The 2021 icpc

Asia Macau Regional Contest

Practice Session

April 3, 2022

Problem List

A	K-skip Permutation
B	Hotpot
C	Wandering Robot

This problem set should contain 3 (three) problems on 3 (three) numbered pages. Please inform a runner immediately if something is missing from your problem set.

Prepared by SUA Programming Contest Problem Setter Team.
https://sua.ac/

Hosted by

Problem Set Prepared by

It's against the rules to open non-contest websites during the contest.
If you're interested (which is our pleasure),
please scan the QR code only after the contest.

Problem A. K-skip Permutation

For a permutation $P = p_1, p_2, \cdots, p_n$ of n, let $f(P, k)$ be the number of i satisfying $1 \leq i < n$ and $p_i + k = p_{i+1}$.

Given two integers n and k, your task is to find a permutation P of n such that $f(P, k)$ is maximized.

Recall that in a permutation of n, each integer from 1 to n (both inclusive) appears exactly once.

Input

There is only one test case in each test file.

The first and only line contains two integers n and k ($1 \leq n, k \leq 10^6$).

Output

Output one line containing n integers indicating a permutation P of n that maximizes $f(P, k)$. If there are multiple valid answers you can output any of them.

Please, DO NOT output extra spaces at the end of the line, or your answer may be considered incorrect!

Examples

standard input	standard output
3 1	1 2 3
7 3	2 5 1 4 7 3 6
3 7	1 3 2

Problem B. Hotpot

Sichuan hotpot is one of the most famous dishes around the world. People love its spicy taste.

There are n tourists, numbered from 0 to $(n-1)$, sitting around a hotpot. There are k types of ingredients for the hotpot in total and the i-th tourist favors ingredient a_i most. Initially, every tourist has a happiness value of 0 and the pot is empty.

The tourists will perform m moves one after another, where the i-th (numbered from 0 to $(m-1)$) move is performed by tourist $(i \mod n)$. When tourist t moves:

- If ingredient a_t exists in the pot, he will eat them all and gain 1 happiness value.
- Otherwise, he will put one unit of ingredient a_t into the pot. His happiness value remains unchanged.

Your task is to calculate the happiness value for each tourist after m moves.

Input

There are multiple test cases. The first line of the input contains an integer T $(1 \le T \le 10^3)$ indicating the number of test cases. For each test case:

The first line contains three integers n, k and m $(1 \le n \le 10^5, 1 \le k \le 10^5, 1 \le m \le 10^9)$ indicating the number of tourists, the number of types of ingredients and the number of moves.

The second line contains n integers $a_0, a_1, \cdots, a_{n-1}$ $(1 \le a_i \le k)$ where a_i indicates the favorite ingredient of tourist i.

It's guaranteed that neither the sum of n nor the sum of k of all the test cases will exceed 2×10^5.

Output

For each test case output n integers $h_0, h_1, \cdots, h_{n-1}$ in one line separated by a space, where h_i indicates the happiness value of tourist i after m moves.

Please, DO NOT output extra spaces at the end of each line, or your answer might be considered incorrect!

Example

standard input	standard output
4 3 2 6 1 1 2 1 1 5 1 2 2 10 1 2 2 2 10 1 1	0 2 1 2 2 2 0 5

Note

The first sample test case is explained as follows:

Move	Tourist	Action	Pot after move
0	0	Puts ingredient 1 into the pot	{1}
1	1	Eats ingredient 1 in the pot	{}
2	2	Puts ingredient 2 into the pot	{2}
3	0	Puts ingredient 1 into the pot	{1, 2}
4	1	Eats ingredient 1 in the pot	{2}
5	2	Eats ingredient 2 in the pot	{}

Problem C. Wandering Robot

DreamGrid creates a programmable robot to explore an infinite two-dimension plane. The robot has a basic instruction sequence $a_1, a_2, \ldots a_n$ and a "repeating parameter" k, which together form the full instruction sequence $s_1, s_2, \ldots, s_n, s_{n+1}, \ldots, s_{nk}$ and control the robot.

There are 4 types of valid instructions in total, which are 'U' (up), 'D' (down), 'L' (left) and 'R' (right). Assuming that the robot is currently at (x, y), the instructions control the robot in the way below:

- U: Moves the robot to $(x, y + 1)$.
- D: Moves the robot to $(x, y - 1)$.
- L: Moves the robot to $(x - 1, y)$.
- R: Moves the robot to $(x + 1, y)$.

The full instruction sequence can be derived from the following equations

$$\begin{cases} s_i = a_i & \text{if } 1 \le i \le n \\ s_i = s_{i-n} & \text{otherwise} \end{cases}$$

The robot is initially at $(0, 0)$ and executes the instructions in the full instruction sequence one by one. To estimate the exploration procedure, DreamGrid would like to calculate the largest Manhattan distance between the robot and the start point $(0, 0)$ during the execution of the nk instructions.

Recall that the Manhattan distance between (x_1, y_1) and (x_2, y_2) is defined as $|x_1 - x_2| + |y_1 - y_2|$.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains two integers n and k ($1 \le n \le 10^5, 1 \le k \le 10^9$), indicating the length of the basic instruction sequence and the repeating parameter.

The second line contains a string $A = a_1 a_2 \ldots a_n$ ($|A| = n$, $a_i \in \{\text{'L'}, \text{'R'}, \text{'U'}, \text{'D'}\}$), where a_i indicates the i-th instruction in the basic instriction sequence.

It's guaranteed that the sum of $|A|$ of all test cases will not exceed 2×10^6.

Output

For each test case output one line containing one integer indicating the answer.

Example

standard input	standard output
2 3 3 RUL 1 1000000000 D	4 1000000000

Note

For the first sample test case, the final instruction sequence is "RULRULRUL" and the route of the robot is (0, 0) - (1, 0) - (1, 1) - (0, 1) - (1, 1) - (1, 2) - (0, 2) - (1, 2) - (1, 3) - (0, 3). It's obvious that the farthest point on the route is (1, 3) and the answer is 4.