Codeforces Round #611 (Div. 3)

A. Minutes Before the New Year

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

New Year is coming and you are excited to know how many minutes remain before the New Year. You know that currently the clock shows h hours and m minutes, where $0 \le hh < 24$ and $0 \le mm < 60$. *We use 24-hour time format!*

Your task is to find the number of minutes before the New Year. You know that New Year comes when the clock shows 0 hours and 0 minutes.

You have to answer t independent test cases.

Input

The first line of the input contains one integer t ($1 \le t \le 1439$) — the number of test cases.

The following t lines describe test cases. The i-th line contains the time as two integers h and m ($0 \le h < 24$, $0 \le m < 60$). It is guaranteed that this time is **not** a midnight, i.e. the following two conditions can't be met at the same time: $h = 0$ and $m = 0$. It is guaranteed that both h and m are given without leading zeros.

Output

For each test case, print the answer on it — the number of minutes before the New Year.

Example

input
5 23 55 23 0 0 1 4 20 23 59
output
5 60 1439 1180 1

B. Candies Division

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Santa has n candies and he wants to gift them to k kids. He wants to divide as many candies as possible between all k kids. Santa can't divide one candy into parts but he is allowed to not use some candies at all.

Suppose the kid who recieves the minimum number of candies has a candies and the kid who recieves the maximum number of candies has b candies. Then Santa will be **satisfied**, if the both conditions are met at the same time:

- $b - a \le 1$ (it means $b = a$ or $b = a + 1$);
- the number of kids who has $a + 1$ candies (**note that** $a + 1$ **not necessarily equals** b) does not exceed $\lfloor \frac{k}{2} \rfloor$ (less than or equal to $\lfloor \frac{k}{2} \rfloor$).

$\lfloor \frac{k}{2} \rfloor$ is k divided by 2 and rounded **down** to the nearest integer. For example, if $k = 5$ then $\lfloor \frac{k}{2} \rfloor = \lfloor \frac{5}{2} \rfloor = 2$.

Your task is to find the maximum number of candies Santa can give to kids so that he will be **satisfied**.

You have to answer t independent test cases.

Input

The first line of the input contains one integer t ($1 \le t \le 5 \cdot 10^4$) — the number of test cases.

The next t lines describe test cases. The i-th test case contains two integers n and k ($1 \le n, k \le 10^9$) — the number of candies and the number of kids.

Output
For each test case print the answer on it — the maximum number of candies Santa can give to kids so that he will be **satisfied**.

Example

input
5 5 2 19 4 12 7 6 2 100000 50010
output
5 18 10 6 75015

Note
In the first test case, Santa can give 3 and 2 candies to kids. There $a = 2, b = 3, a + 1 = 3$.

In the second test case, Santa can give $5, 5, 4$ and 4 candies. There $a = 4, b = 5, a + 1 = 5$. The answer cannot be greater because then the number of kids with 5 candies will be 3.

In the third test case, Santa can distribute candies in the following way: $[1, 2, 2, 1, 1, 2, 1]$. There $a = 1, b = 2, a + 1 = 2$. He cannot distribute two remaining candies in a way to be satisfied.

In the fourth test case, Santa can distribute candies in the following way: $[3, 3]$. There $a = 3, b = 3, a + 1 = 4$. Santa distributed all 6 candies.

C. Friends and Gifts

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

There are n friends who want to give gifts for the New Year to each other. Each friend should give **exactly** one gift and receive **exactly** one gift. The friend **cannot** give the gift to himself.

For each friend the value f_i is known: it is either $f_i = 0$ if the i-th friend doesn't know whom he wants to give the gift to or $1 \le f_i \le n$ if the i-th friend wants to give the gift to the friend f_i.

You want to fill in the unknown values ($f_i = 0$) in such a way that each friend gives **exactly** one gift and receives **exactly** one gift and there is **no** friend who gives the gift to himself. It is guaranteed that the initial information isn't contradictory.

If there are several answers, you can print any.

Input
The first line of the input contains one integer n ($2 \le n \le 2 \cdot 10^5$) — the number of friends.

The second line of the input contains n integers $f_1, f_2, \ldots, f_n$ ($0 \le f_i \le n$, $f_i \ne i$, all $f_i \ne 0$ are distinct), where f_i is the either $f_i = 0$ if the i-th friend doesn't know whom he wants to give the gift to or $1 \le f_i \le n$ if the i-th friend wants to give the gift to the friend f_i. It is also guaranteed that there is **at least two** values $f_i = 0$.

Output
Print n integers $nf_1, nf_2, \ldots, nf_n$, where nf_i should be equal to f_i if $f_i \ne 0$ or the number of friend whom the i-th friend wants to give the gift to. All values nf_i should be distinct, nf_i cannot be equal to i. Each friend gives **exactly** one gift and receives **exactly** one gift and there is **no** friend who gives the gift to himself.

If there are several answers, you can print any.

Examples

input
5 5 0 0 2 4
output
5 3 1 2 4

input
7 7 0 0 1 4 0 6
output
7 3 2 1 4 5 6

input
7 7 4 0 3 0 5 1
output
7 4 2 3 6 5 1

input
5 2 1 0 0 0
output
2 1 4 5 3

D. Christmas Trees

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

There are n Christmas trees on an infinite number line. The i-th tree grows at the position x_i. All x_i are guaranteed to be distinct.

Each **integer** point can be either occupied by the Christmas tree, by the human or not occupied at all. Non-integer points cannot be occupied by anything.

There are m people who want to celebrate Christmas. Let $y_1, y_2, \ldots, y_m$ be the positions of people (note that all values $x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m$ should be **distinct** and all y_j should be **integer**). You want to find such an arrangement of people that the value $\sum\limits_{j=1}^{m} \min\limits_{i=1}^{n} |x_i - y_j|$ is the minimum possible (in other words, the sum of distances to the nearest Christmas tree for all people should be minimized).

In other words, let d_j be the distance from the j-th human to the nearest Christmas tree ($d_j = \min\limits_{i=1}^{n} |y_j - x_i|$). Then you need to choose such positions $y_1, y_2, \ldots, y_m$ that $\sum\limits_{j=1}^{m} d_j$ is the minimum possible.

Input

The first line of the input contains two integers n and m ($1 \le n, m \le 2 \cdot 10^5$) — the number of Christmas trees and the number of people.

The second line of the input contains n integers $x_1, x_2, \ldots, x_n$ ($-10^9 \le x_i \le 10^9$), where x_i is the position of the i-th Christmas tree. It is guaranteed that all x_i are distinct.

Output

In the first line print one integer res — the minimum possible value of $\sum\limits_{j=1}^{m} \min\limits_{i=1}^{n} |x_i - y_j|$ (in other words, the sum of distances to the nearest Christmas tree for all people).

In the second line print m integers $y_1, y_2, \ldots, y_m$ ($-2 \cdot 10^9 \le y_j \le 2 \cdot 10^9$), where y_j is the position of the j-th human. All y_j should be distinct and all values $x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m$ should be **distinct**.

If there are multiple answers, print any of them.

Examples

input
2 6 1 5
output
8 -1 2 6 4 0 3

input
3 5 0 3 1
output
7 5 -2 4 -1 2

E. New Year Parties

time limit per test: 2 seconds

memory limit per test: 256 megabytes
input: standard input
output: standard output

Oh, New Year. The time to gather all your friends and reflect on the heartwarming events of the past year...

n friends live in a city which can be represented as a number line. The i-th friend lives in a house with an integer coordinate x_i. The i-th friend can come celebrate the New Year to the house with coordinate $x_i - 1$, $x_i + 1$ or stay at x_i. Each friend is allowed to move no more than once.

For all friends $1 \le x_i \le n$ holds, however, they can come to houses with coordinates 0 and $n + 1$ (if their houses are at 1 or n, respectively).

For example, let the initial positions be $x = [1, 2, 4, 4]$. The final ones then can be $[1, 3, 3, 4]$, $[0, 2, 3, 3]$, $[2, 2, 5, 5]$, $[2, 1, 3, 5]$ and so on. The number of occupied houses is the number of distinct positions among the final ones.

So all friends choose the moves they want to perform. After that the number of occupied houses is calculated. What is the minimum and the maximum number of occupied houses can there be?

Input
The first line contains a single integer n ($1 \le n \le 2 \cdot 10^5$) — the number of friends.

The second line contains n integers $x_1, x_2, \ldots, x_n$ ($1 \le x_i \le n$) — the coordinates of the houses of the friends.

Output
Print two integers — the minimum and the maximum possible number of occupied houses after all moves are performed.

Examples

input
4 1 2 4 4
output
2 4

input
9 1 1 8 8 8 4 4 4 4
output
3 8

input
7 4 3 7 1 4 3 3
output
3 6

Note
In the first example friends can go to $[2, 2, 3, 3]$. So friend 1 goes to $x_1 + 1$, friend 2 stays at his house x_2, friend 3 goes to $x_3 - 1$ and friend 4 goes to $x_4 - 1$. $[1, 1, 3, 3]$, $[2, 2, 3, 3]$ or $[2, 2, 4, 4]$ are also all valid options to obtain 2 occupied houses.

For the maximum number of occupied houses friends can go to $[1, 2, 3, 4]$ or to $[0, 2, 4, 5]$, for example.

F. DIY Garland

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Polycarp has decided to decorate his room because the New Year is soon. One of the main decorations that Polycarp will install is the garland he is going to solder himself.

Simple garlands consisting of several lamps connected by one wire are too boring for Polycarp. He is going to solder a garland consisting of n lamps and $n - 1$ wires. Exactly one lamp will be connected to power grid, and power will be transmitted from it to other lamps by the wires. Each wire connectes exactly two lamps; one lamp is called **the main lamp** for this wire (the one that gets power from some other wire and transmits it to this wire), the other one is called **the auxiliary lamp** (the one that gets power from this wire). Obviously, each lamp has at most one wire that brings power to it (and this lamp is the auxiliary lamp for this wire, and the main lamp for all other wires connected directly to it).

Each lamp has a brightness value associated with it, the i-th lamp has brightness 2^i. We define the **importance** of the wire as the sum of brightness values over all lamps that become disconnected from the grid if the wire is cut (and all other wires are still working).

Polycarp has drawn the scheme of the garland he wants to make (the scheme depicts all n lamp and $n-1$ wires, and the lamp that will be connected directly to the grid is marked; the wires are placed in such a way that the power can be transmitted to each lamp). After that, Polycarp calculated the importance of each wire, enumerated them from 1 to $n-1$ in descending order of their importance, and then wrote the index of the main lamp for each wire (in the order from the first wire to the last one).

The following day Polycarp bought all required components of the garland and decided to solder it — but he could not find the scheme. Fortunately, Polycarp found the list of indices of main lamps for all wires. Can you help him restore the original scheme?

Input

The first line contains one integer n ($2 \le n \le 2 \cdot 10^5$) — the number of lamps.

The second line contains $n-1$ integers a_1, a_2, ..., a_{n-1} ($1 \le a_i \le n$), where a_i is the index of the main lamp for the i-th wire (wires are numbered in descending order of importance).

Output

If it is impossible to restore the original scheme, print one integer -1.

Otherwise print the scheme as follows. In the first line, print one integer k ($1 \le k \le n$) — the index of the lamp that is connected to the power grid. Then print $n-1$ lines, each containing two integers x_i and y_i ($1 \le x_i, y_i \le n$, $x_i \ne y_i$) — the indices of the lamps connected by some wire. The descriptions of the wires (and the lamps connected by a wire) can be printed in any order. The printed description must correspond to a scheme of a garland such that Polycarp could have written the list a_1, a_2, ..., a_{n-1} from it. If there are multiple such schemes, output any of them.

Example

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

Note

The scheme for the first example (R denotes the lamp connected to the grid, the numbers on wires are their importance values):