

МакФБЦ. 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 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, количество тех чисел последовательности, которые равны наибольшему из введённых X_i . Программа выводит в первой строке запись числа $\max_{i=1 \dots N} X_i$ в ФБЦ-системе без незначащих нулей, а во второй – искомое количество его вхождений в последовательность, записанное десятичным натуральным числом без знака.

Формат ввода: В первой 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: В первой строке выводится запись числа $\max_{i=1 \dots N} X_i$ в ФБЦ-системе без незначащих нулей. Во второй строке выводится беззнаковое десятичное натуральное число, равное количеству всех тех X_k , для которых верно, что $X_k = \max_{i=1 \dots N} X_i$.

Пример №1:

ввод:

1

4000

вывод:

4000

1

Пример №2:

ввод:

3

00000000000000000000000000000000

0000000000000000

0000000

вывод:

0

3

Пример №3:

ввод:

2

LKKJJIHHGGFFEEDDCCBBA554433221100

MKKJJIHHGGFFEEDDCCBBA554433221100

вывод:

MKKJJIHHGGFFEEDDCCBBA554433221100

1

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) подсчёт количества всех элементов последовательности, равных максимальному, который основан 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 среди всех чисел, которые программа успела считать) и количество всех элементов последовательности равных текущему рекорду. Очередное число после считывания сравнивается с рекордом. При равенстве количество *max*-ов увеличивается на 1. Если очередное число меньше, то делается переход к обработке следующего числа. Если очередное число больше, то оно становится рекордом, количество *max*-ов приравнивается 1. По окончании обработки выводится найденный максимум и количество *max*-ов.

Код возможного решения

```
program FBC0709 (input, output);
type   fbcnumber = array [1..35] of char;
       answer = record quantity : word; max : fbcnumber end;
var     CURNUM : fbcnumber;
       N, I : word;
       CHECK : integer;
       CURANSWER : answer;
procedure readnumber(var FBCNUM : fbcnumber);
var     S : string;
       I, J : byte;
begin
  readln(S);
  J:=35;
  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;
begin
  readln(N);
  with CURANSWER do begin
    quantity := 1;
    readnumber(max);
    for I := 2 to N do begin
      readnumber(CURNUM);
      CHECK := CompareChar(max, CURNUM, 35);
      if (CHECK < 0) then begin quantity := 1; max := CURNUM end
      else if (CHECK = 0) then begin quantity := quantity + 1 end;
    end;
    I:=1;
    while (max[I] = '0') and (I < 35) do I := I + 1;
    while I < 36 do begin write(max[I]); I := I + 1 end;
    writeln;
    writeln(quantity);
  end;
end.
```



Judge [Олимпиада "Ломоносов" по информатике. 2021-22. 7-9 класс. Отборочный тур.]: Submit a solution for 2

Problem 2: operations: Получи единицу

[\[Prev Problem\]](#) Problem: [\[Next Problem\]](#)

Prob Id: 2

Full score: 100

Input file name: input.txt or standard input

Time limit: 1 s

Real time limit: 5 s

Memory limit: 512M

Получи единицу

Вам дана десятичная запись целого неотрицательного числа без ведущих нулей. С ней можно делать следующие операции:

- '1' - умножить число на 2 (например, 12 преобразуется в 24);
- '2' - разделить число на 2, если оно четное (например, 12 преобразуется в 6);
- '3' и '4' - поделить запись числа на две половины равной длины и оставить только одну из половин ('3' - первую половину, '4' - вторую половину), если в записи числа четное число знаков (например, 12 можно преобразовать в 1 или 2);
- '5' и '6' - поделить запись числа на две половины равной длины и заменить одну из половин другой ('5' - используется первая половина, '6' - используется вторая половина), если в записи числа четное число знаков (например, 12 можно преобразовать в 11 или 22);

Если в результате операции получаются ведущие нули, их надо отбросить.

Требуется за по возможности минимальное количество операций из заданного числа получить 1.

Input format

В единственной строке ввода размещается десятичная запись целого неотрицательного числа без ведущих нулей. Длина числа не превышает 100 знаков.

Output format

В первой строке вывода размещается десятичная запись количества операций по получению 1. В следующей строке необходимо вывести строку из этих операций. Каждая операция выводится в виде одного символа ('1'..'6') без пробелов. Длина строки не должна превышать 10000 символов.

Если число 1 получить невозможно, то выведите единственное число -1.

Examples

Input

100

Output

4
2313

Notes

Для каждого теста результат будет оцениваться исходя из числа операций.

Submit a solution

Language: 

File Файл не выбран

[Main page](#)

[ejudge 3.9.1 \(2022-03-23 13:37:10\)](#).

Copyright © 2000-2022 Alexander Chernov.



Judge [Олимпиада "Ломоносов" по информатике. 2021-22. 7-9 класс. Отборочный тур.]: Submit a solution for 3

Problem 3: pseudofil: Числовые филворды

[\[Prev Problem\]](#) Problem: [\[Next Problem\]](#)

Prob Id: 3

Full score: 100

Input file name: input.txt or standard input

Time limit: 1 s

Real time limit: 5 s

Memory limit: 512M

Числовые филворды

Перед вами квадратная матрица ячеек. В каждой ячейке находится одно целое число. Одинаковых чисел в матрице нет. Вам надо разделить матрицу на непересекающиеся цепочки.

Цепочкой называется такая последовательность различных ячеек, в которой две соседние ячейки имеют общую сторону и числа в этой последовательности «почти» возрастают. Это означает, что можно исключить не более 1 числа из последовательности, чтобы получилась возрастающая последовательность.

Цепочки пересекаются, если имеют общую ячейку.

Не должно быть двух цепочек, ячейки которых можно объединить и получить новую цепочку.

Input format

Сначала идет размерность матрицы — целое число от 1 до 100. Затем идут элементы матрицы по строкам. Каждое число по модулю не превосходит 10000.

Output format

Распечатайте одно из разбиений матрицы на цепочки. Сначала распечатайте количество цепочек. Затем печатайте каждую цепочку на отдельной строке. Цепочку требуется выводить в формате - число элементов в цепочки, затем сами элементы. Числа в строке печатайте в десятичном виде через пробелы.

Examples

Input

```
3
2 7 1
```

```
4 5 8
6 3 99
```

Output

```
2
6 1 7 2 4 5 8
3 99 3 6
```

Notes

Для каждого теста результат будет оцениваться исходя из средней длины цепочек.

Submit a solution

Language:

File Файл не выбран

[Main page](#)

[ejudge 3.9.1 \(2022-03-23 13:37:10\)](#).

Copyright © 2000-2022 Alexander Chernov.



Judge [Олимпиада "Ломоносов" по информатике. 2021-22. 7-9 класс. Отборочный тур.]: Submit a solution for 4

Problem 4: pseudofil-2: Числовые филворды - 2

[\[Prev Problem\]](#) Problem: [\[Next Problem\]](#)

Prob Id: 4

Full score: 100

Input file name: input.txt or standard input

Time limit: 1 s

Real time limit: 5 s

Memory limit: 512M

Числовые филворды - 2

В этой задаче вам предстоит проверить на корректность набор цепочек из предыдущей задачи

Input format

Сначала идет размерность матрицы - целое число от 1 до 100. Затем идут элементы матрицы по строкам. Каждое число по модулю не превосходит 10000. Затем идет предлагаемое разбиение в формате описанном в предыдущей задаче

Output format

Выведите в первой строке "YES" или "NO" в зависимости от того является ли разбиение корректным.

Examples

Input

```
3
2 7 1
4 5 8
6 3 99
2
6 1 7 2 4 5 8
3 99 3 6
```

Output

```
YES
```

Input

3
2 7 1
4 5 8
6 3 99
1
9 1 7 2 4 5 3 6 99 8

Output

NO

Notes

Задача будет проверяться по группам тестов, в каждой из которых будут разбиения на цепочки разной корректности

Submit a solution

Language:

File Файл не выбран

[Main page](#)

[ejudge 3.9.1 \(2022-03-23 13:37:10\)](#).

Copyright © 2000-2022 Alexander Chernov.



Judge [Олимпиада "Ломоносов" по информатике. 2021-22. 7-9 класс. Отборочный тур.]: Submit a solution for 5

Problem 5: blackbox: Черный ящик

[\[Prev Problem\]](#) Problem: [\[Next Problem\]](#)

Prob Id: 5

Full score: 100

Input file name: input.txt or standard input

Time limit: 1 s

Real time limit: 5 s

Memory limit: 512M

Problem 5: Черный ящик

Вам на исследование дается некоторая [программа](#) без исходного кода. Программа принимает на вход единственное 16-битное беззнаковое число. Программа выводит результат на стандартный поток вывода.

Для некорректных входных данных программа выводит строку `invalid`.

Напишите программу, которая воспроизводит функциональность обратную к данной программы, то есть для выходных данных исходной программы будет выдавать входные.

Вы можете использовать [jslinux](#), на который загрузить программу для эмулятора.

После перехода по ссылке в окне браузера загрузится операционная система Linux и вы получите в окне браузера такой скрин:



Кнопка загрузки файла в операционную систему находится в левом нижнем углу. Загрузите туда этот [файл](#). После загрузки наберите в командной строке команду `"ls -l"`. Вы должны получить следующее:



Выполните команду `"chmod 755 prog"`. Затем можно запускать программу на выполнение с помощью команды `"./prog"`.



Examples

Input

Output

3

Notes

Во время тура сдаваемая программа будет тестироваться только на тестах из условия. В случае успешного прохождения тестов из условия посылка получит статус AC (Accepted for testing - принято на проверку).

После тура для окончательного тестирования будет выбрана последняя попытка, которая получила статус AC.

Submit a solution

Language: ▼

File Файл не выбран

[Main page](#)

[ejudge 3.9.1 \(2022-03-23 13:37:10\)](#).

Copyright © 2000-2022 Alexander Chernov.



Judge [Олимпиада "Ломоносов" по информатике. 2021-22. 7-9 класс. Отборочный тур.]: Submit a solution for 6

Problem 6: voting-2: Опрос

[\[Prev Problem\]](#) Problem:

Prob Id: 6

Full score: 100

Input file name: input.txt or standard input

Time limit: 1 s

Real time limit: 5 s

Memory limit: 512M

Problem 6: Опрос

Одна торговая сеть проводит опрос потребителей о предпочтительной марке одного из товаров. Марки товара пронумерованы от 1 до K . Каждый участник опроса выбирает одну из K марок. Результаты опроса записываются в блокчейн.

Каждый блок блокчейна имеет следующую структуру. Первое число — это количество записанных голосов в данном блок блокчейна. Назовем его `count`. Второе число (назовем его `sum`) — это контрольная сумма всех голосов в **предыдущем** блоке блокчейна по модулю 4294967291 (то есть остаток от деления суммы голосов на это число). Далее в блоке следует `count` чисел от 1 до K — зарегистрированные голоса.

В первом блоке контрольная сумма предыдущего блока равна 0.

Последний блок блокчейна особенный, в нем количество записей равно 0, и хранится только контрольная сумма голосов предыдущего блока.

К сожалению, в программу сохранения голосов в блокчейне вкралась ошибка, которая может приводить к искажению информации в блоках блокчейна. А именно, в испорченном блоке все записи об отданных голосах уменьшаются на 1 кроме тех, которые и так равны 1. То есть, если запись об отданном голосе была равна 5, в испорченном блоке она станет равна 4, но запись о голосовании 1 останется 1. Значения `count` и `sum` в блоке не искажаются, то есть остаются правильными. Один блок может быть испорчен не более одного раза.

Например, пусть блокчейн состоит из двух блоков следующей структуры:

```
10 0 1 2 3 4 5 6 1 2 1 3    -- первый блок
0 28                        -- второй блок
```

Контрольная сумма всех голосов (1 2 3 4 5 6 1 2 1 3) первого блока равна 28, что совпадает с суммой голосов, записанной во втором блоке. Это означает, что блок не испорчен, что позволяет просто посчитать популярность каждой марки.

Пусть дан блокчейн следующей структуры:

```
10 0 1 1 2 3 4 5 1 1 1 2    -- первый блок
0 28                          -- второй блок
```

Сумма всех голосов первого блока равна 21, что не совпадает с суммой голосов, записанной во втором блоке. Значит блок испорчен. Можно однозначно восстановить, сколько голосов было подано за каждую марку.

На стандартном потоке ввода сначала задается число K , затем подаются данные в формате, описанном выше. Количество товаров K не меньше 1 и не больше 255. Общее количество участников опроса не превосходит 1000. Числа во вводе могут разделяться произвольным количеством пробелов, табуляций и переводов строк.

На стандартный поток вывода напечатайте K чисел: количество голосов, отданных марки товара.

Examples

Input

```
6 10 0
1 2 3 4 5 6 1 2 1 3
0 28
```

Output

```
3 2 2 1 1 1
```

Submit a solution

Language:

File Файл не выбран

[Main page](#)

[ejudge 3.9.1 \(2022-03-23 13:37:10\)](#).

Copyright © 2000-2022 Alexander Chernov.

Loading...

Welcome to JS/Linux (i586)

Use 'vflogin username' to connect to your account.

You can create a new account at <https://vfsync.org/signup> .

Use 'export_file filename' to export a file to your computer.

Imported files are written to the home directory.

localhost:~# █



Loading...

Welcome to JS/Linux (i586)

Use 'vflogin username' to connect to your account.

You can create a new account at <https://vfsync.org/signup> .

Use 'export file filename' to export a file to your computer.

Imported files are written to the home directory.

localhost:~# ls -l

total 32

-rw-r--r--	1	root	root	114	Jul	5	2020	bench.py
-rw-r--r--	1	root	root	76	Jul	3	2020	hello.c
-rw-r--r--	1	root	root	22	Jun	26	2020	hello.js
-rw-----	1	1000	root	15208	Mar	9	11:20	prog
-rw-r--r--	1	root	root	151	Jul	5	2020	readme.txt

localhost:~#



```
localhost:~# ls
bench.py    hello.c    hello.js   prog       readme.txt
localhost:~# ls -Al
total 3108
-rw-----  1 root    root      43 Nov   5 09:57 .ash_history
drwx-----  3 root    root      61 Jul   6 2020 .cache
drwx-----  5 root    root     124 Jul   6 2020 .mozilla
drwxr-xr-x  4 root    root     202 Jul   6 2020 .wine
-rw-r--r--  1 root    root     114 Jul   5 2020 bench.py
-rw-r--r--  1 root    root      76 Jul   3 2020 hello.c
-rw-r--r--  1 root    root      22 Jun  26 2020 hello.js
-rw-----  1 1000    root    3148332 Nov   5 09:57 prog
-rw-r--r--  1 root    root     151 Jul   5 2020 readme.txt
localhost:~# ./prog
sh: ./prog: Permission denied
localhost:~# chmod +x ./prog
localhost:~# ls
bench.py    hello.c    hello.js   prog       readme.txt
localhost:~# ./prog
3
2
localhost:~#
```



Решения задач участников оценивались следующим образом:

- 1 задача — до 100 баллов
- 2 задача — до 100 баллов
- 3 задача — до 100 баллов
- 4 задача — до 100 баллов
- 5 задача — до 100 баллов
- 6 задача — до 100 баллов

Итоговая сумма баллов, набранная участником, переводилась в столбальную систему

Настройки тестового сервера и наборы тестов для проверки решений участников опубликованы по адресу: <https://ejudge.cs.msu.ru/lom2022.tgz>