23, 29, 31, 37, 53, 59, 71, 73, 79, 233, 239 и 293.

Решение:

Оказва се, че простите числа с описаното свойство са краен (и даже – малък) брой – всичко на всичко 83, като най-малкото от тях е 2, а най-голямото е 73939133. Генерирането им става лесно и бързо, което прави ненужно предварителното им изчисляване (но и това е допустимо на състезание): започвайки от списъка $\{2, 3, 5, 7\}$ посочваме първото от тях и му долепваме отдясно всяка една от цифрите 1, 3, 7 и 9. Проверяваме всяко получено число за простота. Ако полученото е просто – добавяме го в края на списъка. Посочваме следващото в списъка и повтаряме процеса. Той приключва, когато показалецът надхвърли големината на текущия списък. Понеже числата се генерират наредени в нарастващ ред, броенето е лесно: прескачаме тези от списъка, които са по-малки от a и броим по-нататък в списъка тези, които са не по-големи от b .

Примерна реализация на С:

```

#include <stdio.h>
#include <math.h>

unsigned long a,b,rprimes[1024]={2,3,5,7};
int count=4;

int isPrime(unsigned long a)
{unsigned long p,d;
  if (a==1) return 0;
  if (a==2) return 1;
  if (!(a&1)) return 0;
  d=ceil(sqrt(a));
  p=3;
  while (p<=d)
  {if (!(a%p)) return 0;
   p+=2;
  }
  return 1;
}

void makeRPrimes(void)
{int p=0;
  unsigned long d;
  do
  {d=10*rprimes[p]+1;
   if (isPrime(d)) rprimes[count++]=d;
   d+=2;
   if (isPrime(d)) rprimes[count++]=d;
   d+=4;
   if (isPrime(d)) rprimes[count++]=d;
   d+=2;
   if (isPrime(d)) rprimes[count++]=d;
   p++;
  } while (p<count);
}

int main (void)
{int c=0,i;
  scanf("%lu %lu",&a,&b);
  makeRPrimes();
  for (i=0;i<count&&rprimes[i]<a;i++);
  for (;i<count&&rprimes[i]<=b;i++,c++);
  printf("%d\n",c);
  return 0;
}

```

Примерна реализация на Pascal:

```

Var
  a,b:LongInt;

```

```

rprimes: Array [0..1023] of LongInt;
count,c,i:Integer;

Function isPrime(a:LongInt):Boolean;
Var p,d:LongInt;
Begin
  if a=1 then Begin isPrime:=FALSE; Exit; End;
  if a=2 then Begin isPrime:=TRUE; Exit; End;
  if Not odd(a) then Begin isPrime:=FALSE; Exit; End;
  d:=round(sqrt(a));
  p:=3;
  while p<=d do
  begin
    if a mod p=0 then Begin isPrime:=FALSE; Exit; End;
    p:=p+2;
  end;
  isPrime:=TRUE;
End;

procedure makeRPrimes;
Var p:Integer;
    d:LongInt;
Begin
  p:=0;
  Repeat
    d:=10*rprimes[p]+1;
    if isPrime(d) then Begin rprimes[count]:=d; Inc(count); End;
    d:=d+2;
    if isPrime(d) then Begin rprimes[count]:=d; Inc(count); End;
    d:=d+4;
    if isPrime(d) then Begin rprimes[count]:=d; Inc(count); End;
    d:=d+2;
    if isPrime(d) then Begin rprimes[count]:=d; Inc(count); End;
    Inc(p);
  Until p=count;
End;

BEGIN
  rprimes[0]:=2;rprimes[1]:=3;
  rprimes[2]:=5;rprimes[3]:=7;
  count:=4;
  ReadLn(a,b);
  makeRPrimes;
  c:=0;
  for i:=0 to count-1 do if rprimes[i]>=a Then break;
  for i:=i to count-1 do if rprimes[i]>b then break else Inc(c);
  WriteLn(c);
END.

```

Павлин Пеев