

Незнайка и простые бразильские числа. 11 классы

Незнайка бродил по интернету и наткнулся на архив задач Международной иберо-американской олимпиады. Его внимание привлекла задача о простых бразильских числах. Незнайка выяснил, что простое бразильское число – это простое число, которое в некоторой позиционной системе по основанию b , где $b > 1$, записывается тремя или более чем тремя единицами, при этом никаких других цифр кроме единиц в записи числа нет. Например, $7 = 111_2$, $13 = 111_3$, $31 = 11111_2 = 111_5$. Но оказалось, что такую запись могут иметь некоторые составные числа. Например, $21 = 111_4$, $111 = 111_{10}$. Такие числа к простым бразильским числам не относятся.

Незнайка очень увлёкся простыми бразильскими числами. Буквально про каждое встреченное им число он хотел знать, является ли оно простым бразильским числом. Помогите Незнайке и составьте для него программу.

Программа считывает десятичное натуральное ненулевое число N , а затем последовательность из N десятичных натуральных ненулевых чисел $A[i]$. Программа находит все простые бразильские числа в последовательности. Среди них она находит простое бразильское число, которое встречается в последовательности наибольшее количество раз. Если таких чисел оказывается несколько, то программа выбирает из них наибольшее. Найденное число программа выводит. Если простых бразильских чисел в последовательности нет, то программа выводит 0.

Формат ввода: В первой строке содержится десятичное натуральное число N : $0 < N < 2001$. Во второй строке содержится последовательность из N десятичных натуральных чисел $A[i]$: $0 < A[i] < 160001$, $i = 1, \dots, N$.

Формат вывода: Выводится десятичная запись без незначащих нулей простого бразильского числа, встречающегося в последовательности наибольшее количество раз. Если есть несколько искоемых чисел, то выводится наибольшее из них. При отсутствии простых бразильских чисел в последовательности выводится 0.

Ввод примера №1:		Ввод примера №2:		Ввод примера №3:
4		6		1
31 7 31 7		31 13 31 13 7 13		21
Вывод примера №1:		Вывод примера №2:		Вывод примера №3:
31		13		0

Решение

В диапазон, заданный в условии задачи, попадает лишь 99 простых бразильских чисел: 7, 13, 31, ..., 158803. Подробнее см. на странице про последовательность A085104 в OEIS (On-Line Encyclopedia of Integer Sequences): <https://oeis.org/A085104>. Для получения эффективной программы следует самостоятельно рассчитать эти числа. Массив из них будет использоваться для быстрой проверки того, является ли число простым бразильским.

В решении используем массив BP из предварительно рассчитанных простых бразильских чисел. Также используем массив C из 99 числовых значений. Каждый элемент массива C будет хранить количество встреченных вхождений соответствующего простого бразильского числа в последовательность. Массив C инициализируем нулевыми значениями. В цикле будем считывать очередное число $A[i]$. Бинарным поиском по массиву BP будем проверять, является ли число простым бразильским. При успешной проверке получим индекс числа в массиве BP . У элемента массива C с тем же индексом увеличим значение на единицу. После обработки последовательности запустим второй цикл, в котором найдём наибольший argmax массива C . Если максимум массива C равен нулю, то это означает, что простых бразильских чисел в последовательности не было, и следует вывести 0. В другом случае используем найденный argmax как индекс в массиве BP и выводим искомое число.

Код возможного решения

```
program BRAZILPRIMES11(input, output);
const NUMBP =99;
    BP: array [1 .. NUMBP] of dword = (7, 13, 31, 43, 73, 127, 157, 211,
    241, 307, 421, 463, 601, 757, 1093, 1123, 1483, 1723, 2551, 2801,
    2971, 3307, 3541, 3907, 4423, 4831, 5113, 5701, 6007, 6163, 6481,
    8011, 8191, 9901, 10303, 11131, 12211, 12433, 13807, 14281, 17293,
```

```

19183, 19531, 20023, 20593, 21757, 22621, 22651, 23563, 24181, 26083,
26407, 27061, 28057, 28393, 30103, 30941, 31153, 35533, 35911, 37057,
37831, 41413, 42643, 43891, 46441, 47743, 53593, 55933, 55987, 60271,
60763, 71023, 74257, 77563, 78121, 82657, 83233, 84391, 86143, 88741,
95791, 98911, 108571, 110557, 113233, 117307, 118681, 121453, 123553,
127807, 131071, 136531, 143263, 145543, 147073, 154057, 156421, 158803);
var A, RESULT : dword;
    N, J : word;
    I, ARGM : byte;
    C: array [1 .. NUMBP] of word;
function binSearch(A : dword) : byte;
var L, M, H : byte;
begin
    L := 1;
    H := NUMBP;
    while L <= H do
    begin
        M := (L + H) div 2;
        if BP[M] > A then
        begin
            H := M - 1;
        end
        else if BP[M] < A then
        begin
            L := M + 1;
        end
        else
        begin
            break;
        end;
    end;
    if (M < NUMBP) AND (BP[M + 1] <= A) then binSearch := M + 1
    else if (M > 1) AND (BP[M] > A) then binSearch := M - 1
    else binSearch := M
end;

begin
    for I := 1 to NUMBP do C[I] := 0;
    readln(N);
    for J := 1 to N do begin
        read(A);
        I := binsearch(A);
        if A = BP[I] then C[I] := C[I] + 1
    end;
    ARGM := NUMBP;
    for I := NUMBP-1 downto 1 do begin
        if C[I] > C[ARGM] then ARGM := I;
    end;
    if C[ARGM] = 0 then RESULT := 0
    else RESULT := BP[ARGM];
    write(RESULT)
end.

```