Codeforces Round 1047 (Div. 3)

A. Collatz Conjecture

1 second, 256 megabytes

You are doing a research paper on the famous Collatz Conjecture. In your experiment, you start off with an integer x, and you do the following procedure k times:

- If x is even, divide x by 2.
- Otherwise, set x to $3 \cdot x + 1$.

For example, starting off with 21 and doing the procedure 5 times, you get $21 \to 64 \to 32 \to 16 \to 8 \to 4$.

After all k iterations, you are left with the final value of x. Unfortunately, you forgot the initial value. Please output any possible initial value of x.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 400$). The description of the test cases follows.

The first line of each test case contains two integers k and x ($1 \le k, x \le 20$).

Output

For each test case, print any possible initial value on a new line. It can be shown that the answer always exists.

input

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

output

```
1
10
21
```

In the first test case, since 1 is odd, performing the procedure $k = 1$ times results in $1 \cdot 3 + 1 = 4$, so 1 is a valid output.

In the second test case, since 10 is even, performing the procedure $k = 1$ times results in $\frac{10}{2} = 5$, so 10 is a valid output.

The third test case is explained in the statement.

B. Fun Permutation

2 seconds, 256 megabytes

You are given a permutation* p of size n.

Your task is to find a permutation q of size n such that $\operatorname{GCD}^\dagger(p_i + q_i, p_{i+1} + q_{i+1}) \ge 3$ for all $1 \le i < n$. In other words, the greatest common divisor of the sum of any two adjacent positions should be at least 3.

It can be shown that this is always possible.

---

*A permutation of length m is an array consisting of m distinct integers from 1 to m in arbitrary order. For example, $[2, 3, 1, 5, 4]$ is a permutation, but $[1, 2, 2]$ is not a permutation (2 appears twice in the array), and $[1, 3, 4]$ is also not a permutation ($m = 3$ but there is 4 in the array).

$^\dagger \gcd(x, y)$ denotes the greatest common divisor (GCD) of integers x and y.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

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

The second line contains n integers $p_1, p_2, \ldots, p_n$ ($1 \le p_i \le n$).

It is guaranteed that the given array forms a permutation.

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

Output

For each test case, output the permutation q on a new line. If there are multiple possible answers, you may output any.

input

```
3
3
1 3 2
5
5 1 2 4 3
7
6 7 1 5 4 3 2
```

output

```
2 3 1
4 5 1 2 3
2 1 3 7 5 6 4
```

In the first test case, $\operatorname{GCD}(1 + 2, 3 + 3) = 3 \ge 3$ and $\operatorname{GCD}(3 + 3, 2 + 1) = 3 \ge 3$, so the output is correct.

C. Maximum Even Sum

2 seconds, 256 megabytes

You are given two integers a and b. You are to perform the following procedure:

First, you choose an integer k such that b is divisible by k. Then, you simultaneously multiply a by k and divide b by k.

Find the greatest possible **even** value of $a + b$. If it is impossible to make $a + b$ even, output -1 instead.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains two integers a and b ($1 \le a, b \le a \cdot b \le 10^{18}$).

Output

For each test case, output the maximum **even** value of $a + b$ on a new line.

input

```
7
8 1
1 8
7 7
2 6
9 16
1 6
4 6
```

output

```
-1
6
50
8
74
-1
14
```

In the first test case, it can be shown it is impossible for $a + b$ to be even.

In the second test case, the optimal k is 2. The sum is $2 + 4 = 6$.

D. Replace with Occurrences

2 seconds, 256 megabytes

Given an array a, let $f(x)$ be the number of occurrences of x in the array a. For example, when $a = [1, 2, 3, 1, 4]$, then $f(1) = 2$ and $f(3) = 1$.

You have an array b of size n. Please determine if there is an array a of size n such that $f(a_i) = b_i$ for all $1 \le i \le n$. If there is one, construct it.

Input

Each test contains multiple test cases. The first line contains the number of test cases t $(1 \le t \le 10^4)$. The description of the test cases follows.

The first line of each test case contains an integer n $(1 \le n \le 2 \cdot 10^5)$.

The second line contains n integers $b_1, b_2, \ldots, b_n$ $(1 \le b_i \le n)$.

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

Output

For each test case, output -1 if there is no valid array a.

Otherwise, print the array a of n integers on a new line. The elements should satisfy $1 \le a_i \le n$.

input

```
3
4
1 2 3 4
6
1 2 2 3 3 3
6
6 6 6 6 6 6
```

output

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

In the first test case, it can be shown that no array matches the requirement.

In the second test case, $4, 5, 6$ appear $1, 2, 3$ times respectively. Thus, the output array is correct as $f(4), f(5), f(5), f(6), f(6), f(6)$ are $1, 2, 2, 3, 3, 3$ respectively.

The problem statement has recently been changed. View the changes. ×

E. Mexification

2 seconds, 256 megabytes

You are given an array a of size n and an integer k. You do the following procedure k times:

- For each element a_i, you set a_i to $\text{mex}^*(a_1, a_2, \ldots, a_{i-1}, a_{i+1}, a_{i+2}, \ldots, a_n)$. In other words, you set a_i to the mex of all other elements in the array. **This is done for all elements in the array at the same time.**

Please find the sum of elements in the array after all k operations.

* The minimum excluded (MEX) of a collection of integers $d_1, d_2, \ldots, d_k$ is defined as the smallest non-negative integer x which does not occur in the collection d.

Input

Each test contains multiple test cases. The first line contains the number of test cases t $(1 \le t \le 10^4)$. The description of the test cases follows.

The first line contains two integers n and k ($2 \le n \le 2 \cdot 10^5, 1 \le k \le 10^9$) – the number of elements in a and the number of operations done.

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

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

Output

For each test case, output the sum of elements after all k operations on a new line.

input

```
5
3 3
0 2 1
2 4
0 2
4 1
0 0 1 1
8 7
6 6 2 4 3 0 1 8
2 2
0 0
```

output

```
3
1
8
25
0
```

In the first test case, we performed the operation on the array $[0, 2, 1]$ three times. Let's compute the result after the first time:

- The first element becomes $\text{MEX}(2, 1) = 0$
- The second element becomes $\text{MEX}(0, 1) = 2$
- The third element becomes $\text{MEX}(0, 2) = 1$

So, after the first operation, $[0, 2, 1]$ becomes $[0, 2, 1]$ again. It can be shown that the array will not change no matter how many times we perform the operation, so the final array after three operations is still $[0, 2, 1]$. The sum is $0 + 2 + 1 = 3$.

In the third test case, the array becomes $[2, 2, 2, 2]$.

F. Prefix Maximum Invariance

3 seconds, 256 megabytes

Given two arrays x and y both of size m, let z be another array of size m such that the prefix maximum at each position of z is the same as the prefix maximum at each position of x. Formally, $\max(x_1, x_2, \ldots, x_i) = \max(z_1, z_2, \ldots, z_i)$ should hold for all $1 \le i \le m$. Define $f(x, y)$ to be the maximum number of positions where $z_i = y_i$ over all possible arrays z.

You are given two sequences of integers a and b, both of size n. Please find the value of $\sum_{l=1}^{n} \sum_{r=l}^{n} f([a_l, a_{l+1}, \ldots, a_r], [b_l, b_{l+1}, \ldots, b_r])$.

Input

Each test contains multiple test cases. The first line contains the number of test cases t $(1 \le t \le 10^4)$. The description of the test cases follows.

The first line of each test case contains an integer n $(1 \le n \le 2 \cdot 10^5)$.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 2 \cdot n)$.

The third line contains n integers $b_1, b_2, \ldots, b_n$ $(1 \le b_i \le 2 \cdot n)$.

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

Output

For each test case, output the sum of $f([a_l, a_{l+1}, \ldots, a_r], [b_l, b_{l+1}, \ldots, b_r])$ over all pairs of (l, r).

input

```
6
3
5 3 1
4 2 1
5
1 2 3 4 5
1 2 3 4 5
6
7 1 12 10 5 8
9 2 4 3 6 5
1
1
2
6
5 1 2 6 3 4
3 1 6 5 2 4
2
2 2
1 1
```

output

```
5
35
26
0
24
1
```

In the first test case, the answer is the sum of the following:

- $f([5],[4]) = 0$, using $z = [5]$.
- $f([3],[2]) = 0$, using $z = [3]$.
- $f([1],[1]) = 1$, using $z = [1]$.
- $f([5,3],[4,2]) = 1$, using $z = [5,2]$.
- $f([3,1],[2,1]) = 1$, using $z = [3,1]$.
- $f([5,3,1],[4,2,1]) = 2$, using $z = [5,2,1]$.

G. Cry Me a River

2 seconds, 256 megabytes

There is a directed acyclic graph with n nodes and m edges. Each node is initially colored blue.

Let's define the *fun graph game* as follows:

- Initially, a token is placed on node s.
- Cry and River take turns moving the token to a node such that there exists a directed edge from its current position to that node, with Cry going first.
- Cry wins if the token ever reaches a node with no outgoing edges, after either player's turn.
- River wins if the token reaches a red node after either player's turn.
- **If the players reach a node that is both red and does not have outgoing edges, River wins.**

Since the graph is acyclic, it can be shown that the game always ends in a finite number of turns.

Note that Cry and River can win the game immediately if the starting node s doesn't have outgoing edges, or is red respectively.

You must handle q queries of the following kind:

- `1 u`: update the color of node u to red. It is guaranteed that node u was blue before this update.
- `2 u`: If a *fun graph game* is played with the token initially at node u, and both players play optimally, does Cry win?

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains three integers n, m, q ($2 \le n \le 2 \cdot 10^5$,$1 \le m, q \le 2 \cdot 10^5$).

The following m lines each contain two integers u and v ($1 \le u, v \le n$), meaning that there is an edge from u to v.

The following q lines each contain two integers x and u ($1 \le x \le 2, 1 \le u \le n$) – denoting the type of query and the node that the query is done on.

It is guaranteed that the given graph is a directed acyclic graph. Additionally, no edge is given more than once.

It is guaranteed that the sum of n, the sum of m, and the sum of q each do not exceed $2 \cdot 10^5$ over all test cases.

Output

For each query of the second type, output `YES` if Cry wins. Otherwise, output `NO`. Each letter may be outputted in uppercase or lowercase.

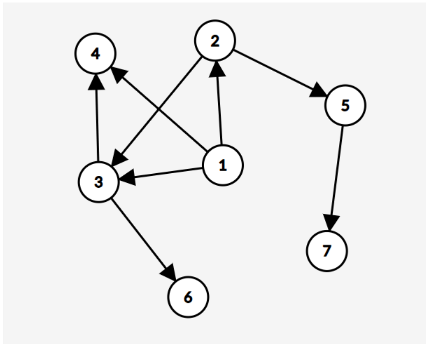
input

```
1
7 8 10
1 2
1 3
1 4
2 5
3 6
5 7
2 3
3 4
2 1
1 3
1 4
2 1
2 2
2 3
2 4
2 5
2 6
2 7
```

output

```
YES
NO
YES
NO
NO
YES
YES
YES
```

Below shows the graph in the sample.



Initially, all nodes are blue. Thus, Cry cannot lose, and eventually the token will be moved to a node without outgoing edges.

After nodes 3 and 4 are painted red, the nodes $1, 3, 4$ now start off as a win for River when playing optimally. If the game starts at nodes 3 and 4, River wins immediately. If the game starts at node 1, one way the game can go is as follows:

- Cry moves the token to node 2.
- River moves the token to node 3, which is red, so River wins.

.

It can be shown that Cry still wins with optimal play for all other nodes.