Problem statements of ACM ICPC India Online Round

Problem Code: ICPC16A
Problem Name: Help Lost Robot!

Robot Bunny is lost. It wants to reach its home as soon as possible. Currently it is standing at coordinates **(x_1, y_1)** in 2-D plane. Its home is at coordinates **(x_2, y_2)**. Bunny is extremely worried. Please help it by giving a command by telling the direction in which it should to go so as to reach its home. If you give it a direction, it will keep moving in that direction till it reaches its home. There are four possible directions you can give as command - **"left", "right", "up", "down"**. It might be possible that you can't instruct the robot in such a way that it reaches its home. In that case, output **"sad"**.

Input

First line of the input contains an integer **T** denoting the number of test cases. **T** test cases follow.

First line of each test case contains four space separated integers **x_1, y_1, x_2, y_2**.

Output

For each test case, output a single line
containing **"left"** or **"right"** or **"up"** or **"down"** or **"sad"** (without quotes).

Constraints

- **$1 \le T \le 5000$**
- **$0 \le x_1, y_1, x_2, y_2. \le 100$**
- It's guaranteed that the initial position of robot is not his home.

Example

Input
3
0 0 1 0
0 0 0 1
0 0 1 1

Output:
right
up
sad

Explanation

Test case 1. If you give Bunny the command to move to the right, it will reach its home.

Problem Code: ICPC16B
Problem Name: Beautiful Arrays

An array **a** is called *beautiful* if for every pair of numbers $\mathbf{a_i}$**,** $\mathbf{a_j}$, $(\mathbf{i \neq j})$, there exists an $\mathbf{a_k}$ such that $\mathbf{a_k = a_i * a_j}$. Note that **k** can be equal to **i** or **j** too.

Find out whether the given array **a** is *beautiful* or not!

Input

First line of the input contains an integer **T** denoting the number of test cases. **T** test cases follow.

First line of each test case contains an integer **n** denoting number of elements in **a**.

Next line contains **n** space separated integers denoting the array **a**.

Output

For each test case, output a single line containing "yes" or "no" (without quotes) corresponding to the answer of the problem.

Constraints

- $\mathbf{1 \leq T \leq 10^6}$
- $\mathbf{1 \leq n \leq 10^5}$
- Sum of **n** over all the test cases ≤ $\mathbf{10^6}$
- $\mathbf{-10^9 \leq a_i \leq 10^9}$

Example

Input
3
2
0 1
2
1 2
2

5 6

Output:
yes
yes
no

Explanation

Test case 1. If you multiply 0 with 1, you get 0, we see that $\mathbf{a_0 = 0}$. So, the array is *beautiful*.

Test case 3. If you multiply 5 with 6, you get 30, there does not exist an **k** such that $\mathbf{a_k = 30}$. So, the array is not *beautiful*.

Problem Code: ICPC16C
Problem Name: Watson and Digit Sums

Watson is going to play mind games with Sherlock. He gives an integer **d** to Sherlock.

Among all possible positive integers **N** having sum of digits **d**, consider the integer **N+1**, and find the minimum possible digit sum **N+1** can have.

For example, if **d** = 9, then **N** could be 9, 18, 27, 36,711 and so on. However, if Sherlock chooses **N** = 9, then sum of digits of **N + 1** *i.e.* 10 will be 1, which is the minimum possible.

Input

First line of the input contains an integer **T**, the number of test cases.

The only line of each test case consists of an integer **d**.

Output

For each test case output the required answer in one line.

Constraints

- $\mathbf{1 \leq T \leq 100}$
- $\mathbf{1 \leq d \leq 100000}$

Example

Input:
1
5

Output:
6

Explanation

Possible value of **N** in given case are: 32,23,113,131 and so on, in each case digit sum of **(N+1)** would be **6**.

Problem Code: ICPC16D
Problem Name: Good Sets

You are given array **a** consisting of **n** distinct integers. A set **s** of numbers is called *good* if you can rearrange the elements in such a way that each element divides the next element in the order, i.e. $s_i \mid s_{i+1}$, such that $i < |s|$, where $|s|$ denotes size of set $|s|$.

Find out number of distinct *good* sets that you can create using the values of the array. You can use one value only once in a particular set; ie. a set cannot have duplicate values. Two sets are said to be different from each other if there exists an element in one set, which does not exist in the other.

As the answer could be large, print your answer modulo $10^9 + 7$.

Input

First line of the input contains an integer **T** denoting the number of test cases. **T** test cases follow.

First line of each test case contains an integer **n** denoting number of elements in array **a**.

Next line contains **n** space separated integers denoting the elements of array **a**.

Output

For each test case, output a single line containing the corresponding answer.

Constraints

- $1 \leq T \leq 3$
- $1 \leq n \leq 7.5 * 10^5$
- $1 \leq a_i \leq 7.5 * 10^5$
- All the elements of array **a** are distinct.

Example

Input
2
2
1 2
3
6 2 3

Output:
3
5

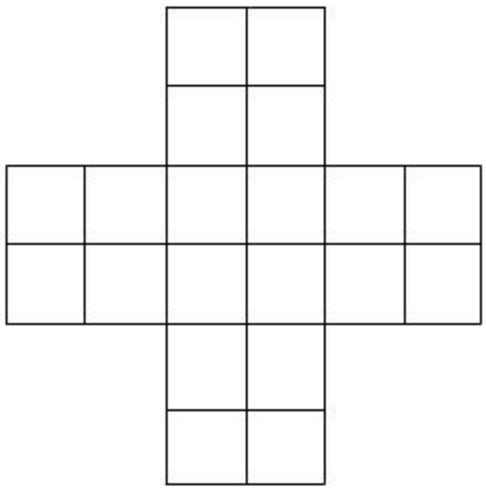
Explanation

Test case 1. There are three sets which are good **{1}**, **{2}**, **{1, 2}**.

Test case 2. There are five sets which are good **{6}**, **{2}**, **{3}**, **{6 2}**, **{6, 3}**.

Problem Code: ICPC16E
Problem Name: Colorful Grids

You are given five **N** x **N** grids, where each grid consists of N^2 unit squares. These grids have been placed in the form of a plus sign(+). For example, image below shows the figure for **N** = 2.



Your aim is to count total distinct ways to color all $5*N^2$ unit squares with **C** colors. Two ways are counted as same if one figure can be rotated in the two dimensional plane to obtain the other one.

Input

The first line of the input contains an integer **T** denoting the number of test cases. Each test case contains two space separated integers **N** and **C**, in one line.

Output

For each test case, output a single line containing the required answer modulo $10^9 + 7$.

Constraints

- $1 \le T \le 10000$
- $1 \le N \le 10^{18}$
- $1 \le C \le 10^9$

Example

Input:
2
1 1
1 2

Output:
1
12

Explanation

Example case 1. The only way is to fill all cells with same color.
Example case 2. Assume that the two colors are 1 and 2. The 12 distinct ways are:

```
  2
2 2 2
  2
```

```
  1
2 2 2
  2
```

```
  2
2 1 2
  2
```

```
  2
1 2 1
  2
```

```
  1
2 2 1
  2
```

```
  1
2 1 2
  2
```

```
  1
2 1 2
  1
```

1
2 1 1
2

1
2 2 1
1

1
2 1 1
1

1
1 2 1
1

1
1 1 1
1

Problem Code: ICPC16F
Problem Name: Chef and Bipartite Graphs

Chef is very interested in studying biparite graphs. Today he wants to construct a bipartite graph with **n** vertices each, on the two parts, and with total number of edges equal to **m**. The vertices on the left are numbered from **1** to **n**. And the vertices on the right are also numbered from **1** to **n**. He also wants the degree of every vertex to be greater than or equal to **d**, and to be lesser than or equal to **D**. ie. for all **v**, **d ≤ deg(v) ≤ D**

Given four integers, **n, m, d, D**, you have to help Chef in constructing some bipartite graph satisfying this property. If there does not exist any such graph, output -1

.

Input

First line of the input contains an integer **T** denoting the number of test cases. **T** test cases follow.

The only line of each test case contains four space separated integers **n, m, d, D**.

Output

For each test case, output **-1** if there is no bipartite graph satisfying this property. Otherwise output **m** lines, each of the lines should contain two integers **u, v** denoting that there is an edge between vertex **u** on the left part and vertex **v** on the right part. If there can be multiple possible answers, you can print any. Note that the bipartite graph should not have multi-edges.

Constraints

- $1 \le T, n \le 100$
- $1 \le d \le D \le n$
- $0 \le m \le n * n$

Example

Input
2
2 3 1 2
2 3 1 1

Output:
1 1
2 2
1 2
-1

Problem Code: ICPC16G
Problem Name: Robolympic Batons

Today is the Robolympics athletic event. **N** batons numbered 0 to **N - 1** are placed in a circular fashion where baton numbers increase in clockwise direction. Also, some of the batons are *special*. **M** robots start in the event by standing near a baton position and picking it up. Note that a baton cannot be held by more than one robot, therefore all robots have distinct positions in the circular array of batons.

Now, this event lasts for **N** seconds. In each second, each robot drops the baton currently in its hand and moves to next position in clockwise direction *i.e.* if a robot has baton **i** in its hand at **t = 0**, at **t = 1** he will be holding baton **(i + 1) % N**.

Crowd goes berserk whenever there comes a moment when all robots have picked up *special* batons. Your aim is to count how many times it will happen during the whole race. Note that crowd can go berserk even at **t = 0**. But since race is over at **t = N**, crowd doesn't care anymore. e. The crowd will not go berserk at **t = N**.

Input

First line contains **N** and **M** denoting the number of batons and number of robots participating in the event. Next line contains **N** space separated integers where **i**th integer is 0 or 1 denoting whether baton numbered **i - 1** is *special* or not. Next line contains **M** integers denoting the index(0 to N-1) of initial baton that robots have picked up.

Output

Output one and only integer denoting the required answer.

Constraints

- $1 \leq N \leq 5*10^5$
- $1 \leq M \leq N$

Example

Input 1:
3 1
1 0 1
0

Output 1:
2
Input 2:
4 2
1 0 1 0
1 3

Output 2:
2

Explanation

Example input 1.
Batons numbered 0 and 2 are *special*. Batons held by robots at **t = 0**: [0] Batons held by robots

at **t = 1**: [1] Batons held by robots at **t = 2**: [2] Note that at **t = 0, 2** all robots have special batons held.

Example input 2.

Batons numbered 0 and 2 are *special*. Batons held by robots at **t = 0**: [1, 3] Batons held by robots at **t = 1**: [2, 0] Batons held by robots at **t = 2**: [3, 1] Batons held by robots at **t = 3**: [0, 2] Note that at **t = 1, 3** all robots have special batons held.