

Прайморадичная система счисления. 7-9 класс. Заключительный этап

В задаче рассмотрим прайморадичную систему счисления, в которой будем записывать неотрицательные целые числа. В привычной позиционной системе, например, десятичной запись числа основывается на единственном его представлении в виде суммы попарных произведений цифр на степени 10: $d_n d_{n-1} \dots d_i \dots d_2 d_1 = d_n * 10^{n-1} + d_{n-1} * 10^{n-2} + \dots + d_i * 10^{i-1} + \dots + d_2 * 10^1 + d_1 * 10^0$, $0 \leq d_k \leq 10 - 1$. В смешанных системах счисления похожее представление имеет место, но используются не степени, а другие последовательности – числа Фибоначчи, факториалы и прочее.

Прайморадичная система счисления – смешанная, и в ней 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 #, помечающий, 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, то есть, записывая d_k , не используют излишние нули слева.

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 – прайморадичные записи чисел X_i без завершающего знака #, $1 \leq i \leq N$. Известно, что в каждой записи числа X_i не более чем 25 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 количество тех чисел X_i , которые равны найденному условному максимуму. Программа выводит в единственной строке искомое количество вхождений условного максимума в последовательность, записанное десятичным натуральным числом без знака.

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 25 позиций, разделённых двоеточиями :. В каждой позиции неотрицательное целое десятичное число, допустимое правилами прайморадичной системы.

Формат вывода: Выводится беззнаковое десятичное натуральное число, равное количеству искомых чисел.

Ввод примера №1:

1
4:3:2:1

Вывод примера №1:

1

3

```
1:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0
```

0

2

3

96:88:82:78:72:70:66:60:58:52:46:42:40:36:30:28:22:18:16:12:10:6:0:2:0

Вывод примера №3:

2

В решении можно запрограммировать следующие подзадачи: 1) Считывание очередного числа и представление его в виде строки из 50 символов с незначащими нулями, дополняющими запись прайморадичного числа до 25 позиций, а каждую считанную позицию записи – до двух цифр (в такой строке позиции идут подряд без разделения :). Подсчёт количества нулей, завершающих запись введённого числа. 2) Подсчёт количества всех элементов последовательности, равных условному максимуму, 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 (25, если было встречено нулевое число), максимальное среди всех чисел, заканчивающихся на рекордное количество нулей (или равных 0), которые программа успела считать) и количество 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 искомое количество чисел увеличивается на 1. Если очередное число меньше, то делается переход к обработке следующего числа. Если очередное число больше, то оно становится рекордом, искомое количество приравнивается 1. Если количество нулей в хвосте очередного числа меньше, чем рекорд, то делается переход к обработке следующего числа. Если оно больше, то рекорды обновляются, искомое количество чисел приравнивается 1. По окончании обработки выводится найденное количество чисел.

Код возможного решения

```

program PRIMORADICS79 (input, output);
type    primoradics = array [1..50] of char;
        answer = record quantity : word; zeros: word; max : primoradics end;
var  CURNUM : primoradics; N, I, Z : word; CHECK : integer; CURANSWER : answer;
procedure readnumber(var P : primoradics; var Z : word);
var    S : string; I, J, K : byte;
begin  readln(S); J := 50; I := Length(S);
        while (I > 0) and (J > 0) do begin
            K := 2;
            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;
            if (I > 0) and (S[I] = ':') and (J > 0) and (K > 0) then 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 := 50;
end;

```

```

        while (J > 1) and (P[J] = '0') and (P[J - 1] = '0') do J := J - 2;
        Z := 25 - J div 2
    end;
begin
    readln(N);
    with CURANSWER do begin
        quantity := 1; readnumber(max, zeros);
        for I := 2 to N do begin
            readnumber(CURNUM, Z);
            CHECK := CompareChar(max, CURNUM, 50);
            if (zeros < Z) or (zeros = Z) and (CHECK < 0)
            then begin quantity := 1; zeros := Z; max := CURNUM end
            else if (zeros = Z) and (CHECK = 0)
            then begin quantity := quantity + 1 end;
        end;
        writeln(quantity);
    end;
end.

```