A. Johnny and Ancient Computer

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

Johnny has recently found an ancient, broken computer. The machine has only one register, which allows one to put in there one variable. Then in one operation, you can shift its bits left or right by at most three positions. The right shift is forbidden if it **cuts off some ones**. So, in fact, in one operation, you can multiply or divide your number by 2, 4 or 8, and division is only allowed if the number is divisible by the chosen divisor.

Formally, if the register contains a positive integer x, in one operation it can be replaced by one of the following:

- $x \cdot 2$
- $x \cdot 4$
- $x \cdot 8$
- $x/2$, if x is divisible by 2
- $x/4$, if x is divisible by 4
- $x/8$, if x is divisible by 8

For example, if $x = 6$, in one operation it can be replaced by 12, 24, 48 or 3. Value 6 isn't divisible by 4 or 8, so there're only four variants of replacement.

Now Johnny wonders how many operations he needs to perform if he puts a in the register and wants to get b at the end.

Input

The input consists of multiple test cases. The first line contains an integer t ($1 \le t \le 1000$) — the number of test cases. The following t lines contain a description of test cases.

The first and only line in each test case contains integers a and b ($1 \le a, b \le 10^{18}$) — the initial and target value of the variable, respectively.

Output

Output t lines, each line should contain one integer denoting the minimum number of operations Johnny needs to perform. If Johnny cannot get b at the end, then write -1.

Example

input

```
10
10 5
11 44
17 21
1 1
96 3
2 128
1001 1100611139403776
1000000000000000000 1000000000000000000
7 1
10 8
```

output

```
1
1
-1
0
2
2
14
0
-1
-1
```

Note

In the first test case, Johnny can reach 5 from 10 by using the shift to the right by one (i.e. divide by 2).

In the second test case, Johnny can reach 44 from 11 by using the shift to the left by two (i.e. multiply by 4).

In the third test case, it is impossible for Johnny to reach 21 from 17.

In the fourth test case, initial and target values are equal, so Johnny has to do 0 operations.

In the fifth test case, Johnny can reach 3 from 96 by using two shifts to the right: one by 2, and another by 3 (i.e. divide by 4 and by 8).

B. Johnny and His Hobbies

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

Among Johnny's numerous hobbies, there are two seemingly harmless ones: applying bitwise operations and sneaking into his dad's office. As it is usually the case with small children, Johnny is unaware that combining these two activities can get him in a lot of trouble.

There is a set S containing very important numbers on his dad's desk. The minute Johnny heard about it, he decided that it's a good idea to choose a **positive** integer k and replace each element s of the set S with $s \oplus k$ ($\oplus$ denotes the exclusive or operation).

Help him choose such k that Johnny's dad will not see any difference after his son is done playing (i.e. Johnny will get the same set as before playing). It is possible that no such number exists. It is also possible that there are many of them. In such a case, output the smallest one. Note that the order of elements in a set doesn't matter, i.e. set $\{1, 2, 3\}$ equals to set $\{2, 1, 3\}$.

Formally, find the smallest positive integer k such that $\{s \oplus k | s \in S\} = S$ or report that there is no such number.

For example, if $S = \{1, 3, 4\}$ and $k = 2$, new set will be equal to $\{3, 1, 6\}$. If $S = \{0, 1, 2, 3\}$ and $k = 1$, after playing set will stay the same.

Input

In the first line of input, there is a single integer t ($1 \leq t \leq 1024$), the number of test cases. In the next lines, t test cases follow. Each of them consists of two lines.

In the first line there is a single integer n ($1 \leq n \leq 1024$) denoting the number of elements in set S. Second line consists of n **distinct** integers s_i ($0 \leq s_i < 1024$), elements of S.

It is guaranteed that the sum of n over all test cases will not exceed 1024.

Output

Print t lines; i-th line should contain the answer to the i-th test case, the minimal positive integer k satisfying the conditions or -1 if no such k exists.

Example

input

```
6
4
1 0 2 3
6
10 7 14 8 3 12
2
0 2
3
1 2 3
6
1 4 6 10 11 12
2
0 1023
```

output

```
1
4
2
-1
-1
1023
```

Note

In the first test case, the answer is 1 because it is a minimum positive integer and it satisfies all the conditions.

C. Johnny and Another Rating Drop

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

The last contest held on Johnny's favorite competitive programming platform has been received rather positively. However, Johnny's rating has dropped again! He thinks that the presented tasks are lovely, but don't show the truth about competitors' skills.

The boy is now looking at the ratings of consecutive participants written in a binary system. He thinks that the more such ratings differ, the more unfair is that such people are next to each other. He defines the difference between two numbers as the number of bit positions, where one number has zero, and another has one (we suppose that numbers are padded with leading zeros to the same length). For example, the difference of $5 = 101_2$ and $14 = 1110_2$ equals to 3, since 0101 and 1110 differ in 3 positions. Johnny defines the unfairness of the contest as the sum of such differences counted for neighboring participants.

Johnny has just sent you the rating sequence and wants you to find the unfairness of the competition. You have noticed that you've got a sequence of **consecutive** integers from 0 to n. That's strange, but the boy stubbornly says that everything is right. So help him and find the desired unfairness for received numbers.

Input

The input consists of multiple test cases. The first line contains one integer t ($1 \le t \le 10\,000$) — the number of test cases. The following t lines contain a description of test cases.

The first and only line in each test case contains a single integer n ($1 \le n \le 10^{18}$).

Output

Output t lines. For each test case, you should output a single line with one integer — the unfairness of the contest if the rating sequence equals to 0, 1, ..., $n - 1$, n.

Example

input

```
5
5
7
11
1
2000000000000
```

output

```
8
11
19
1
3999999999987
```

Note

For $n = 5$ we calculate unfairness of the following sequence (numbers from 0 to 5 written in binary with extra leading zeroes, so they all have the same length):

- 000
- 001
- 010
- 011
- 100
- 101

The differences are equal to 1, 2, 1, 3, 1 respectively, so unfairness is equal to $1 + 2 + 1 + 3 + 1 = 8$.

D. Johnny and Contribution

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Today Johnny wants to increase his contribution. His plan assumes writing n blogs. One blog covers one topic, but one topic can be covered by many blogs. Moreover, some blogs have references to each other. Each pair of blogs that are connected by a reference has to cover different topics because otherwise, the readers can notice that they are split just for more contribution. Set of blogs and bidirectional references between some pairs of them is called blogs network.

There are n different topics, numbered from 1 to n sorted by Johnny's knowledge. The structure of the blogs network is already prepared. Now Johnny has to write the blogs in some order. He is lazy, so each time before writing a blog, he looks at it's already written neighbors (the blogs referenced to current one) and chooses the topic with the smallest number which is not covered by neighbors. It's easy to see that this strategy will always allow him to choose a topic because there are at most $n - 1$ neighbors.

For example, if already written neighbors of the current blog have topics number 1, 3, 1, 5, and 2, Johnny will choose the topic number 4 for the current blog, because topics number 1, 2 and 3 are already covered by neighbors and topic number 4 isn't covered.

As a good friend, you have done some research and predicted the best topic for each blog. Can you tell Johnny, in which order he has to write the blogs, so that his strategy produces the topic assignment chosen by you?

Input

The first line contains two integers n $(1 \le n \le 5 \cdot 10^5)$ and m $(0 \le m \le 5 \cdot 10^5)$ — the number of blogs and references, respectively.

Each of the following m lines contains two integers a and b ($a \ne b$; $1 \le a, b \le n$), which mean that there is a reference between blogs a and b. It's guaranteed that the graph doesn't contain multiple edges.

The last line contains n integers $t_1, t_2, \ldots, t_n$, i-th of them denotes desired topic number of the i-th blog ($1 \le t_i \le n$).

Output

If the solution does not exist, then write -1. Otherwise, output n distinct integers $p_1, p_2, \ldots, p_n$ $(1 \le p_i \le n)$, which describe the numbers of blogs in order which Johnny should write them. If there are multiple answers, print any.

Examples

input

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

output

```
2 1 3
```

input

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

output

```
-1
```

input

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

output

```
2 5 1 3 4
```

Note

In the first example, Johnny starts with writing blog number 2, there are no already written neighbors yet, so it receives the first topic. Later he writes blog number 1, it has reference to the already written second blog, so it receives the second topic. In the end, he writes blog number 3, it has references to blogs number 1 and 2 so it receives the third topic.

Second example: There does not exist any permutation fulfilling given conditions.

Third example: First Johnny writes blog 2, it receives the topic 1. Then he writes blog 5, it receives the topic 1 too because it doesn't have reference to single already written blog 2. Then he writes blog number 1, it has reference to blog number 2 with topic 1, so it receives the topic 2. Then he writes blog number 3 which has reference to blog 2, so it receives the topic 2. Then he ends with writing blog number 4 which has reference to blog 5 and receives the topic 2.

E. Johnny and Grandmaster

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Johnny has just found the new, great tutorial: "*How to become a grandmaster?*". The tutorial tells many strange and unexpected for Johnny things, such as you have to be patient or that very important is solving many harder and harder problems.

The boy has found an online judge with tasks divided by topics they cover. He has picked p^{k_i} problems from i-th category (p is his favorite number). He wants to solve them in two weeks (the patience condition is too hard for Johnny, so for simplicity, he looks only at easy tasks, which can be solved in such a period). Now our future grandmaster has to decide which topics to cover first and which the second week. Help him assign topics in such a way, that workload is balanced.

Formally, given n numbers p^{k_i}, the boy wants to divide them into two disjoint sets, minimizing the absolute difference between sums of numbers in each set. Find the minimal absolute difference. Output the result modulo $10^9 + 7$.

Input

Input consists of multiple test cases. The first line contains one integer t $(1 \le t \le 10^5)$ — the number of test cases. Each test case is described as follows:

The first line contains two integers n and p $(1 \le n, p \le 10^6)$. The second line contains n integers k_i $(0 \le k_i \le 10^6)$.

The sum of n over all test cases doesn't exceed 10^6.

Output

Output one integer — the reminder of division the answer by $1\,000\,000\,007$.

Example

input

```
4
5 2
2 3 4 4 3
3 1
2 10 1000
4 5
0 1 1 100
1 8
89
```

output

```
4
1
146981438
747093407
```

Note

You have to minimize the difference, not it's remainder. For example, if the minimum difference is equal to 2, but there is also a distribution where the difference is $10^9 + 8$, then the answer is 2, not 1.

In the first test case of the example, there're the following numbers: 4, 8, 16, 16, and 8. We can divide them into such two sets: $4, 8, 16$ and $8, 16$. Then the difference between the sums of numbers in sets would be 4.

F. Johnny and Megan's Necklace

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Johnny's younger sister Megan had a birthday recently. Her brother has bought her a box signed as "*Your beautiful necklace — do it yourself!*". It contains many necklace parts and some magic glue.

The necklace part is a chain connecting two pearls. Color of each pearl can be defined by a non-negative integer. The magic glue allows Megan to merge two pearls (possibly from the same necklace part) into one. The beauty of a connection of pearls in colors u and v is defined as follows: let 2^k be the greatest power of two dividing $u \oplus v$ — exclusive or of u and v. Then the beauty equals k. If $u = v$, you may assume that beauty is equal to 20.

Each pearl can be combined with another at most once. Merging two parts of a necklace connects them. Using the glue multiple times, Megan can finally build the necklace, which is a cycle made from connected necklace parts (so every pearl in the necklace is combined with precisely one other pearl in it). The beauty of such a necklace is the minimum beauty of a single connection in it. The girl wants to use all available necklace parts to build **exactly one** necklace consisting of **all of them** with the largest possible beauty. Help her!

Input

The first line contains n $(1 \leq n \leq 5 \cdot 10^5)$ — the number of necklace parts in the box. Each of the next n lines contains two integers a and b $(0 \leq a, b < 2^{20})$, which denote colors of pearls presented in the necklace parts. Pearls in the i-th line have indices $2i - 1$ and $2i$ respectively.

Output

The first line should contain a single integer b denoting the maximum possible beauty of a necklace built from all given parts.

The following line should contain $2n$ distinct integers p_i $(1 \leq p_i \leq 2n)$ — the indices of initial pearls in the order in which they appear on a cycle. Indices of pearls belonging to the same necklace part have to appear at neighboring positions in this permutation (so $1\,4\,3\,2$ is not a valid output, whereas $2\,1\,4\,3$ and $4\,3\,1\,2$ are). If there are many possible answers, you can print any.

Examples

input

```
5
13 11
11 1
3 5
17 1
9 27
```

output

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

input

```
5
13 11
11 1
3 5
17 1
7 29
```

output

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

input
1 1 1

output
20 2 1

Note

In the first example the following pairs of pearls are combined: $(7, 9)$, $(10, 5)$, $(6, 1)$, $(2, 3)$ and $(4, 8)$. The beauties of connections equal correspondingly: 3, 3, 3, 20, 20.

The following drawing shows this construction.