

ФБЦ. 7-9 класс

В задаче рассмотрим факториальную буквенно-цифровую систему счисления (далее будем писать сокращённо – ФБЦ-систему). В ней используются цифры $d_i, i = 0 \dots 35$: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z. Строчные буквы не являются цифрами, используются только заглавные (прописные). Каждой цифре d_i приписано её числовое значение $val(d_i) = i$, так $val(0) = 0, \dots, val(9) = 9, val(A) = 10, \dots, val(Z) = 35$. Запись в ФБЦ-системе $d_{i_n}d_{i_{n-1}}d_{i_{n-2}} \dots d_{i_2}d_{i_1}$, где $0 \leq val(d_{i_k}) \leq k$ и $1 \leq k \leq n \leq 35$, означает беззнаковое число, равное $val(d_{i_n}) * n! + val(d_{i_{n-1}}) * (n-1)! + \dots + val(d_{i_2}) * 2! + val(d_{i_1}) * 1!$. Запись числа в ФБЦ-системе содержит не более чем 35 цифр и не содержит знака числа, то есть не все числа могут быть в ней записаны. На k -ой позиции в ФБЦ-записи (нумерация позиций ведётся справа налево, начиная с 1) допускается указывать цифру, числовое значение которой $val(d_{i_k})$ не превышает k и не меньше нуля. На самой правой позиции может быть либо 0, либо 1. На второй справа позиции может быть либо 0, либо 1, либо 2. И так далее. Допускается наличие незначащих нулей в левых позициях ФБЦ-записи. Незначащим является любой нуль, стоящий до первой стоящей слева ненулевой цифры, или, если записано нулевое число, то все нули, кроме самого правого. Например, десятичное число $100_{10} = 96 + 0 + 4 + 0 = 4 * 4! + 0 * 3! + 2 * 2! + 0 * 1! = 4020_{\text{ФБЦ}}$. Здесь приписанное снизу **ФБЦ** помечает запись числа в ФБЦ-системе. То же самое число может быть записано с незначащими нулями. Например, $0004020_{\text{ФБЦ}}$. Здесь три незначащих нуля.

Составьте программу, которая принимает на вход в первой строке десятичное число K – положительное натуральное число ($1 \leq K \leq 35$); во второй строке десятичное число N – положительное натуральное число ($1 \leq N \leq 50000$) – длину последовательности ФБЦ-чисел, и в последующих N строках – записи чисел X_i в ФБЦ-системе счисления, где $1 \leq i \leq N$. Программа находит элемент последовательности ФБЦ-чисел, который кратен $K!$ и при этом отстоит как можно дальше от начала последовательности. Программа выводит в первой строке запись найденного элемента последовательности в ФБЦ-системе без незначащих нулей, а во второй – номер этого элемента, записанный десятичным натуральным числом без знака. Нумерация элементов последовательности ведётся по порядку их следования от начала к концу, начальный элемент имеет номер 1. Если искомый элемент отсутствует, то программа выводит единственную строку с числом -1 .

Формат ввода: В первой строке содержится десятичное число K ($1 \leq K \leq 35$). Во второй строке содержится десятичное число N – длина последовательности ($1 \leq N \leq 50000$). В следующих N строках содержатся записи чисел X_i в ФБЦ-системе счисления, где $1 \leq i \leq N$. В записи числа X_i используются только десятичные цифры (0, ..., 9) и заглавные латинские буквы (A, ..., Z).

Формат вывода: Если среди введенных чисел X_i отсутствуют те, которые кратны $K!$, то программа в первой строке выводит -1 и ничего более не выводит. Иначе в первой строке выводится запись числа X_i в ФБЦ-системе без незначащих нулей, и это X_i таково, что оно кратно $K!$ и находится дальше от начала последовательности, чем другие элементы последовательности, кратные $K!$. Во второй строке выводится беззнаковое десятичное натуральное число, равное i – номеру элемента последовательности X_i , выведенного в первой строке. Нумерация элементов последовательности ведётся по порядку их следования от начала к концу, начальный элемент имеет номер 1.

Пример №1:

ввод:

2

1

4000

вывод:

4000

1

Пример №2:

ввод:

35

3

00000000000000000000000000000000

0000000000000000

0000000

вывод:

0

3

Пример №3:

ввод:

4

4

NKKJJIIHHGGFFEEDDCCBBAA554433222000

LKKJJIIHHGGFFEEDDCCBBAA554433220000

NKKJJIIHHGGFFEEDDCCBBAA554433222000

MKKJJIIHHGGFFEEDDCCBBAA554433221100

вывод:

NKKJJIIHHGGFFEEDDCCBBAA554433222000

3

Решение

В решении можно запрограммировать следующие подзадачи: 1) считывание очередного числа и представление его в виде строки из 35 символов с незначащими нулями, дополняющими считанную запись слева до 35 цифр; 2) определение кратности $K!$ считанного числа на основе подсчёта количества нулей, которыми заканчивается справа запись числа, и того факта, что ноль кратен любому ненулевому числу; 3) определение последнего числа, кратного $K!$, среди считанных чисел. Для решения третьей подзадачи достаточно одного прохода по последовательности, в котором совмещены построчный ввод чисел и их обработка. Следует хранить текущего кандидата на ответ (последнее число, кратное $K!$, среди всех чисел, которые программа успела считать) и его номер. Очередное число после считывания проверяется на кратность. При удачной проверке кандидат на ответ обновляется. Иначе делается переход к обработке следующего числа. По окончании обработки выводится искомое число и его номер (или -1 при неудачном поиске).

Код возможного решения

```
program FBC0709 (input, output);
const   MAXN = 35;
type    fbcnumber = array [1..MAXN] of char;
        answer = record no : word; number : fbcnumber end;
var      CURNUM : fbcnumber;
        K, N, I : word;
        CURANSWER : answer;
procedure readnumber(var FBCNUM : fbcnumber);
var      S : string;
        I, J : byte;
begin    readln(S);
        J := MAXN;
        for I := Length(S) downto 1 do begin
            FBCNUM[J] := S[I];
            J := J - 1;
        end;
        for I := J downto 1 do FBCNUM[I] := '0';
end;
function isdividable(var FBCNUM : fbcnumber; K : word) : boolean;
var      I : word;
begin    I := MAXN;
        while (I > 0) and (FBCNUM[I] = '0') do I := I - 1;
        isdividable := (I = 0) or (I <= (MAXN + 1 - K))
end;
begin    readln(K);
        readln(N);
        with CURANSWER do begin
```

```

no := 65535;
for I := 1 to MAXN do number[I] := '0';
for I := 1 to N do begin
    readnumber(CURNUM);
    if isdividable(CURNUM, K) then begin no := I; number := CURNUM end;
end;
if (no = 65535) then writeln(-1)
else begin
    I := 1;
    while (number[I] = '0') and (I < MAXN) do I := I + 1;
    while I <= MAXN do begin write(number[I]); I := I + 1 end;
    writeln;
    writeln(no);
end;
end;
end.

```