

ФБЦ. 10-11 класс

В задаче рассматриваются факториальную буквенно-цифровую систему счисления (далее будем писать сокращённо – ФБЦ-систему). В ней используются цифры  $d_i, i = 0 \dots 61$ : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,  $a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z$ . Строчные и прописные (заглавные) буквы-цифры различаются. Каждой цифре  $d_i$  приписано её числовое значение  $val(d_i) = i$ , так  $val(0) = 0, \dots, val(9) = 9, val(a) = 10, \dots, val(z) = 35, val(A) = 36, \dots, val(Z) = 61$ . 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  $1 \leq k \leq n \leq 61$ , 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 61 цифру 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. На третьей справа позиции может быть либо 0, либо 1, либо 2, либо 3. И так далее. Допускается наличие незначащих нулей в левых позициях ФБЦ-записи. Незначащим является любой ноль, 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,  $004020_{\text{ФБЦ}}$ . Здесь два незначащих нуля.

Составьте программу, которая принимает на вход в первой строке десятичное число  $K$  – положительное натуральное число ( $1 \leq K \leq 61$ ); 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 среди ФБЦ-чисел максимальное число, которое кратно  $K!$  и номера всех элементов последовательности, равных этому числу. Программа выводит в первой строке запись найденного числа в ФБЦ-системе без незначащих нулей. Затем программа выводит все найденные номера элементов в порядке возрастания, построчно, т. е. по одному номеру в одной строке. Формат вывода номеров – десятичное натуральное число без знака. Нумерация элементов последовательности ведётся по порядку их следования от начала к концу, начальный элемент имеет номер 1. Если искомое число отсутствует, то программа выводит единственную строку с числом  $-1$ .

*Формат ввода:* В первой строке содержится десятичное число  $K$  ( $1 \leq K \leq 61$ ); 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 ( $0, \dots, 9$ ), строчные латинские буквы ( $a, \dots, z$ ) и заглавные латинские буквы ( $A, \dots, Z$ ).

*Формат вывода:* Если среди введённых 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  $X_i$  должно быть максимальным среди элементов последовательности, кратных  $K!$ . В остальных строках выводятся по возрастанию в формате беззнакового десятичного натурального числа номера  $j$  всех тех  $X_j$ , которые равны числу, выведенному в первой строке. Каждый номер выводится в отдельной строке.

*Пример №1:*

*ввод:*

2

1

4000

ব্যবোধ:

4000

1

*Пример №2:*

ввод:

61

3

000000000000000000000000000000000000

[illegible]

00000000

вывод:

0

1

2

3

Пример №3:

ввод:

2

3

90CCbBaArrppoonnmllkkjjiihhggf feeddccbbaa554433221000

90CCbBaArrppoonnmllkkjjiihhggf feeddccbbaa554433221100

90CcbBaArrppoonnmllkkjjiihhggf feeddccbbaa554433221110

вывод:

90CCbBaArrppoonnmllkkjjiihhggf feeddccbbaa554433221100

2

### Решение

В решении можно запрограммировать следующие подзадачи: 1) считывание очередного числа и представление его в виде строки из 61 символа с незначащими нулями, дополняющими считанную запись слева до 61 цифры; 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) поиск всех условных *argmax*-ов последовательности, основанный на посимвольном сравнении строк, эквивалентном поразрядному сравнению считанных чисел. При сравнении строк следует помнить, что в *strcmp* или в *CompareChar* строчные буквы больше заглавных, что не согласуется с условием. Если инвертировать регистр букв при вводе, то *strcmp* и *CompareChar* будут работать. Для решения третьей подзадачи достаточно одного прохода по последовательности, в котором совмещены построчный ввод чисел и их обработка. Следует хранить текущий рекорд (максимальное среди всех чисел, кратных  $K!$ , которые программа успела считать) и номера всех чисел, равных текущему рекорду. Очередное число после считывания и успешной проверки на кратность  $K!$  сравнивается с рекордом. 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. По окончании обработки выводится найденное число и собранная коллекция номеров. Если кратных чисел в последовательности не было, то выводится -1.

### Код возможного решения

```
program FBC1011 (input, output);
const   MAXN = 61;
        BLANK : char = chr(ord('0')-1);
type    fbcnumber = array [1..MAXN] of char;
        numbers = record quantity : word; collection : array [1..50000] of word end;
var      CURMAX, CURNUM : fbcnumber;
        K, N, I : word;
        CHECK : integer;
        CURANSWER : numbers;
function changecase(c: char) : char;
begin   if (c >= 'a') and (c <= 'z') then
            changecase := chr(ord(c) - ord('a') + ord('A'))
        else if (c >= 'A') and (c <= 'Z') then
            changecase := chr(ord(c) - ord('A') + ord('a'))
        else changecase := c
end;
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] := changecase(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);
    for I := 1 to MAXN do CURMAX[I] := BLANK;
    with CURANSWER do begin
        quantity := 0;
        for I := 1 to N do begin
            readnumber(CURNUM);
            if isdividable(CURNUM, K) then begin
                CHECK := CompareChar(CURMAX, CURNUM, MAXN);
                if (CHECK < 0) or (quantity = 0) then
                    begin quantity := 1; collection[1] := I; CURMAX := CURNUM end
                else if (CHECK = 0) then
                    begin quantity := quantity + 1; collection[quantity] := I end;
            end;
        end;
        if (quantity = 0) then writeln(-1)
        else begin
            I := 1;
            while (CURMAX[I] = '0') and (I < MAXN) do I := I + 1;
            while I <= MAXN do begin write(changecase(CURMAX[I])); I := I + 1
            end;
            writeln;
            for I := 1 to quantity do writeln(collection[I]);
        end;
    end;
end;
end.

```