

Прайморадичная система счисления. 10-11 класс. Заключительный тур

В задаче рассмотрим прайморадичную систему счисления, в которой будем записывать неотрицательные целые числа. Это смешанная система счисления, в которой основаниями являются праймориалы $p_i\#$: 1, 2, 6, 30, 210, 2310, ... , где $i = 1, 2, 3, \dots$. Каждый праймориал является произведением вроде факториала, но множителями в нём являются подряд идущие простые числа: $p_n\# = 1 * 2 * \dots * p_i * \dots * p_n$, где p_i – i -ое простое число, а $p_1 = 1$ (в нашей нумерации $p_2 = 2$, так как мы начали последовательность p_i с 1, не являющейся простым числом). Так праймориал $p_5\# = p_1 * p_2 * p_3 * p_4 * p_5 = 1 * 2 * 3 * 5 * 7 = 210$. Запись числа в прайморадичной системе делится на позиции, разделённые двоеточиями :. Позиции нумеруются справа налево. Самая правая позиция имеет номер 1 и правее неё двоеточие не ставится. Вместо этого снизу приписывается знак #, отмечающий, что это прайморадичная запись. На k -ой позиции допускается указывать десятичное число d_k , такое что $0 \leq d_k \leq p_{k+1} - 1$. На первой справа позиции может быть указан либо 0, либо 1. На второй справа позиции может стоять либо 0, либо 1, либо 2. На третьей справа позиции может быть либо 0, либо 1, либо 2, либо 3, либо 4. И так далее. Допускается наличие незначащих нулей в левых позициях записи числа. Незначащим является любой нуль, находящийся левее первой встреченной при чтении слева направо ненулевой цифры, или, если записано нулевое число, то все нули, кроме самого правого. Чтобы понять, какое число записано, нужно выполнить вычисления: $d_n : d_{n-1} : \dots : d_i : \dots : d_2 : d_1\# = d_n * p_n\# + d_{n-1} * p_{n-1}\# + \dots + d_i * p_i\# + \dots + d_2 * p_2\# + d_1 * p_1\#$. Например, $6 : 3 : 0 : 1\# = 6 * p_4\# + 3 * p_3\# + 0 * p_2\# + 1 * p_1\# = 6 * 30 + 3 * 6 + 0 * 2 + 1 * 1 = 180 + 18 + 0 + 1 = 199_{10}$. То же самое число может быть записано с незначащими нулями: $0 : 0 : 6 : 3 : 0 : 1\#$. Заметим, что значащая позиция прайморадичной записи не содержит незначащих нулей.

Составьте программу, принимающую на вход в первой строке десятичное число N – положительное натуральное число ($1 \leq N \leq 4000$) – длину последовательности, а в последующих N строках – прайморядичные записи чисел X_i без завершающего знака $\#$, $1 \leq i \leq N$. Известно, что в каждой записи числа X_i не более чем 65 позиций. Программа определяет наибольший праймориал $p_k\#$, на который делится без остатка хотя бы одно из чисел последовательности. Если в последовательности X_i встречается нулевое число, то $p_k\#$ считается неопределённым, так как нуль делится без остатка на все праймориалы. Затем программа рассматривает только те числа последовательности X_i , которые делятся нацело на $p_k\#$ (или только нули, если они есть в последовательности). Среди таких чисел программа находит максимум, а затем выводит номера тех чисел, которые равны найденному условному максимуму. Программа выводит все найденные номера в порядке возрастания, построчно, т. е. по одному номеру в одной строке.

Формат ввода: В первой строке содержится десятичное число N — длина последовательности ($1 \leq N \leq 4000$). В следующих N строках содержатся прайморадичные записи чисел X_i без завершающего знака $\#$, $1 \leq i \leq N$. В каждой такой записи не более чем 65 позиций, разделённых двоеточиями $:$. В каждой позиции неотрицательное целое десятичное число, допустимое правилами прайморадичной системы.

Формат вывода: Выводятся по возрастанию в формате беззнакового десятичного натурального числа номера искоемых чисел X_i . Каждый номер выводится в отдельной строке.

Вывод примера №1:

1

Вывод примера №2:

3

```

:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0

```

12:40:36:30:28:22:18:16:12:10:6:4:0:0

42:40:36:30:28:22:18:16:12:10:6:4:2:0

42:40:36:30:28:22:18:16:12:10:6:4:0:0

Решение

В решении можно запрограммировать следующие подзадачи: 1) Считывание очередного числа и представление его в виде строки из 195 символов с незначащими нулями, дополняющими запись прайморадического числа до 65 позиций, а каждую считанную позицию записи – до трёх цифр (в такой строке позиции идут подряд без разделения :). В ходе считывания определяется, является ли число 0, а если не является, то на сколько нулей заканчивается число. 2) поиск всех условных *argmax*-ов последовательности, основанный на посимвольном сравнении строк, эквивалентном поразрядному сравнению считанных чисел. Для решения второй подзадачи достаточно одного прохода по последовательности, в котором совмещены построчный ввод чисел и их обработка. Следует хранить текущий рекорд (максимальное количество нулей, встреченных в конце какого-либо введённого числа (65, если было встречено нулевое число), максимальное среди всех чисел, заканчивающихся на рекордное количество нулей (или равных 0), которые программа успела считать, и номера всех чисел, равных текущему рекордному числу последовательности. Количество нулей в хвосте очередного считанного числа сравнивается с рекордным количеством нулей. При равенстве очередное число сравнивается с рекордным числом. При равенстве номер очередного числа добавляется к хранящимся номерам. Если очередное число меньше, то делается переход к обработке следующего числа. Если очередное число больше, то оно становится рекордом, коллекция номеров опустошается и в неё записывается номер очередного числа. Если количество нулей в хвосте очередного числа меньше, чем рекорд, то делается переход к обработке следующего числа. Если оно больше, то рекорды обновляются, коллекция номеров опустошается и в неё записывается номер очередного числа. По окончании обработки выводится собранная коллекция номеров.

Код возможного решения

```
program PRIMORADICS1011 (input, output);
type    primoradics = array [1..195] of char;
        numbers = record quantity : word; zeros : byte;
                      collection : array [1..4000] of word end;
var  CURMAX, CURNUM : primoradics; N, I, J, Z : word; CHECK : integer;
     CURANSWER : numbers;
procedure readnumber(var P : primoradics; var zeros : word);
var    S : string; I, J, K, L : byte;
begin  readln(S); J := 195; I := Length(S);
       while (I > 0) and (J > 0) do begin
         K := 3;
         while (I > 0) and (S[I] <> ':') and (J > 0) and (K > 0) do begin
           P[J] := S[I];
           I := I - 1;
           J := J - 1;
           K := K - 1;
         end;
         while (I > 0) and (S[I] = ':') and (J > 0) and (K > 0) do begin
           P[J] := '0';
           J := J - 1;
           K := K - 1;
         end;
         if (I > 0) and (S[I] = ':') then I := I - 1;
       end;
       for I := J downto 1 do P[I] := '0';
       J := 195;
       while (J > 2) and (P[J] = '0') and (P[J - 1] = '0') and (P[J - 2] = '0')
         do J := J - 3;
       zeros := 65 - J div 3;
end;
begin  readln(N);
       readnumber(CURMAX, Z);
```

```

with CURANSWER do begin
  quantity := 1;
  zeros := Z;
  collection[1] := 1;
  for I := 2 to N do begin
    readnumber(CURNUM, Z);
    CHECK := CompareChar(CURMAX, CURNUM, 195);
    if (CURANSWER.zeros < Z) or (CURANSWER.zeros = Z) and (CHECK < 0)
      then begin quantity := 1; zeros := Z; collection[1] := I;
              CURMAX := CURNUM end
    else if (CURANSWER.zeros = Z) and (CHECK = 0) then
      begin quantity := quantity + 1; collection[quantity] := I end;
    end;
    for I := 1 to quantity do writeln(collection[I]);
  end;
end.

```