CODEFORCES β
Sponsored by Telegram

Divide by Zero 2021 and Codeforces Round #714 (Div. 2)

A. Array and Peaks

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

A sequence of n integers is called a permutation if it contains all integers from 1 to n exactly once.

Given two integers n and k, construct a permutation a of numbers from 1 to n which has **exactly** k peaks. An index i of an array a of size n is said to be a peak if $1 < i < n$ and $a_i > a_{i-1}$ and $a_i > a_{i+1}$. If such permutation is not possible, then print -1.

Input

The first line contains an integer t ($1 \leq t \leq 100$) — the number of test cases.

Then t lines follow, each containing two space-separated integers n ($1 \leq n \leq 100$) and k ($0 \leq k \leq n$) — the length of an array and the required number of peaks.

Output

Output t lines. For each test case, if there is no permutation with given length and number of peaks, then print -1. Otherwise print a line containing n space-separated integers which forms a permutation of numbers from 1 to n and contains exactly k peaks.

If there are multiple answers, print any.

Example

input

```
5
1 0
5 2
6 6
2 1
6 1
```

output

```
1
2 4 1 5 3
-1
-1
1 3 6 5 4 2
```

Note

In the second test case of the example, we have array $a = [2, 4, 1, 5, 3]$. Here, indices $i = 2$ and $i = 4$ are the peaks of the array. This is because $(a_2 > a_1, a_2 > a_3)$ and $(a_4 > a_3, a_4 > a_5)$.

B. AND Sequences

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

A sequence of n non-negative integers ($n \geq 2$) $a_1, a_2, \dots, a_n$ is called good if for all i from 1 to $n-1$ the following condition holds true:

$$a_1 \,\&\, a_2 \,\&\, \dots \,\&\, a_i = a_{i+1} \,\&\, a_{i+2} \,\&\, \dots \,\&\, a_n,$$

where $\&$ denotes the bitwise AND operation.

You are given an array a of size n ($n \geq 2$). Find the number of permutations p of numbers ranging from 1 to n, for which the sequence a_{p_1}, a_{p_2}, ... ,a_{p_n} is good. Since this number can be large, output it modulo $10^9 + 7$.

Input

The first line contains a single integer t ($1 \leq t \leq 10^4$), denoting the number of test cases.

The first line of each test case contains a single integer n ($2 \leq n \leq 2 \cdot 10^5$) — the size of the array.

The second line of each test case contains n integers $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 10^9$) — the elements of the array.

It is guaranteed that the sum of n over all test cases doesn't exceed $2 \cdot 10^5$.

Output

Output t lines, where the i-th line contains the number of good permutations in the i-th test case modulo $10^9 + 7$.

Example

input

```
4
3
1 1 1
5
1 2 3 4 5
5
0 2 0 3 0
4
1 3 5 1
```

output

```
6
0
36
4
```

Note

In the first test case, since all the numbers are equal, whatever permutation we take, the sequence is good. There are a total of 6 permutations possible with numbers from 1 to 3: $[1, 2, 3]$, $[1, 3, 2]$, $[2, 1, 3]$, $[2, 3, 1]$, $[3, 1, 2]$, $[3, 2, 1]$.

In the second test case, it can be proved that no permutation exists for which the sequence is good.

In the third test case, there are a total of 36 permutations for which the sequence is good. One of them is the permutation $[1, 5, 4, 2, 3]$ which results in the sequence $s = [0, 0, 3, 2, 0]$. This is a good sequence because

- $s_1 = s_2 \,\&\, s_3 \,\&\, s_4 \,\&\, s_5 = 0$,
- $s_1 \,\&\, s_2 = s_3 \,\&\, s_4 \,\&\, s_5 = 0$,
- $s_1 \,\&\, s_2 \,\&\, s_3 = s_4 \,\&\, s_5 = 0$,
- $s_1 \,\&\, s_2 \,\&\, s_3 \,\&\, s_4 = s_5 = 0$.

C. Add One

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

You are given an integer n. You have to apply m operations to it.

In a single operation, you **must** replace every digit d of the number with the decimal representation of integer $d + 1$. For example, 1912 becomes 21023 after applying the operation once.

You have to find the length of n after applying m operations. Since the answer can be very large, print it modulo 10^9+7.

Input

The first line contains a single integer t ($1 \le t \le 2 \cdot 10^5$) — the number of test cases.

The only line of each test case contains two integers n ($1 \le n \le 10^9$) and m ($1 \le m \le 2 \cdot 10^5$) — the initial number and the number of operations.

Output

For each test case output the length of the resulting number modulo 10^9+7.

Example

input

```
5
1912 1
5 6
999 1
88 2
12 100
```

output

```
5
2
6
4
2115
```

Note

For the first test, 1912 becomes 21023 after 1 operation which is of length 5.

For the second test, 5 becomes 21 after 6 operations which is of length 2.

For the third test, 999 becomes 101010 after 1 operation which is of length 6.

For the fourth test, 88 becomes 1010 after 2 operations which is of length 4.

D. GCD and MST

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

You are given an array a of n ($n \ge 2$) positive integers and an integer p. Consider an undirected weighted graph of n vertices numbered from 1 to n for which the edges between the vertices i and j ($i<j$) are added in the following manner:

- If $gcd(a_i, a_{i+1}, a_{i+2}, \dots, a_j) = min(a_i, a_{i+1}, a_{i+2}, \dots, a_j)$, then there is an edge of weight $min(a_i, a_{i+1}, a_{i+2}, \dots, a_j)$ between i and j.
- If $i+1=j$, then there is an edge of weight p between i and j.

Here $gcd(x, y, \ldots)$ denotes the greatest common divisor (GCD) of integers x, y, ...

Note that there could be multiple edges between i and j if both of the above conditions are true, and if both the conditions fail for i and j, then there is no edge between these vertices.

The goal is to find the weight of the minimum spanning tree of this graph.

Input

The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases.

The first line of each test case contains two integers n ($2 \le n \le 2 \cdot 10^5$) and p ($1 \le p \le 10^9$) — the number of nodes and the parameter p.

The second line contains n integers $a_1, a_2, a_3, \dots, a_n$ ($1 \le a_i \le 10^9$).

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

Output t lines. For each test case print the weight of the corresponding graph.

Example

input

```
4
2 5
10 10
2 5
3 3
4 5
5 2 4 9
8 8
5 3 3 6 10 100 9 15
```

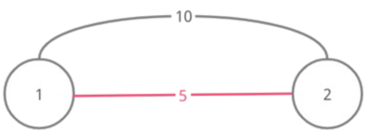
output

```
5
3
12
46
```

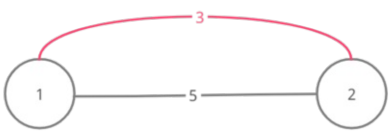
Note

Here are the graphs for the four test cases of the example (the edges of a possible MST of the graphs are marked pink):

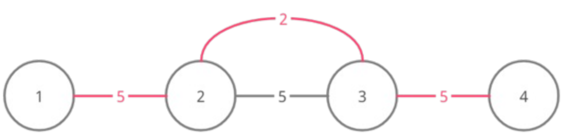
For test case 1



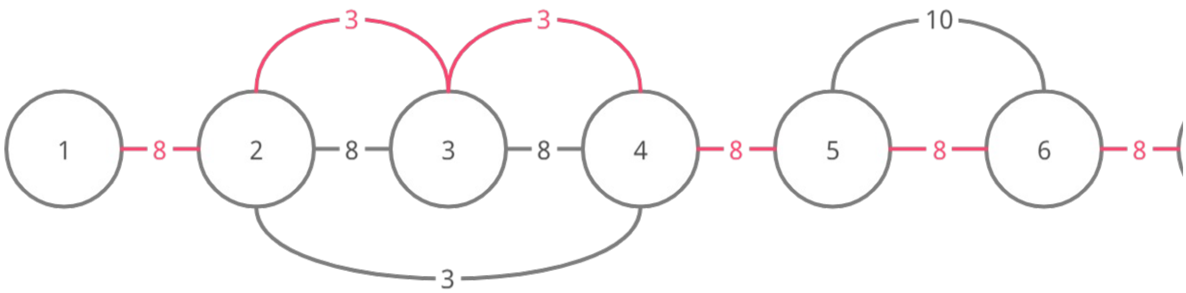
For test case 2



For test case 3



For test case 4



E. Cost Equilibrium

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

An array is called beautiful if all the elements in the array are equal.

You can transform an array using the following steps any number of times:

1. Choose two indices i and j ($1 \le i, j \le n$), and an integer x ($1 \le x \le a_i$). Let i be the source index and j be the sink index.
2. Decrease the i-th element by x, and increase the j-th element by x. The resulting values at i-th and j-th index are $a_i - x$ and $a_j + x$ respectively.
3. The cost of this operation is $x \cdot |j - i|$.
4. Now the i-th index can no longer be the sink and the j-th index can no longer be the source.

The total cost of a transformation is the sum of all the costs in step 3.
For example, array $[0, 2, 3, 3]$ can be transformed into a beautiful array $[2, 2, 2, 2]$ with total cost $1 \cdot |1-3| + 1 \cdot |1-4| = 5$.

An array is called balanced, if it can be transformed into a beautiful array, and the cost of such transformation is **uniquely** defined. In other words, the minimum cost of transformation into a beautiful array equals the maximum cost.

You are given an array $a_1, a_2, \ldots, a_n$ of length n, consisting of non-negative integers. Your task is to find the number of balanced arrays which are permutations of the given array. Two arrays are considered different, if elements at some position differ. Since the answer can be large, output it modulo $10^9 + 7$.

Input

The first line contains a single integer n ($1 \le n \le 10^5$) — the size of the array.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 10^9$).

Output

Output a single integer — the number of balanced permutations modulo $10^9 + 7$.

Examples

input

```
3
1 2 3
```

output

```
6
```

input

```
4
0 4 0 4
```

output

```
2
```

input

```
5
0 11 12 13 14
```

output

```
120
```

Note

In the first example, $[1, 2, 3]$ is a valid permutation as we can consider the index with value 3 as the source and index with value 1 as the sink. Thus, after conversion we get a beautiful array $[2, 2, 2]$, and the total cost would be 2. We can show that this is the only transformation of this array that leads to a beautiful array. Similarly, we can check for other permutations too.

In the second example, $[0, 0, 4, 4]$ and $[4, 4, 0, 0]$ are balanced permutations.

In the third example, all permutations are balanced.

F. Swapping Problem

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

You are given 2 arrays a and b, both of size n. You can swap two elements in b at most **once** (or leave it as it is), and you are required to minimize the value

$$\sum_i |a_i - b_i|.$$

Find the minimum possible value of this sum.

Input

The first line contains a single integer n ($1 \le n \le 2 \cdot 10^5$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$).

The third line contains n integers $b_1, b_2, \ldots, b_n$ ($1 \le b_i \le 10^9$).

Output

Output the minimum value of $\sum_i |a_i - b_i|$.

Examples

input

```
5
5 4 3 2 1
1 2 3 4 5
```

output

```
4
```

input
2 1 3 4 2

output
2

Note

In the first example, we can swap the first and fifth element in array b, so that it becomes $[5, 2, 3, 4, 1]$.

Therefore, the minimum possible value of this sum would be $|5-5|+|4-2|+|3-3|+|2-4|+|1-1|=4$.

In the second example, we can swap the first and second elements. So, our answer would be 2.