Not Decreasing

СЕЗОН 2021/2022 – ШЕСТИ РУНД

За да бъде толкова успешна задачата **`increasing`** от първите тазгодишни контролни състезания за младши и момичешки отбор, Сашка трябваше да генерира най-добрите възможни тестове за нея. За поредния тест, тя решила да използва любимата си редица $a_0, a_1, a_2, \ldots, a_{N-1}$ от N числа. За всеки елемент от редицата е спазено $1 \leq a_i \leq P$. За всяко $1 \leq x \leq P$, има поне един елемент от редицата a_i, равен на x. За да получи теста, тя трябва да изпълни кода, даден по-долу. Любимата редица на Сашка е дадена в масива `a[]`. Броят елементи в редицата N е даден в параметъра `N`, а броят различни стойности в масива P – в параметъра `P`. Тестът се съдържа във вектора b.

```
int cnt[100000 + 1]; // cnt = {0,0,0,…,0}
std::vector < int > getB(int a[], int N, int P)
{
    std::vector < int > b(N, 0);
    for (int i = 0 ; i <= N-1 ; i++)
    {
        cnt[a[i]]++;
        int minCnt = cnt[1];
        for (int j = 2 ; j <= P ; j++)
        {
            minCnt = std::min(minCnt, cnt[j]);
        }
        b[i] = minCnt;
    }
    return b;
}
```

По-точно казано, функцията връща вектор b, в който i-тата стойност е равна на минималния брой срещания на стойност $1 \leq x \leq P$ измежду първите $i + 1$ елемента на редицата. За $N = 7, P = 3$ и $a = \{3,2,1,3,2,2,1\}$, $b = \{0,0,1,1,1,1,2\}$. Сашка генерирала прекрасен тест, но с изненада открила, че някак си е успяла да изтрие файла, съдържащ любимата ѝ редица. Сега, тя има на разположение само вектора b. Сашка, възмутена от номера, който успяла да си изиграе, иска по всевъзможен начин да възстанови a. Заради това, тя решила да даде задачата на шестия рунд на CodeIT, надявайки се поне един от състезателите да намери любимата ѝ редица. За да помогне на участниците в шестия рунд, тя подсказва, че от всички редици, които отговарят на стойностите във вектора b, любимата ѝ редица е тази с възможно най-къса ***най-дълга ненамаляваща подредица***. *Най-дълга ненамаляващата подредица* на дадена редица се получава, като се изтрият няколко елемента (потенциално 0) от редицата и се запази подредбата на останалите, така че броят на оставащите елементи да е възможно най-голям и те да образуват ненамаляваща редица. Една редица е ненамаляваща, ако няма двойка елементи в нея, за които левият е с по-

Not Decreasing

СЕЗОН 2021/2022 – ШЕСТИ РУНД

голяма стойност от десния. Още едно условие от Сашка е за всяка стойност $1 \leq x \leq P$, редицата да съдържа поне един елемент равен на x.

След твърде много обяснения, остава на Вас честта да напишете програма `notdecreasing.cpp`, която да намира редица a, която да отговаря на стойностите в b и да има възможно най-къса ***най-дълга ненамаляваща подредица***. Сашка би приела всяка една редица a, отговаряща на условията.

Вход

На първия ред от `notdecreasing.in` са дадени числата N и P. На втория ред от файла са дадени N числа, съответно $b_0, b_1, b_2, \ldots, b_{N-1}$.

Изход

На първия ред от `notdecreasing.out` изведете N числа, съответно $a_0, a_1, a_2, \ldots, a_{N-1}$. Ако има няколко редици, които отговарят на гореописаните условия, изведете която и да е от тях. Ако няма нито една редица, която отговаря на условията, изведете -1.

Ограничения

$1 \leq P \leq N \leq 10^5$

$0 \leq b_i \leq 10^9$

Ограничение по време: 0.5 sec.
Ограничение по памет: 256 MB.

Примерни тестове

Вход (notdecreasing.in)	**Изход (notdecreasing.out)**
4 2 0 1 1 2	2 1 2 1
4 2 0 1 1 0	-1
7 3 0 0 1 1 1 1 2	3 2 1 3 2 2 1

Дадените изходи са примерни и може да има други редици, отговарящи на условията.