Codeforces Round #752 (Div. 2)

A. Era

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

Shohag has an integer sequence $a_1, a_2, \ldots, a_n$. He can perform the following operation any number of times (possibly, zero):

- Select any positive integer k (it can be different in different operations).
- Choose any position in the sequence (possibly the beginning or end of the sequence, or in between any two elements) and insert k into the sequence at this position.
- This way, the sequence a changes, and the next operation is performed on this changed sequence.

For example, if $a = [3, 3, 4]$ and he selects $k = 2$, then after the operation he can obtain one of the sequences $[\underline{2}, 3, 3, 4]$, $[3, \underline{2}, 3, 4]$, $[3, 3, \underline{2}, 4]$, or $[3, 3, 4, \underline{2}]$.

Shohag wants this sequence to satisfy the following condition: for each $1 \le i \le |a|$, $a_i \le i$. Here, $|a|$ denotes the size of a.

Help him to find the minimum number of operations that he has to perform to achieve this goal. We can show that under the constraints of the problem it's always possible to achieve this goal in a finite number of operations.

Input

The first line contains a single integer t ($1 \le t \le 200$) — the number of test cases.

The first line of each test case contains a single integer n ($1 \le n \le 100$) — the initial length of the sequence.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the elements of the sequence.

Output

For each test case, print a single integer — the minimum number of operations needed to perform to achieve the goal mentioned in the statement.

Example

input
4 3 1 3 4 5 1 2 5 7 4 1 1 3 69 6969 696969
output
1 3 0 696966

Note

In the first test case, we have to perform at least one operation, as $a_2 = 3 > 2$. We can perform the operation $[1, 3, 4] \to [1, \underline{2}, 3, 4]$ (the newly inserted element is underlined), now the condition is satisfied.

In the second test case, Shohag can perform the following operations:

$[1, 2, 5, 7, 4] \to [1, 2, \underline{3}, 5, 7, 4] \to [1, 2, 3, \underline{4}, 5, 7, 4] \to [1, 2, 3, 4, 5, \underline{3}, 7, 4]$.

In the third test case, the sequence already satisfies the condition.

B. XOR Specia-LIS-t

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

YouKn0wWho has an integer sequence $a_1, a_2, \ldots a_n$. Now he will split the sequence a into one or more consecutive subarrays so that each element of a belongs to exactly one subarray. Let k be the number of resulting subarrays, and $h_1, h_2, \ldots, h_k$ be the

lengths of the longest increasing subsequences of corresponding subarrays.

For example, if we split $[2, 5, 3, 1, 4, 3, 2, 2, 5, 1]$ into $[2, 5, 3, 1, 4]$, $[3, 2, 2, 5]$, $[1]$, then $h = [3, 2, 1]$.

YouKn0wWho wonders if it is possible to split the sequence a in such a way that the bitwise XOR of $h_1, h_2, \ldots, h_k$ is equal to 0. You have to tell whether it is possible.

The longest increasing subsequence (LIS) of a sequence $b_1, b_2, \ldots, b_m$ is the longest sequence of valid indices $i_1, i_2, \ldots, i_k$ such that $i_1 < i_2 < \ldots < i_k$ and $b_{i_1} < b_{i_2} < \ldots < b_{i_k}$. For example, the LIS of $[2, 5, 3, 3, 5]$ is $[2, 3, 5]$, which has length 3.

An array c is a subarray of an array b if c can be obtained from b by deletion of several (possibly, zero or all) elements from the beginning and several (possibly, zero or all) elements from the end.

Input
The first line contains a single integer t ($1 \le t \le 10\,000$) — the number of test cases.

The first line of each test case contains a single integer n ($2 \le n \le 10^5$).

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

It is guaranteed that the sum of n over all test cases doesn't exceed $3 \cdot 10^5$.

Output
For each test case, print "YES" (without quotes) if it is possible to split into subarrays in the desired way, print "NO" (without quotes) otherwise. You can print each letter in any register (upper or lower).

Example

input
4 7 1 3 4 2 2 1 5 3 1 3 4 5 1 3 2 4 2 4 4 3 2 1
output
YES NO YES YES

Note
In the first test case, YouKn0wWho can split the sequence in the following way: $[1, 3, 4]$, $[2, 2]$, $[1, 5]$. This way, the LIS lengths are $h = [3, 1, 2]$, and the bitwise XOR of the LIS lengths is $3 \oplus 1 \oplus 2 = 0$.

In the second test case, it can be shown that it is impossible to split the sequence into subarrays that will satisfy the condition.

C. Di-visible Confusion

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

YouKn0wWho has an integer sequence $a_1, a_2, \ldots, a_n$. He will perform the following operation until the sequence becomes empty: select an index i such that $1 \le i \le |a|$ and a_i is **not** divisible by $(i + 1)$, and erase this element from the sequence. Here $|a|$ is the length of sequence a at the moment of operation. Note that the sequence a changes and the next operation is performed on this changed sequence.

For example, if $a = [3, 5, 4, 5]$, then he can select $i = 2$, because $a_2 = 5$ is not divisible by $i + 1 = 3$. After this operation the sequence is $[3, 4, 5]$.

Help YouKn0wWho determine if it is possible to erase the whole sequence using the aforementioned operation.

Input
The first line contains a single integer t ($1 \le t \le 10\,000$) — the number of test cases.

The first line of each test case contains a single integer n ($1 \le n \le 10^5$).

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

It is guaranteed that the sum of n over all test cases doesn't exceed $3 \cdot 10^5$.

Output
For each test case, print "YES" (without quotes) if it is possible to erase the whole sequence using the aforementioned operation, print "NO" (without quotes) otherwise. You can print each letter in any register (upper or lower).

Example

input
5 3 1 2 3 1 2 2 7 7 10 384836991 191890310 576823355 782177068 404011431 818008580 954291757 160449218 155374934 840594328 8 6 69 696 69696 696969 6969696 69696969 696969696

output
YES NO YES YES NO

Note

In the first test case, YouKn0wWho can perform the following operations (the erased elements are underlined): $[1, \underline{2}, 3] \to [\underline{1}, 3] \to [\underline{3}] \to [\,]$.

In the second test case, it is impossible to erase the sequence as i can only be 1, and when $i = 1$, $a_1 = 2$ is divisible by $i + 1 = 2$.

D. Moderate Modular Mode

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

YouKn0wWho has two **even** integers x and y. Help him to find an integer n such that $1 \leq n \leq 2 \cdot 10^{18}$ and $n \bmod x = y \bmod n$. Here, $a \bmod b$ denotes the remainder of a after division by b. If there are multiple such integers, output any. It can be shown that such an integer always exists under the given constraints.

Input

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

The first and only line of each test case contains two integers x and y ($2 \leq x, y \leq 10^9$, both are **even**).

Output

For each test case, print a single integer n ($1 \leq n \leq 2 \cdot 10^{18}$) that satisfies the condition mentioned in the statement. If there are multiple such integers, output any. It can be shown that such an integer always exists under the given constraints.

Example

input
4 4 8 4 2 420 420 69420 42068

output
4 10 420 9969128

Note

In the first test case, $4 \bmod 4 = 8 \bmod 4 = 0$.

In the second test case, $10 \bmod 4 = 2 \bmod 10 = 2$.

In the third test case, $420 \bmod 420 = 420 \bmod 420 = 0$.

E. Extreme Extension

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

For an array b of n integers, the *extreme value* of this array is the minimum number of times (possibly, zero) the following operation has to be performed to make b **non-decreasing**:

- Select an index i such that $1 \le i \le |b|$, where $|b|$ is the current length of b.
- Replace b_i with two elements x and y such that x and y both are **positive** integers and $x + y = b_i$.
- This way, the array b changes and the next operation is performed on this modified array.

For example, if $b = [2, 4, 3]$ and index 2 gets selected, then the possible arrays after this operation are $[2, \underline{1}, \underline{3}, 3]$, $[2, \underline{2}, \underline{2}, 3]$, or $[2, \underline{3}, \underline{1}, 3]$. And consequently, for this array, this single operation is enough to make it non-decreasing: $[2, 4, 3] \to [2, \underline{2}, \underline{2}, 3]$.

It's easy to see that every array of positive integers can be made non-decreasing this way.

YouKn0wWho has an array a of n integers. Help him find the sum of *extreme values* of all nonempty subarrays of a modulo $998\,244\,353$. If a subarray appears in a multiple times, its extreme value should be counted the number of times it appears.

An array d is a subarray of an array c if d can be obtained from c by deletion of several (possibly, zero or all) elements from the beginning and several (possibly, zero or all) elements from the end.

Input
The first line contains a single integer t ($1 \le t \le 10\,000$) — the number of test cases.

The first line of each test case contains a single integer n ($1 \le n \le 10^5$).

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^5$).

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

Output
For each test case, print a single integer — the sum of *extreme values* of all subarrays of a modulo $998\,244\,353$.

Example

input
4 3 5 4 3 4 3 2 1 4 1 69 8 7264 40515 28226 92776 35285 21709 75124 48163
output
5 9 0 117

Note
Let $f(l, r)$ denote the *extreme value* of $[a_l, a_{l+1}, \ldots, a_r]$.

In the first test case,

- $f(1, 3) = 3$, because YouKn0wWho can perform the following operations on the subarray $[5, 4, 3]$ (the newly inserted elements are underlined):
  $[5, 4, 3] \to [\underline{3}, \underline{2}, 4, 3] \to [3, 2, \underline{2}, \underline{2}, 3] \to [\underline{1}, \underline{2}, 2, 2, 2, 3]$;
- $f(1, 2) = 1$, because $[5, 4] \to [\underline{2}, \underline{3}, 4]$;
- $f(2, 3) = 1$, because $[4, 3] \to [\underline{1}, \underline{3}, 3]$;
- $f(1, 1) = f(2, 2) = f(3, 3) = 0$, because they are already non-decreasing.

So the total sum of *extreme values* of all subarrays of a $= 3 + 1 + 1 + 0 + 0 + 0 = 5$.

F. Artistic Partition

time limit per test: 3 seconds
memory limit per test: 1024 megabytes
input: standard input
output: standard output

For two positive integers l and r ($l \le r$) let $c(l, r)$ denote the number of integer pairs (i, j) such that $l \le i \le j \le r$ and $\gcd(i, j) \ge l$. Here, $\gcd(i, j)$ is the greatest common divisor (GCD) of integers i and j.

YouKn0wWho has two integers n and k where $1 \le k \le n$. Let $f(n, k)$ denote the minimum of $\sum_{i=1}^{k} c(x_i + 1, x_{i+1})$ over all integer sequences $0 = x_1 < x_2 < \ldots < x_k < x_{k+1} = n$.

Help YouKn0wWho find $f(n, k)$.

Input
The first line contains a single integer t ($1 \le t \le 3 \cdot 10^5$) — the number of test cases.

The first and only line of each test case contains two integers n and k ($1 \le k \le n \le 10^5$).

Output

For each test case, print a single integer — $f(n, k)$.

Example

input
4 6 2 4 4 3 1 10 3
output
8 4 6 11

Note

In the first test case, YouKn0wWho can select the sequence $[0, 2, 6]$. So $f(6, 2) = c(1, 2) + c(3, 6) = 3 + 5 = 8$ which is the minimum possible.