

1

{

УЧИТЕЛЮ моему – Сергею Петровичу Митрофанову,
искренне выражая безмерную благодарность
за встречу с ним в моей жизни,
посвящаю.

File: degree.pas

Compiler. Borland Pascal 7.0

Задача. Для того чтобы проверить, как ее ученики умеют считать, Мария Ивановна каждый год задает им на дом одну и ту же задачу – для заданного натурального A найти минимальное натуральное N такое, что N в степени N делится на A . От года к году меняется только число A .
Во входном файле `Input.txt` содержится единственное число A ($1 \leq A \leq 10^9$).
В выходной файл `Output.txt` необходимо вывести единственное число N .

Input.txt	Output.txt
8	4
13	13

Решение. Калмыков Вадим (ProCrypt),
г. Сургут, ЦНИТ "Северная Звезда",
23:21, 22.09.2008

Source : <http://acm.dvpion.ru>

}

29

{

Разбор задачи

Согласно основной теореме арифметики, которая гласит, что любое натуральное A можно представить в виде произведения простых чисел, входящих в каноническое разложение A в разных степенях, имеем:

$$A = p_1^{k_1} * p_2^{k_2} * \dots * p_n^{k_n}, \text{ где } p_i - \text{ простое число}$$

(простым называется число, которое не имеет делителей, кроме себя самого и единицы)

k_i – степень, в которой p_i входит в разложение A .

i

Понятно, что если $A|N$ (A делит N), то $N = A * q$, q – какое-то целое число, следовательно, имеем

$$N = p_1^{k_1} * p_2^{k_2} * \dots * p_n^{k_n} * q,$$

это означает, что все простые числа, входящие в разложение A с разными степенями также входят в разложение N^N .

Для нахождения N мы поступим следующим образом: перемножим все простые числа из разложения A , но возьмем их все в первой степени.

Таким образом, получим число, содержащее все простые множители A . Затем, чтобы наше число делилось на A , необходимо чтобы все сомножители встретились m раз, где m – самая большая степень среди входящих в разложение простых чисел, например,

$$24 = 2^3 * 3, \text{ тройка входит в первой степени (один раз), двойка в третьей (три раза), следовательно } m = 3$$

После чего мы получим число d , которое будет делиться на A , где d – произведение всех простых множителей A , взятых в первой степени.

Но при этом должно выполняться условие $d = m$, а если это не так? Тут может быть несколько вариантов:

- 1) $d > m$, в этом случае будет ответ d , так как d будет делиться на A (объясните почему).
- 2) $d < m$, в этом случае нам остается только увеличивать d , но как?

Больше всего раз в разложении числа A встречается число u . Количество повторений u равно m . Нам необходимо найти такое число $d * x$, что количество повторений в нем числа u будет больше или равно m . Но тут возникает вопрос, как мы найдем это число x ?

Ответ прост. Мы будем перебирать x от 2 до тех пор, пока не достигнем результата: каждый раз будем умножать d на x и смотреть, сколько раз встречается число u в нем.

Пусть $F = d * x$, количество повторений числа u равно y .

Имеем число F , где в одном F u встречается y раз. Раз мы умножаем F на себя F раз, то количество повторений u

в числе F^F будет равно $F * y$

Таким образом будем перебирать x , чтобы выполнилось неравенство

$$F * y \geq m.$$

Рассмотрим на примере:

$A = 32 = 2^5$, значит $d = 2$, $m = 5$
 Найдем $F = d * x$ (начинаем перебирать от $x = 2$), тогда

$$1$$
 получаем $F = 2 * 2 = 4$, $u = 2$, $y = 2$
 Общее количество повторений u равно
 $y * F = 2 * 4 = 8 > m$ ($m = 5$)

Следовательно, ответ: 4

}

96 `Program Degree;`

97 `Var`

98 `i, { очередной простой делитель A }`

99 `A : LongInt; { число из входного файла }`

100 `x, { счетчик }`

101 `u, { делитель, входящий в самой большое степени
 в разложение A }`

103 `j, { степень, в которой входит i в разложение A }`

104 `N, { искомое число }`

105 `m : LongInt; { самая большая степень в разложении }`

106 {

`FDiv - First Diviser`

`Функция возвращает первый делитель числа, больший единицы.`

`Поиск делителей начинается от левой границы left`

Как будем искать первый делитель? Метод перебора будет очень долгим, если число A будет простое, так как цикл выполнится A раз. Можно заметить, что при делении на самое маленькое простое число, то есть на 2 мы получим самый большой делитель A . Это означает, что все делители A меньше либо равны $A / 2$.

Но при больших ограничениях A мы все равно не пройдем тесты.

Необходимо искать другое решение.

Пусть A – составное число. Тогда $A = p * q$, где p – наименьший простой делитель A , то есть выполняется неравенство $q \geq p$

Мы имеем систему:

$$A = p * q,$$

$$q \geq p;$$

Умножая почленно неравенство на первое равенство, получим:

$$A * q \geq p^2 * q; \text{ делим обе части на } q$$

$$A \geq p^2, \text{ следовательно } p \leq \text{Sqrt}(A)$$

Это неравенство свидетельствует о том, что если число A не имеет

делителей меньших либо равных квадратному корню из A, то число A является простым.

Поэтому будем искать до корня из A.

}

```
130 Function FDiv (A, left : LongInt) : LongInt;
```

```
131 var
```

```
132   d,           { первый делитель }
```

```
133   i : LongInt; { счетчик }
```

```
134 begin
```

```
135   {
```

```
    первым делителем любого числа является 1, но 1 не является ни
    простым числом, ни составным
```

```
  }
```

```
139   d := 1; { инициализация }
```

```
140   for i := left to Trunc (Sqrt (A)) do
```

```
141     if (A mod i = 0)
```

```
142       then
```

```
143         begin
```

```
144           d := i; { нашли делитель }
```

```
145           Break; { выходим из цикла }
```

```
146         end;
```

```
147     { если d <> 1, то число A составное и d - первый делитель }
```

```
148     if (d <> 1)
```

```
149       then
```

```
150         FDiv := d
```

```
151       else { в противном случае число A - простое }
```

```
152         FDiv := A;
```

```
153     end;
```

```
154   {
```

```
    Функция вычисляет в какой степени u входит в разложение числа N
```

```
    Описание этой функции значительно проще, чем предыдущей. Сколько
    раз удастся нацело разделить N на u, значит столько раз u
    встречается в разложении N.
```

```
  }
```

```
160 Function g (N, u : LongInt) : LongInt;
```

```
161 var
```

```
162   i : LongInt; { счетчик }
```

```
163 begin
```

```
164   i := 0; { инициализация }
```

```
165   while (N mod u = 0)
```

```
166     do
```

```

167     begin
168         Inc (i);
169         N := N div u;
170     end;

171     g := i;
172 end;

173 Begin
174     Assign (Input, 'input.txt');
175     Assign (Output, 'output.txt');

176     Reset (Input);
177     Read (A);
178     Close (Input);

179     m := 0;
180     N := 1; { инициализация }

181     { первый делитель A ищем от первого простого числа }
182     i := FDiv (A, 2);
183
184     u := 2; { первое простое число }
185     while (A > 1)
186     do
187     begin
188         {
189             делитель обязательно будет найден, в крайнем случае само
190             число A
191         }
192         N := N * i; { умножаем N на очередной простой делитель A }

193         { избавляемся от сомножителя i }
194         j := 0; { инициализация }
195         while (A mod i = 0)
196         do
197             begin
198                 Inc (j);
199                 A := A div i;
200             end;

201         if (j > m)
202         then
203             begin
204                 m := j; { в этом случае j максимальная степень }
205                 u := i; { непосредственно сам делитель }
206             end;

207         {
208             Теперь нужно искать новое просто число. После того, как

```

найденно было простое число i , то следующее будет обязательно больше его, поэтому гораздо эффективнее будет искать делители большие i . Так как число i – простое, то оно нечетное. Значит $i + 1$ будет обязательно четным числом, следовательно, нам нужно следующее за i нечетное число, то есть $i + 2$

```
    }
215     i := FDiv (A, i + 2);
216   end;

217   { берем x = 1 для более простого решения с технической стороны }
218   x := 1;
219
220   while (g (N * x, u) * N * x < m)
221   do
222     Inc (x);

223   N := N * x; { искомое решение }

224   Rewrite (Output);

225   Write (N);

226   Close (Output);
227   End.

228   {
    За все время, которое автор занимается программированием, ему еще
    не встречалась задача столь интересная и занимательная, именно
    поэтому он считает данную работу лучшей из когда-либо им
    сделанных.

    Калмыков В. В.

    26 сентября 2008 года
  }
```

Listing данной задачи опубликован в сети Internet по адресу
<http://www.Best-Listing.ru/color-10-task-644.html>

Sergey Mitrofanov, 09.09.14, 17:15

E-mail: infostar@mail.ru

© <http://www.Best-Listing.ru/>, 2006–2014