A. Array Rearrangment

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

You are given two arrays a and b, each consisting of n positive integers, and an integer x. Please determine if one can rearrange the elements of b so that $a_i + b_i \leq x$ holds for each i ($1 \leq i \leq n$).

Input

The first line of input contains one integer t ($1 \leq t \leq 100$) — the number of test cases. t blocks follow, each describing an individual test case.

The first line of each test case contains two integers n and x ($1 \leq n \leq 50$; $1 \leq x \leq 1000$) — the length of arrays a and b, and the parameter x, described in the problem statement.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_1 \leq a_2 \leq \dots \leq a_n \leq x$) — the elements of array a in non-descending order.

The third line of each test case contains n integers $b_1, b_2, \ldots, b_n$ ($1 \leq b_1 \leq b_2 \leq \dots \leq b_n \leq x$) — the elements of array b in non-descending order.

Test cases are separated by a blank line.

Output

For each test case print Yes if one can rearrange the corresponding array b so that $a_i + b_i \leq x$ holds for each i ($1 \leq i \leq n$) or No otherwise.

Each character can be printed in any case.

Example

input

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

2 6
1 4
2 5

4 4
1 2 3 4
1 2 3 4

1 5
5
5
```

output

```
Yes
Yes
No
No
```

Note

In the first test case, one can rearrange b so it'll look like $[1, 2, 1]$. In this case, $1 + 1 \leq 4$; $2 + 2 \leq 4$; $3 + 1 \leq 4$.

In the second test case, one can set b to $[5, 2]$, then $1 + 5 \leq 6$; $4 + 2 \leq 6$.

In the third test case, no matter how one shuffles array b, $a_4 + b_4 = 4 + b_4 > 4$.

In the fourth test case, there is only one rearrangement of array b and it doesn't satisfy the condition since $5 + 5 > 5$.

B. Elimination

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

There is a famous olympiad, which has more than a hundred participants. The Olympiad consists of two stages: the elimination stage, and the final stage. At least a hundred participants will advance to the final stage. The elimination stage in turn consists of two contests.

A result of the elimination stage is the total score in two contests, but, unfortunately, the jury lost the final standings and has only standings for the first and for the second contest separately.

In each contest, the participants are ranked by their point score in non-increasing order. When two participants have a tie (earned the same score), they are ranked by their passport number (in accordance with local regulations, all passport numbers are distinct).

In the first contest, the participant on the 100-th place scored a points. Also, the jury checked all participants from the 1-st to the 100-th place (inclusive) in the first contest and found out that all of them have at least b points in the second contest.

Similarly, for the second contest, the participant on the 100-th place has c points. And the jury checked that all the participants from the 1-st to the 100-th place (inclusive) have at least d points in the first contest.

After two contests, all participants are ranked by their total score in two contests in non-increasing order. When participants have the same total score, tie-breaking with passport numbers is used. The **cutoff score** to qualify to the final stage is the total score of the participant on the 100-th place.

Given integers a, b, c, d, please help the jury determine the smallest possible value of the cutoff score.

Input

You need to process t test cases.

The first line contains an integer t ($1 \le t \le 3025$) — the number of test cases. Then descriptions of t test cases follow.

The first line of each test case contains four integers a, b, c, d ($0 \le a,\ b,\ c,\ d \le 9$; $d \le a$; $b \le c$).

One can show that for any test case satisfying the constraints above, there is at least one olympiad scenario possible.

Output

For each test case print a single integer — the smallest possible cutoff score in some olympiad scenario satisfying the given information.

Example

input

```
2
1 2 2 1
4 8 9 2
```

output

```
3
12
```

Note

For the first test case, consider the following olympiad scenario: there are 101 participants in the elimination stage, each having 1 point for the first contest and 2 points for the second contest. Hence the total score of the participant on the 100-th place is 3.

For the second test case, consider the following olympiad scenario:

- there are 50 participants with points 5 and 9 for the first and second contest respectively;
- 50 participants with points 4 and 8 for the first and second contest respectively;
- and 50 participants with points 2 and 9 for the first and second contest respectively.

Hence the total point score of the participant on the 100-th place is 12.

C. Division

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

Oleg's favorite subjects are History and Math, and his favorite branch of mathematics is division.

To improve his division skills, Oleg came up with t pairs of integers p_i and q_i and for each pair decided to find the **greatest** integer x_i, such that:

- p_i is divisible by x_i;
- x_i is not divisible by q_i.

Oleg is really good at division and managed to find all the answers quickly, how about you?

Input

The first line contains an integer t ($1 \le t \le 50$) — the number of pairs.

Each of the following t lines contains two integers p_i and q_i ($1 \le p_i \le 10^{18}$; $2 \le q_i \le 10^9$) — the i-th pair of integers.

Output

Print t integers: the i-th integer is the largest x_i such that p_i is divisible by x_i, but x_i is not divisible by q_i.

One can show that there is always at least one value of x_i satisfying the divisibility conditions for the given constraints.

Example

input

```
3
10 4
12 6
179 822
```

output

```
10
4
179
```

Note

For the first pair, where $p_1 = 10$ and $q_1 = 4$, the answer is $x_1 = 10$, since it is the greatest divisor of 10 and 10 is not divisible by 4.

For the second pair, where $p_2 = 12$ and $q_2 = 6$, note that

- 12 is not a valid x_2, since 12 is divisible by $q_2 = 6$;
- 6 is not valid x_2 as well: 6 is also divisible by $q_2 = 6$.

The next available divisor of $p_2 = 12$ is 4, which is the answer, since 4 is not divisible by 6.

D. Divide and Sum

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

You are given an array a of length $2n$. Consider a partition of array a into two subsequences p and q of length n each (each element of array a should be in exactly one subsequence: either in p or in q).

Let's sort p in non-decreasing order, and q in non-increasing order, we can denote the sorted versions by x and y, respectively. Then the *cost* of a partition is defined as $f(p, q) = \sum_{i=1}^{n} |x_i - y_i|$.

Find the sum of $f(p, q)$ over all correct partitions of array a. Since the answer might be too big, print its remainder modulo 998244353.

Input

The first line contains a single integer n ($1 \leq n \leq 150\,000$).

The second line contains $2n$ integers $a_1, a_2, \ldots, a_{2n}$ ($1 \leq a_i \leq 10^9$) — elements of array a.

Output

Print one integer — the answer to the problem, modulo 998244353.

Examples

input

```
1
1 4
```

output

```
6
```

input

```
2
2 1 2 1
```

output

```
12
```

input

```
3
2 2 2 2 2 2
```

output

```
0
```

input

```
5
13 8 35 94 9284 34 54 69 123 846
```

output

```
2588544
```

Note

Two partitions of an array are considered different if the sets of indices of elements included in the subsequence p are different.

In the first example, there are two correct partitions of the array a:

1. $p = [1]$, $q = [4]$, then $x = [1]$, $y = [4]$, $f(p, q) = |1 - 4| = 3$;
2. $p = [4]$, $q = [1]$, then $x = [4]$, $y = [1]$, $f(p, q) = |4 - 1| = 3$.

In the second example, there are six valid partitions of the array a:

1. $p = [2, 1]$, $q = [2, 1]$ (elements with indices 1 and 2 in the original array are selected in the subsequence p);
2. $p = [2, 2]$, $q = [1, 1]$;
3. $p = [2, 1]$, $q = [1, 2]$ (elements with indices 1 and 4 are selected in the subsequence p);
4. $p = [1, 2]$, $q = [2, 1]$;
5. $p = [1, 1]$, $q = [2, 2]$;
6. $p = [2, 1]$, $q = [2, 1]$ (elements with indices 3 and 4 are selected in the subsequence p).

E. Team-Building

time limit per test: 3 seconds

memory limit per test: 512 megabytes

input: standard input

output: standard output

The new academic year has started, and Berland's university has n first-year students. They are divided into k academic groups, however, some of the groups might be empty. Among the students, there are m pairs of acquaintances, and each acquaintance pair might be both in a common group or be in two different groups.

Alice is the curator of the first years, she wants to host an entertaining game to make everyone know each other. To do that, she will select two different academic groups and then divide the students of those groups into two teams. The game requires that there are no acquaintance pairs inside each of the teams.

Alice wonders how many pairs of groups she can select, such that it'll be possible to play a game after that. All students of the two selected groups must take part in the game.

Please note, that the teams Alice will form for the game don't need to coincide with groups the students learn in. Moreover, teams may have different sizes (or even be empty).

Input

The first line contains three integers n, m and k ($1 \le n \le 500\,000$; $0 \le m \le 500\,000$; $2 \le k \le 500\,000$) — the number of students, the number of pairs of acquaintances and the number of groups respectively.

The second line contains n integers $c_1, c_2, \dots, c_n$ ($1 \le c_i \le k$), where c_i equals to the group number of the i-th student.

Next m lines follow. The i-th of them contains two integers a_i and b_i ($1 \le a_i, b_i \le n$), denoting that students a_i and b_i are acquaintances. It's guaranteed, that $a_i \neq b_i$, and that no (unordered) pair is mentioned more than once.

Output

Print a single integer — the number of ways to choose two different groups such that it's possible to select two teams to play the game.

Examples

input

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

output

```
2
```

input

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

output

```
3
```

input

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

output

```
0
```

input

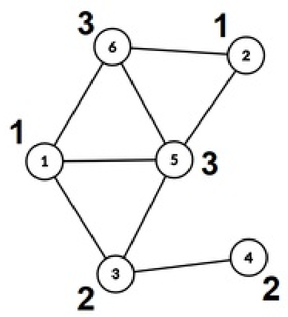
```
5 5 2
1 2 1 2 1
1 2
2 3
3 4
4 5
5 1
```

output

```
0
```

Note

The acquaintances graph for the first example is shown in the picture below (next to each student there is their group number written).



In that test we can select the following groups:

- Select the first and the second groups. For instance, one team can be formed from students 1 and 4, while other team can be formed from students 2 and 3.
- Select the second and the third group. For instance, one team can be formed 3 and 6, while other team can be formed from students 4 and 5.
- We can't select the first and the third group, because there is no way to form the teams for the game.

In the second example, we can select any group pair. Please note, that even though the third group has no students, we still can select it (with some other group) for the game.