Mascherine sporche (`mascherine`)

Filippo can't wait to go back to school! In fact, he just bought a set of new washable masks that he can't wait to show off. The set includes K types of mask, each of a different color, numbered from 1 to K.

Figure 1: Filippo's washing machine

There are N days left before the end of the school and Filippo has already decided, for each day, which mask he wants to use. On the i-th day Filippo will wear the mask with color C_i to go to school and, as soon as he will come back home, he will put the mask in the washing machine so that he can reuse it. zUnfortunately, Filippo has forgotten that masks take a whole day to dry and therefore he will not be able to use the same mask for two consecutive days. Therefore Filippo must modify his plan so that every day he can use a clean, dry mask.

Help Filippo modify his plan by changing the **minimum number** of days.

Implementation

You should submit a single file, with either a `.c` or `.cpp` extension.

> ☞ Among the attachments of this task you will find the templates `mascherine.c` and `mascherine.cpp` with a sample implementation.

You will have to implement the following function:

C	`void riprogramma(int N, int K, int C[]);`
C++	`void riprogramma(int N, int K, vector<int>& C);`

The function will be called with the following parameters:

- The integer N represents the number of school days.
- The integer K represents the number of Filippo's masks.
- The vector C, indexed from 0 to $N-1$, contains the mask Filippo decided to use on the i-th day.
- At the end of the execution of the function, the vector C must contain a valid plan that minimizes the differences with the original one.

Sample Grader

Among this task's attachments you will find a simplified version of the grader used during the evaluation, which you can use to test your solutions locally. The sample grader reads data from `stdin`, calls the functions that you should implement and writes back on `stdout` using the following format.

The input file is formed by 2 lines, containing:

- Line 1: the integers N and K.
- Line 2: the N integers $C_0, \ldots, C_{N-1}$.

The output file is formed by a single line:

- Line 1: the integers $C_0, C_1, \ldots, C_{N-1}$ at the end of the execution.

Constraints

- $2 \leq K \leq N \leq 2 \cdot 10^5$.
- $1 \leq C_i \leq K$.
- If there are multiple solutions you can output any.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

- **Subtask 1 [0 points]**: Examples.
- **Subtask 2 [10 points]**: $N = 2$
- **Subtask 3 [15 points]**: Every mask is used at most 2 times.
- **Subtask 4 [25 points]**: $N \leq 1000$
- **Subtask 5 [20 points]**: $K \geq 3$
- **Subtask 6 [30 points]**: No additional limitations.

Examples

stdin	stdout
5 3 1 2 2 3 1	1 2 1 3 1
5 2 2 1 1 2 1	1 2 1 2 1

Explanations

In the **first example** Filippo can't use mask number 2 two days in a row. A possible solution is to use mask number 1 on the third day.

In the **second example** a possible solution is to use mask number 1 on the first day and mask number 2 on the second day.