ACM-ICPC Asia-Amritapuri Site,

Problem set for Online Round 2015

AMR15A

Mahasena

Kattapa, as you all know was one of the greatest warriors of his time. The kingdom of Maahishmati had never lost a battle under him (as army-chief), and the reason for that was their really powerful army, also called as **Mahasena.**

Kattapa was known to be a very superstitious person. He believed that a soldier is "lucky" if the soldier is holding an **even number** of weapons, and "unlucky" otherwise. He considered the army as "READY FOR BATTLE" if the count of "lucky" soldiers is strictly greater than the count of "unlucky" soldiers, and "NOT READY" otherwise.

Given the number of weapons each soldier is holding, your task is to determine whether the army formed by all these soldiers is "READY FOR BATTLE" or "NOT READY".

Note: You can find the definition of an even number here.

Input

The first line of input consists of a single integer **N** denoting the number of soldiers. The second line of input consists of **N** space separated integers A_1, A_2, ..., A_N, where A_i denotes the number of weapons that the i^{th} soldier is holding.

Output

Generate one line output saying "READY FOR BATTLE", if the army satisfies the conditions that Kattapa requires or "NOT READY" otherwise (quotes for clarity).

Constraints

- $1 \le N \le 100$
- $1 \le A_i \le 100$

Example 1

Input:
```
1
1
```

Output:
```
NOT READY
```

Example 2

Input:
```
1
2
```

Output:
```
READY FOR BATTLE
```

Example 3

Input:
```
4
11 12 13 14
```

Output:
```
NOT READY
```

Example 4

Input:
```
3
2 3 4
```

Output:
```
READY FOR BATTLE
```

Example 5

Input:
```
5
1 2 3 4 5
```

Output:
```
NOT READY
```

Explanation

- **Example 1**: For the first example, **N = 1** and the array **A = [1]**. There is only 1 soldier and he is holding 1 weapon, which is odd. The number of soldiers holding an even number of weapons = 0, and number of soldiers holding an odd number of weapons = 1. Hence, the answer is "NOT READY" since the number of soldiers holding an even number of weapons is not greater than the number of soldiers holding an odd number of weapons.

- **Example 2**: For the second example, **N = 1** and the array **A = [2]**. There is only 1 soldier and he is holding 2 weapons, which is even. The number of soldiers holding an even number of weapons = 1, and number of soldiers holding an odd number of weapons = 0. Hence, the answer is "READY FOR BATTLE" since the number of soldiers holding an even number of weapons is greater than the number of soldiers holding an odd number of weapons.

- **Example 3**: For the third example, **N = 4** and the array **A = [11, 12, 13, 14]**. The 1st soldier is holding 11 weapons (which is odd), the 2nd soldier is holding 12 weapons (which is even), the 3rd soldier is holding 13 weapons (which is odd), and the 4th soldier is holding 14 weapons (which is even). The number of soldiers holding an even number of weapons = 2, and number of soldiers holding an odd number of weapons = 2. Notice that we have an **equal** number of people holding even number of weapons and odd number of weapons. The answer here is "NOT READY" since the number of soldiers holding an even number of weapons is **not strictly greater than** the number of soldiers holding an odd number of weapons.

- **Example 4**: For the fourth example, **N = 3** and the array **A = [2, 3, 4]**. The 1st soldier is holding 2 weapons (which is even), the 2nd soldier is holding 3 weapons (which is odd), and the 3rdsoldier is holding 4 weapons (which is even). The number of soldiers holding an even number of weapons = 2, and number of soldiers holding an odd number of weapons = 1. Hence, the answer is "READY FOR BATTLE" since the number of soldiers holding an even number of weapons is greater than the number of soldiers holding an odd number of weapons.

- **Example 5**: For the fifth example, **N = 5** and the array **A = [1, 2, 3, 4, 5]**. The 1st soldier is holding 1 weapon (which is odd), the 2nd soldier is holding 2 weapons (which is even), the 3rdsoldier is holding 3 weapons (which is odd), the 4th soldier is holding 4 weapons (which is even), and the 5th soldier is holding 5 weapons (which is odd). The number of soldiers holding an even number of weapons = 2, and number of soldiers holding an odd number of weapons = 3. Hence, the answer is "NOT READY" since the number of soldiers holding an even number of weapons is not greater than the number of soldiers holding an odd number of weapons.

AMR15B

Devasena

Devasena was the princess of an unknown kingdom (we'll all know which one, in Baahubali 2 ;)), and her father arranged for a Swayamvara to get her married. He gave all of them a question, and the question was so hard that nobody was able to answer it (Yes you guessed it right, both Baahubali and Bhallaladeva were not present there for the Swayamvara). We all know that Amarendra Baahubali married her later and as a consequence of so many things, she was imprisoned for 25 years. Who knows, if someone else had answered the question that day, then she would have got married to him and things could have been different. Phew! But the bad part, we wouldn't have had the story of Baahubali. :)

Although everything is history now, recently archaeologists discovered the secret question that was asked at the Swayamvara, and you think - "Well, it's such a simple question. I could use a computer to solve it!". The question goes as follows:

You are given **N** integers (not necessarily distinct) => $A_1, A_2, A_3, \ldots, A_N$. There are 2^N possible subsets (including the empty subset). The **GCD** of a subset is defined as the greatest common divisor of all the integers in that subset. You need to find the product of the **GCDs** of all the 2^N possible subsets you can construct from **A**. Since the answer can be large, you need to output the answer modulo **1000000007**. Do you think you can solve this question?

Note: The greatest common divisor of an empty subset is **1**. To know more about the definition of greatest common divisor, check here.

Input

The first line of input consists of a single integer **T** denoting the number of test cases. The description of **T** test cases follow. The first line of each test case consists of a single integer **N**. The second line of each test case consists of **N** space separated integers $A_1, A_2, \ldots, A_N$

Output

For each test case, output an single integer on a separate line denoting the answer for that test case. Note that you need to output all the values modulo **1000000007 (10^9 + 7)**.

Constraints

- **$1 \le T \le 30$**
- **$1 \le N \le 10^5$**
- **$1 \le A_i \le 10^5$**

Example

```
Input:
3
1
1
2
1 2
3
1 2 2

Output:
1
2
8
```

Explanation

For the first test case, **N = 1**, and **A = [1].** There are 2 possible subsets => **{}, {1}** and both have a gcd of 1. Hence, the answer is 1 * 1 = **1.**

For the second test case, **N = 2**, and **A = [1, 2].** There are 4 possible subsets => **{}, {1}, {2}, {1, 2}**, having a gcd of 1, 1, 2 and 1 respectively. Hence, the answer is 1 * 1 * 2 * 1 = **2.**

For the third test case, **N = 3**, and **A = [1, 2, 2].** There are 8 possible subsets => **{}, {1}, {2}, {2}, {1, 2}, {2, 2}, {1, 2}, {1, 2, 2}**, having a gcd of 1, 1, 2, 2, 1, 2, 1, 1. Hence, the answer is 1 * 1 * 2 * 2 * 1 * 2 * 1 * 1 = **8.**

AMR15C

Kalakeya

The **Kalakeyas** were a powerful, ferocious and cruel clan of Danavas. They were known to be really strong and they did not have any war strategy. They would just attack the enemy randomly and overpower them with sheer number of soldiers. However, we all know that Baahubali and Bhallaladeva defeated the Kalakeyas by following the **Thrishul** strategy, and successfully defended their kingdom Maahishmati. We also know that Baahubali was very smart, and the truth is that he predicted how the Kalakeyas would attack and devised a counter strategy for the same, the night before the war. This is what he found:

The Kalakeyas had **N** forts, numbered **1** to **N** and Baahubali had **N** soldiers, numbered **1** to **N**. Baahubali discovered that he can permute his soldiers in any way to get a permutation of **1** to **N** =>$P_1, P_2, ..., P_N$. He would then send his soldiers to attack the forts in the following way: soldier P_1 attacks **fort 1**, soldier P_2 attacks **fort 2**, ..., soldier P_N attacks **fort N**. It is easy to note that each soldier attacks exactly one fort and no two soldiers attack the same fort. Baahubali also got to know about a secret key of the Kalakeyas, which is an integer **K**. A soldier **X** can destroy a fort **Y**, iff**abs(X - Y) >= K**. For more details on the **abs()** function, check here.

Your task is to determine whether Baahubali's soldiers can be permuted in some way, such that all forts can be destroyed. In other words, for a permutation $P_1, P_2, ..., P_N$, Baahubali's soldiers can destroy all the forts iff $abs(P_i - i) >= K$, for all $1 <= i <= N$. If this is possible, you are also required to output the **lexicographically** smallest such permutation. If it is not possible, output **-1**.

Note: A permutation $A_1, A_2, ..., A_N$ is said to be lexicographically smaller than a permutation $B_1, B_2, ..., B_N$, if and only if at the first **i** where A_i and B_i differ, A_i comes before B_i. You can refer herefor a more detailed definition of lexicographic ordering.

Input

The first line of input consists of a single integer **T** denoting the number of test cases. Each of the following **T** lines contain two space separated integers **N** and **K** denoting the values mentioned in the statement above.

Output

For each test case, output a single line containing **N** space separated integers (which should be a permutation of **[1..N]**, if Baahubali's soldiers can break all the forts. If it is not possible to break all the forts, output "-1" (quotes for clarity).

Constraints

- $1 \leq T \leq 1000$
- $1 \leq N \leq 10^5$
- $0 \leq K \leq N$
- The sum of **N** over all test cases in a single test file will not exceed 10^5

Example

```
Input:
3
2 2
3 0
3 1

Output:
-1
1 2 3
2 3 1
```

Explanation

For the first test case, **N = 2** and **K = 2**. It is impossible to permute **[1, 2]** in any way such that **abs(P[1]-1) >= 2** and **abs(P[2]-2) >= 2**. Hence, output is **-1**.

For the second test case, **N = 3** and **K = 0**. We can just set **P[i] = i**, and hence the answer is **1 2 3**

For the third case, the valid permutations are **[2, 3, 1]** and **[3, 1, 2]**. The answer is **[2, 3, 1]** since it is lexicographically smaller than **[3, 1, 2]**.

AMR15D

Bhallaladeva

Bhallaladeva was an evil king who ruled the kingdom of Maahishmati. He wanted to erect a 100ft golden statue of himself and he looted gold from several places for this. He even looted his own people, by using the following unfair strategy:

There are **N** houses in Maahishmati, and the **i**$^{\mathbf{th}}$ house has $\mathbf{A_i}$ gold plates. Each gold plate costs exactly 1 **Nimbda**, which is the unit of currency in the kingdom of Maahishmati. Bhallaladeva would choose an integer **K**, and loots all the houses in several steps. In each step:

1. He would choose a house **i** which hasn't been looted yet, pay the owner exactly $\mathbf{A_i}$ Nimbdas, and take away all the gold plates in that house (Hence, he also ends up looting this house).
2. He would now choose **atmost K** houses which haven't been looted yet and take away all the gold plates from these houses without paying a single Nimbda (Yes, he takes all of them for free).

He repeats the above steps until all the **N** houses have been looted. Your task is to devise a strategy for Bhallaladeva to loot the houses in some order, so that the number of nimbdas he has to pay is **minimium**. You'll also be given multiple values of **K** (**Q** of them to be precise), and you need to find the minimum number of nimbdas for each of these values.

Input

The first line of input consists of a single integer **N** denoting the number of houses in Maahishmati. The second line of input consists of **N** space separated integers denoting $\mathbf{A_1}$**,** $\mathbf{A_2}$**, ...,** $\mathbf{A_N}$, where$\mathbf{A_i}$ denotes the number of gold plates in the **i**$^{\mathbf{th}}$ house. The third line of input consists of a single integer **Q** denoting the number of values of **K** to follow. Each of the following **Q** lines consist of a single integer, where the value on the **i**$^{\mathbf{th}}$ line denotes the value of K for the **i**$^{\mathbf{th}}$ query.

Output

Output exactly **Q** integers on separate lines, where the output on the **i**th line denotes the answer for the **i**th value of **K.**

Constraints

- $\mathbf{1 \leq N \leq 10^5}$
- $\mathbf{1 \leq Q \leq 10^5}$
- $\mathbf{0 \leq K \leq N-1}$
- $\mathbf{1 \leq A_i \leq 10^4}$

Example

```
Input:
4
3 2 1 4
2
0
2

Output:
10
3
```

Explanation

For the first query, **K = 0**. Hence, Bhallaladeva cannot take gold plates from any of the houses for free. It will cost him 3 + 2 + 1 + 4 = **10** nimbdas.

For the second query, **K = 2**. In the first step Bhallaladeva can pay **2** nimbdas for gold plates in house number 2, and take the gold in houses 1 and 4 for free (Note that house 1 has 3 gold plates and house 4 has 4 gold plates). Now, he has looted houses 1, 2 & 4. Now in the second step, he loots house 3, by paying **1** nimbda. Hence, the total cost = 1 + 2 = **3.** Note that there might be multiple ways to achieve the minimum cost, and we have explained only one of them.

AMR15E

Avantika

Young man named Baahubali found a mask next to a giant waterfalls. He imagined that it belonged to a beautiful girl. He was so determined to see the smile on her face that he did the unthinkable - he climbed the mighty waterfalls. When he reached the top, he met Avantika - the girl behind the mask and fell in love with her. He was shocked to learn that she was a warrior, but he was even more impressed after watching her fighting skills.

One fine day, Baahubali made a tattoo on Avantika's hand without her knowledge, and she was eventually very impressed. She asked Shiva to make more tattoos for her, and he sure doesn't want to disappoint her.

Initially, Baahubali has **N** bottles of paint, where each bottle is filled with paint of some color. More specifically, bottle **i** contains paint of color $\mathbf{A_i}$, and two different bottles may contain paint of the same color. Baahubali uses 2 values **L** and **R**.

He has a secret formula which is the following: He chooses all the unique paints from bottles in the range **[L, R]** (both inclusive). He then arranges the paints in increasing order of their color ($\mathbf{A_i}$). Assume that there are **K** unique paints in the range **[L, R]**, and after arranging them in increasing order of their color value, he gets $\mathbf{B_1, B_2, ..., B_K}$. He now mixes all of them by taking exactly**i** units of paint $\mathbf{B_i}$ and summing them up to get a new colored paint. More specifically, he takes $\mathbf{sum(i * B_i)}$ to get a new colored paint.

Avantika asked him to make **M** tattoos for her. Your task is to help Baahubali find the value of the new color he gets using the above method, given the $\mathbf{L_i}$ and $\mathbf{R_i}$ values for each of the M tattoos.

Input

The first line of input consists of a single integer **T** denoting the number of test cases. The description of **T** test cases follow. The first line of each test case consists of a single integer **N**, denoting the number of bottles of paint that Baahubali has. The second line of each test case consists of **N** space separated integers $A_1, A_2, ..., A_N$, where A_i denotes the color of the paint in the i^{th} bottle. The next line of each test case consists of a single integer **M**, denoting the number of tattoos that Baahubali has to make. Each of the following **M** lines contain 2 space separated integers, where the values on the j^{th} line denote L_j and R_j for the j^{th} tattoo.

Output

For each tattoo, output the value of the color that Baahubali creates using the formula mentioned in the statement above.

Constraints

- $1 \le T \le 20$
- $1 \le N \le 10^4$
- $1 \le M \le 10^4$
- $1 \le A_i \le 10^9$
- $1 \le L_i \le R_i \le N$

Example

```
Input:
1
5
1 2 3 2 1
5
1 1
1 2
2 4
1 4
1 5
Output:
1
5
8
14
14
```

Explanation

For the first tattoo, **L = 1** and **R = 1**, and there is only 1 unique color (= **1**) in that range. Hence, the answer is **1**.

For the second tattoo, there are 2 unique colors **[1, 2]**, and the new color formed is 1*1 + 2*2 = **5.**

For the third tattoo, there are 2 unique colors **[2, 3]**, and the new color formed is 1*2 + 2*3 = **8.**

For the fourth and fifth tattoos, there are 3 unique colors **[1, 2, 3]**, and the new color formed is 1*1 + 2*2 + 3*3 = **14.**