Problem A. Minimum XOR

Time limit	1000 ms
Code length Limit	50000 B
OS	Linux

Problem Link

You have N integers - $A_1, A_2, \ldots, A_N$.

You have to make the Bitwise XOR of all the elements as minimum as possible.

You are allowed to remove at most one element. Note that this means that you can also choose to not remove any element.

What is the final minimum XOR that you can achieve after removing at most one element?

Note: In most programming languages, the XOR of two variables x and y can be computed using x ^ y.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of two lines of input.
  - The first line of each test case contains an integer N — the number of elements.
  - The next line contains N space separated integers

Output Format

For each test case, output on a new line the final minimum XOR of the elements.

Constraints

- $1 \leq T \leq 10^5$
- $1 \leq N \leq 3 \cdot 10^5$
- $1 \leq A_i \leq 10^5$

- Sum of N over all the testcases $\leq 3 \cdot 10^5$

Sample 1

Input	Output
3 4 2 4 3 6 2 4 4 5 1 3 5 17 9	0 0 14

Testcase 1: The bitwise XOR of all elements $\{2, 4, 3, 6\}$ is 3. If we remove the element 3, the total XOR of the remaining elements becomes 0 which is minimum possible XOR.

Testcase 2: The bitwise XOR of all elements $\{4, 4\}$ is 0. This is already the minimum possible total XOR, and so we will not remove any element.

Testcase 3: The bitwise XOR of all elements $\{1, 3, 5, 17, 9\}$ is 31. If we remove the element 17, the total XOR of the remaining elements becomes 14 which is minimum possible XOR.

Problem B. OR Permutation

Time limit	1000 ms
Code length Limit	50000 B
OS	Linux

Problem Link

Given an integer N, find a permutation of size N such that:

- $A_i \neq A_{i-1} \mid A_{i-2}$ for all $3 \leq i \leq N$, where $\mid$ denotes the bitwise or operation.

It is guaranteed that such permutation always exists. If multiple such permutations exist, you may print any.
Note that a permutation of size N consists of all integers from 1 to N exactly once.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists a single integer N — the size of the permutation.

Output Format

For each test case, output on a new line, N space-separated integers denoting the permutation satisfying the given conditions.

It is guaranteed that such permutation always exists. If multiple such permutations exist, you may print any.

Constraints

- $1 \leq T \leq 10^5$
- $3 \leq N \leq 10^5$
- The sum of N over all test cases won't exceed 10^6.

Sample 1

Input	Output
3 3 4 5	1 3 2 1 2 4 3 2 1 5 3 4

Problem C. Playing with OR

Time limit	1000 ms
Code length Limit	50000 B
OS	Linux

Problem Link

You are given an array A containing N integers, and an integer K ($1 \leq K \leq N$).
Find the number of *subarrays* of A with length K whose bitwise OR is odd.

Note: A subarray of A is a contiguous segment of elements of A.
For example, if $A = [1, 3, 2]$, then it has 6 non-empty subarrays:
$[1], [3], [2], [1, 3], [3, 2], [1, 3, 2]$.
In particular, $[1, 2]$ is *not* a subarray of A.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of two lines of input.
  - The first line of each test case contains two space-separated integers N and K — the length of the array and the subarray size you have to check, respectively.
  - The second line of each test contains N space-separated integers $A_1, A_2, \ldots, A_N$ — the elements of the array.

Output Format

For each test case, output on a new line the number of length-K subarrays of A whose bitwise OR is odd.

Constraints

- $1 \leq T \leq 10^5$
- $1 \leq K \leq N \leq 5 \cdot 10^5$
- $1 \leq A_i \leq 10^9$

- The sum of N across all tests doesn't exceed $5 \cdot 10^5$.

Sample 1

Input	Output
2 5 2 5 7 13 4 6 4 3 2 6 7 4	3 2

Test case 1: There are four subarrays of length $K = 2$. - $[5, 7]$, with bitwise OR equal to 7. - $[7, 13]$, with bitwise OR equal to 15. - $[13, 4]$, with bitwise OR equal to 13. - $[4, 6]$, with bitwise OR equal to 6.

Three of them are odd, so the answer is 3.

Test case 2: There are two subarrays of length three, both of them have odd bitwise OR.

Problem D. We Need the Zero

Time limit 1000 ms
Mem limit 262144 kB

Problem Link

There is an array a consisting of non-negative integers. You can choose an integer x and denote $b_i = a_i \oplus x$ for all $1 \le i \le n$, where $\oplus$ denotes the bitwise XOR operation. Is it possible to choose such a number x that the value of the expression $b_1 \oplus b_2 \oplus \ldots \oplus b_n$ equals 0?

It can be shown that if a valid number x exists, then there also exists x such that ($0 \le x < 2^8$).

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 1000$). The description of the test cases follows.

The first line of the test case contains one integer n ($1 \le n \le 10^3$) — the length of the array a.

The second line of the test case contains n integers — array a ($0 \le a_i < 2^8$).

It is guaranteed that the sum of n over all test cases does not exceed 10^3.

Output

For each set test case, print the integer x ($0 \le x < 2^8$) if it exists, or -1 otherwise.

Sample 1

Input	Output
5 3 1 2 5 3 1 2 3 4 0 1 2 3 4 1 2 2 3 1 1	6 0 3 -1 1

Note

In the first test case, after applying the operation with the number 6 the array b becomes $[7, 4, 3]$, $7 \oplus 4 \oplus 3 = 0$.

There are other answers in the third test case, such as the number 0.

Problem E. And Then There Were K

Time limit 1000 ms
Mem limit 262144 kB

Problem Link

Given an integer n, find the maximum value of integer k such that the following condition holds:

$$n \,\&\, (n-1) \,\&\, (n-2) \,\&\, (n-3) \,\&\, \ldots\, (k) = 0$$

where & denotes the bitwise AND operation.

Input

The first line contains a single integer t ($1 \leq t \leq 3 \cdot 10^4$). Then t test cases follow.

The first line of each test case contains a single integer n ($1 \leq n \leq 10^9$).

Output

For each test case, output a single integer — the required integer k.

Sample 1

Input	Output
3 2 5 17	1 3 15

Note

In the first testcase, the maximum value for which the continuous & operation gives 0 value, is 1.

In the second testcase, the maximum value for which the continuous & operation gives 0 value, is 3. No value greater then 3, say for example 4, will give the & sum 0.

- $5 \,\&\, 4 \neq 0$,

- $5 \,\&\, 4 \,\&\, 3 = 0$.

Hence, 3 is the answer.

Problem F. Rock and Lever

Time limit 1000 ms
Mem limit 262144 kB

Problem Link

"You must lift the dam. With a lever. I will give it to you.

You must block the canal. With a rock. I will not give the rock to you."

Danik urgently needs rock and lever! Obviously, the easiest way to get these things is to ask Hermit Lizard for them.

Hermit Lizard agreed to give Danik the lever. But to get a stone, Danik needs to solve the following task.

You are given a positive integer n, and an array a of positive integers. The task is to calculate the number of such pairs (i, j) that $i < j$ and $a_i \ \& \ a_j \geq a_i \oplus a_j$, where $\&$ denotes the bitwise AND operation, and $\oplus$ denotes the bitwise XOR operation.

Danik has solved this task. But can you solve it?

Input

Each test contains multiple test cases.

The first line contains one positive integer t ($1 \leq t \leq 10$) denoting the number of test cases. Description of the test cases follows.

The first line of each test case contains one positive integer n ($1 \leq n \leq 10^5$) — length of the array.

The second line contains n positive integers a_i ($1 \leq a_i \leq 10^9$) — elements of the array.

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

Output

For every test case print one non-negative integer — the answer to the problem.

Sample 1

Input	Output
5 5 1 4 3 7 10 3 1 1 1 4 6 2 5 3 2 2 4 1 1	1 3 2 0 0

Note

In the first test case there is only one pair: $(4, 7)$: for it $4 \,\&\, 7 = 4$, and $4 \oplus 7 = 3$.

In the second test case all pairs are good.

In the third test case there are two pairs: $(6, 5)$ and $(2, 3)$.

In the fourth test case there are no good pairs.

Problem G. Min Or Sum

Time limit 1000 ms
Mem limit 262144 kB

Problem Link

You are given an array a of size n.

You can perform the following operation on the array:

- Choose two different integers i, j $(1 \leq i < j \leq n)$, replace a_i with x and a_j with y. In order not to break the array, $a_i | a_j = x | y$ must be held, where $|$ denotes the bitwise OR operation. Notice that x and y are non-negative integers.

Please output the minimum sum of the array you can get after using the operation above any number of times.

Input

Each test contains multiple test cases. The first line contains the number of test cases t $(1 \leq t \leq 1000)$. Description of the test cases follows.

The first line of each test case contains an integer n $(2 \leq n \leq 100)$ — the size of array a.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ $(0 \leq a_i < 2^{30})$.

Output

For each test case, print one number in a line — the minimum possible sum of the array.

Sample 1

Input	Output
4 3 1 3 2 5 1 2 4 8 16 2 6 6 3 3 5 6	3 31 6 7

Note

In the first example, you can perform the following operations to obtain the array $[1, 0, 2]$:

1. choose $i = 1, j = 2$, change $a_1 = 1$ and $a_2 = 2$, it's valid since $1|3 = 1|2$. The array becomes $[1, 2, 2]$.

2. choose $i = 2, j = 3$, change $a_2 = 0$ and $a_3 = 2$, it's valid since $2|2 = 0|2$. The array becomes $[1, 0, 2]$.

We can prove that the minimum sum is $1 + 0 + 2 = 3$

In the second example, We don't need any operations.

Problem H. XOR Palindromes

Time limit 1000 ms
Mem limit 262144 kB
Problem Link

You are given a binary string s of length n (a string that consists only of 0 and 1). A number x is good if there exists a binary string l of length n, containing x ones, such that if each symbol s_i is replaced by $s_i \oplus l_i$ (where $\oplus$ denotes the bitwise XOR operation), then the string s becomes a palindrome.

You need to output a binary string t of length $n+1$, where t_i ($0 \leq i \leq n$) is equal to 1 if number i is good, and 0 otherwise.

A palindrome is a string that reads the same from left to right as from right to left. For example, `01010`, `1111`, `0110` are palindromes.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 10^5$). The description of the test cases follows.

The first line of each test case contains a single integer n ($1 \leq n \leq 10^5$).

The second line of each test case contains a binary string s of length n.

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

Output

For each test case, output a single line containing a binary string t of length $n+1$ - the answer to the problem.

Sample 1

Input	Output
5 6 101011 5 00000 9 100100011 3 100 1 1	0010100 111111 0011111100 0110 11

Note

Consider the first example.

- $t_2 = 1$ because we can choose $l = 010100$, then the string s becomes 111111, which is a palindrome.
- $t_4 = 1$ because we can choose $l = 101011$.
- It can be shown that for all other i, there is no answer, so the remaining symbols are 0.

Problem I. Xorry 1

Time limit	1000 ms
Code length Limit	50000 B
OS	Linux

Problem Link

You are given an integer X. Two integers (A, B) are said to be an Xorry pair, if the following conditions are satisfied:

- $A \oplus B = X$
- $0 \leq A \leq B \leq X$
- Among all the pairs which satisfy the above two conditions, $(B - A)$ is as small as possible

For example, suppose $X = 6$.
Then, the pair $(0, 6)$ satisfies the first two conditions. But, the pair $(2, 4)$ also satisfies the first two conditions. And since, $(6 - 0)$ is larger than $(4 - 2)$, the pair $(0, 6)$ does not have the smallest possible difference, and so, it is not an Xorry pair.
You can verify that the pairs $(2, 4)$ and $(3, 5)$ are the only two Xorry pairs for $X = 6$. This is explained further in the sample explanation below.

Output any one Xorry pair for a given X.

Note that $\oplus$ refers to bitwise XOR.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of a single line which contains a single integer, X.

Output Format

For each test case, output on a new line, two space separated integers, A and B, where (A, B) is an Xorry pair, for that particular X.

If there are multiple Xorry pairs, output any one of them.

Constraints

- $1 \le T \le 10^5$
- $1 \le X \le 10^9$

Sample 1

Input	Output
5 1 2 3 4 6	0 1 0 2 1 2 0 4 2 4

Test case 1: $X = 1$. The only pair of integers which satisfies the first two conditions is $A = 0$, and $B = 1$.\ This satisfies the first two conditions, since $0 \oplus 1 = 1$, and $0 \le 1 \le 1$.\ Since there is only one pair which satisfies the first two conditions, this is also the pair with the smallest $(B - A)$, and hence is the only Xorry pair.\ So, the only answer for this testcase is $(0, 1)$.

Test case 2: $X = 2 = (10)_2$. The only pair of integers which satisfies the first two conditions $A = 0$, and $B = 2$.
This satisfies the first two conditions, since $(00)_2 \oplus (10)_2 = (10)_2 = 2$, and $0 \le 2 \le 2$.
Since there is only one pair which satisfies the first two conditions, this is also the pair with the smallest $(B - A)$, and hence is the only Xorry pair.
So, the only answer for this testcase is $(0, 2)$.

Note that $(10)_2$ refers to the binary representation of 2.

Test case 3: $X = 3 = (11)_2$. There are two pairs which satisfy the first two conditions:

- $A = 0$, and $B = 3$
- $A = 1$, and $B = 2$
  $(1, 2)$ satisfies both the conditions, since $(01)_2 \oplus (10)_2 = (11)_2 = 3$, and $1 \le 2 \le 3$.
  Among these two pairs, $(B - A)$ is smallest only in the second pair. Hence the only Xorry pair is $A = 1$, and $B = 2$.
  So, the only answer for this testcase is $(1, 2)$.

Test case 4: $X = 4 = (100)_2$. The only pair of integers which satisfies the first two

This satisfies the first two conditions, since $(000)_2 \oplus (100)_2 = (100)_2 = 4$, and $0 \leq 4 \leq 4$. Since there is only one pair which satisfies the first two conditions, this is also the pair with the smallest $(B - A)$, and hence is the only Xorry pair.
So, the only answer for this testcase is $(0, 4)$.

Test case 5: $X = 6 = (110)_2$. There are three pairs which satisfy the first two conditions:

- $A = 0$, and $B = 6$
- $A = 2$, and $B = 4$
- $A = 3$, and $B = 5$
  $(3, 5)$ satisfies both the conditions, since $(011)_2 \oplus (101)_2 = (110)_2 = 6$, and $3 \leq 5 \leq 6$.
  But among these three pairs, $(B - A)$ is smallest only in the last two pairs, which have $(B - A)$ as 2, whereas the first pair has $(B - A)$ as 6. Hence the only Xorry pairs are $A = 2$, and $B = 4$, and $A = 3$, and $B = 5$.
  So, the two possible answers for this testcase are $(2, 4)$, or $(3, 5)$. You need to output any one of these two pairs.

Problem J. Xorry 2

Time limit	1000 ms
Code length Limit	50000 B
OS	Linux

Problem Link

[This problem is very similar to XORRY1. The only difference is that instead of printing any one Xorry pair, you are now asked to count the number of Xorry pairs]

You are given an integer X. Two integers (A, B) are said to be an Xorry pair, if the following conditions are satisfied:

- $A \oplus B = X$
- $0 \leq A \leq B \leq X$
- Among all the pairs which satisfy the above two conditions, $(B - A)$ is as small as possible

For example, suppose $X = 6$.
Then, the pair $(0, 6)$ satisfies the first two conditions. But, the pair $(2, 4)$ also satisfies the first two conditions. And since, $(6 - 0)$ is larger than $(4 - 2)$, the pair $(0, 6)$ does not have the smallest possible difference, and so, it is not an Xorry pair.
You can verify that the pairs $(2, 4)$ and $(3, 5)$ are the only two Xorry pairs for $X = 6$. This is explained further in the sample explanation below.

Output the number of Xorry pairs for a given X.

Note that $\oplus$ refers to bitwise XOR.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of a single line which contains a single integer, X.

Output Format

For each test case, output on a new line the number of Xorry pairs, for that particular X.

Constraints

- $1 \leq T \leq 10^5$
- $1 \leq X \leq 10^9$

Sample 1

Input	Output
5 1 2 3 4 6	1 1 1 1 2

Test case 1: $X = 1$. The only pair of integers which satisfies the first two conditions is $A = 0$, and $B = 1$.\ This satisfies the first two conditions, since $0 \oplus 1 = 1$, and $0 \leq 1 \leq 1$.\ Since there is only one pair which satisfies the first two conditions, this is also the pair with the smallest $(B - A)$, and hence is the only Xorry pair.\ So, the answer for this testcase is 1.

Test case 2: $X = 2 = (10)_2$. The only pair of integers which satisfies the first two conditions $A = 0$, and $B = 2$.
This satisfies the first two conditions, since $(00)_2 \oplus (10)_2 = (10)_2 = 2$, and $0 \leq 2 \leq 2$.
Since there is only one pair which satisfies the first two conditions, this is also the pair with the smallest $(B - A)$, and hence is the only Xorry pair.
So, the answer for this testcase is 1.

Note that $(10)_2$ refers to the binary representation of 2.

Test case 3: $X = 3 = (11)_2$. There are two pairs which satisfy the first two conditions:

- $A = 0$, and $B = 3$
- $A = 1$, and $B = 2$
  $(1, 2)$ satisfies both the conditions, since $(01)_2 \oplus (10)_2 = (11)_2 = 3$, and $1 \leq 2 \leq 3$.
  But among these two pairs, $(B - A)$ is smallest only in the second pair. Hence the only Xorry pair is $A = 1$, and $B = 2$.
  So, the answer for this testcase is 1.

Test case 4: $X = 4 = (100)_2$. The only pair of integers which satisfies the first two conditions is $A = 0$, and $B = 4$.
This satisfies the first two conditions, since $(000)_2 \oplus (100)_2 = (100)_2 = 4$, and $0 \leq 4 \leq 4$.

Since there is only one pair which satisfies the first two conditions, this is also the pair with the smallest $(B - A)$, and hence is the only Xorry pair.
So, the answer for this testcase is 1.

Test case 5: $X = 6 = (110)_2$. There are three pairs which satisfy the first two conditions:

- $A = 0$, and $B = 6$
- $A = 2$, and $B = 4$
- $A = 3$, and $B = 5$
  $(3, 5)$ satisfies both the conditions, since $(011)_2 \oplus (101)_2 = (110)_2 = 6$, and $3 \leq 5 \leq 6$.
  But among these three pairs, $(B - A)$ is smallest only in the last two pairs, which have $(B - A)$ as 2, whereas the first pair has $(B - A)$ as 6. Hence the only Xorry pairs are $A = 2$, and $B = 4$, and $A = 3$, and $B = 5$.
  So, the answer for this testcase is 2.

Problem K. Possible or Not

Time limit	1000 ms
Code length Limit	50000 B
OS	Linux

Problem Link

Om has an array A of size N and an integer B.

He wants to find whether there exists a **non-empty** subsequence of A such that the bitwise AND of all elements in the subsequence is equal to B.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of multiple lines of input.
  - The first line of each test case contains two space-separated integers N and B — the size of array and the target integer, respectively.
  - The second line of each test case contains N space-separated integers, the i^{th} integer representing the element at the i^{th} index.

Output Format

For each test case, output `YES` if any such subsequence exists and `NO` otherwise.

You may print each character in uppercase or lowercase. For example `YES`, `Yes`, `yes`, and `yES` are all considered the same.

Constraints

- $1 \leq T \leq 10^5$
- $1 \leq N \leq 10^5$
- $0 \leq B < 2^{30}$
- $0 \leq A_i < 2^{30}$

- The sum of N over all test cases won't exceed $6 \cdot 10^5$.

Sample 1

Input	Output
2 6 5 1 7 3 4 2 13 3 2 1 3 4	YES NO

Test case 1: Consider the subsequence $[A_2, A_6]$. The bitwise AND of the elements is $7 \& 13 = 5$.

Test case 2: It can be shown that no subsequence satisfies the condition.

Problem L. Boxes

Time limit	2500 ms
Code length Limit	50000 B
OS	Linux

Problem Link

You are given an array A of length N such that:

- $2 \times (A_1 \oplus A_2 \oplus \ldots \oplus A_N) \geq (A_1 | A_2 | \ldots | A_N)$, where $\oplus$ and $|$ denote the bitwise xor and or operations respectively.

Your task is to perform the following operation until the array is empty:

- Choose any element A_i from the array.
- Remove A_i from the array.
- Perform **one** of the following:
  - Use a new box and add A_i to the box.
  - Select an existing box such that the XOR of all the elements in the box (before placing A_i) is **greater than or equal to** A_i and add A_i to the box.

Determine the **minimum** number of boxes required to empty the array A.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of two lines of input.
  - The first line of each test case contains one integer N — the number of elements in A.
  - The second line contains N integers $A_1, A_2, \ldots, A_N$, the elements of the array A.

Output Format

For each test case, output on a new line, the **minimum** number of boxes required to empty the array A.

Constraints

- $1 \leq T \leq 10^5$
- $1 \leq N \leq 2 \cdot 10^5$
- $1 \leq A_i \leq 10^{18}$
- The sum of N over all test cases won't exceed $2 \cdot 10^5$.

Sample 1

Input	Output
3 1 6 4 1 2 3 12 3 6 5 4	1 1 2

Test case 1: – Choose 6 from the array and delete it. – Add 6 to a new box.

A is empty now and we used one box.

Test case 2:

- Choose 12 from the array and delete it.
  - Add 12 to a new box. The XOR of the box becomes 12.
- Choose 3 from the array and delete it.
  - Since the XOR of the box $12 \geq 3$, we can add 3 to the box. The XOR of the box becomes $12 \oplus 3 = 15$.
- Choose 2 from the array and delete it.
  - Since the XOR of the box $15 \geq 2$, we can add 2 to the box. The XOR of the box becomes $15 \oplus 2 = 13$.
- Choose 1 from the array and delete it.
  - Since the XOR of the box $13 \geq 1$, we can add 1 to the box. The XOR of the box becomes $13 \oplus 1 = 12$.

A is empty now and we used one box.

Test case 3:

- Choose 5 from the array and delete it.

    - Add 5 to a new box. The XOR of the box becomes 5.
- Choose 4 from the array and delete it.
    - Since the XOR of the box $5 \geq 4$, we can add 4 to the box. The XOR of the box becomes $5 \oplus 4 = 1$.
- Choose 6 from the array and delete it.
    - Add 6 to a new box. The XOR of the second box is 6.

A is now empty and we used two boxes.

Problem M. NASA

Time limit	2500 ms
Code length Limit	50000 B
OS	Linux

Problem Link

I'ma need space, I'ma, I'ma need
You know I'm a star; space, I'ma need space
I'ma need space, I'ma, I'ma need space (N-A-S-A)

Given an array A of size N.
Find total number of pairs in the array (i, j) $(1 \leq i \leq j \leq N)$ such that:

- $A_i \oplus A_j$ is a palindrome (in **decimal** representation), where $\oplus$ denotes the bitwise xor operator.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of multiple lines of input.
  - The first line of each test case contains one integer N - the size of the array.
  - The next line contains N space-seperated integers as array elements.

Output Format

For each test case, output on a new line, the number of pairs satisfying the given condition.

Constraints

- $1 \leq T \leq 10^2$
- $1 \leq N \leq 10^5$
- $0 \leq A_i < 2^{15}$

- The sum of N over all test cases does not exceed $2 \cdot 10^5$.

Sample 1

Input	Output
2 4 13 27 12 26 3 2 2 2	8 6

Test case 1: The pairs which form palindrome are : - $13 \oplus 13 = 0$ - $13 \oplus 27 = 22$ - $13 \oplus 12 = 1$ - $27 \oplus 27 = 0$ - $27 \oplus 26 = 1$ - $12 \oplus 12 = 0$ - $12 \oplus 26 = 22$ - $26 \oplus 26 = 0$

Test case 2: All the pairs form palindrome.

Problem N. Dull Operation

Time limit	1000 ms
Code length Limit	50000 B
OS	Linux

Problem Link

Be afraid, be very afraid.

On Halloween, Chef is in a somber mood.
Chef has an **odd** integer N that he has to decode.
To do so, Chef would like to find a pair of integers x and y ($0 \leq x, y < 2^{30}$) such that:

$$(x \mid y) \cdot (x \oplus y) = N$$

Help Chef find **any** such pair!
It can be proved that a valid pair always exists.

Here, $|$ represents the bitwise OR operation, and $\oplus$ represents the bitwise XOR operation.

Input Format

- The first line of input will contain a single integer T, denoting the number of test cases.
- The first and only line of each test case contains a single **odd** integer N.

Output Format

For each test case, output on a new line two space-separated integers x and y such that $0 \leq x, y < 2^{30}$, and

$$(x \mid y) \cdot (x \oplus y) = N$$

If multiple solutions exist, you may print **any of them**.
It can be proved that a solution always exists under the given constraints.

Constraints

- $1 \leq T \leq 1000$
- $1 \leq N \leq 10^9$
- N is **odd**.

Sample 1

Input	Output
4 1 49 21 35	1 0 3 4 7 4 7 2

Test case 1: We have $N = 1$. Choose $x = 1$ and $y = 0$, which gives us $(x \mid y) = 1$ and $(x \oplus y) = 1$. $1 \cdot 1 = 1$, so this is a valid solution.

Test case 2: We have $N = 49$. Choose $x = 3$ and $y = 4$, which gives us $(x \mid y) = 7$ and $(x \oplus y) = 7$.
$7 \cdot 7 = 49$, so this is a valid solution.

Test case 3: Here, $N = 21$. Choose $x = 7$ and $y = 4$, which gives us $(x \mid y) = 7$ and $(x \oplus y) = 3$.
$7 \cdot 3 = 21$, so this is a valid solution.

Test case 4: Here, $N = 35$. Choose $x = 7$ and $y = 2$, which gives us $(x \mid y) = 7$ and $(x \oplus y) = 5$.
$7 \cdot 5 = 35$, so this is a valid solution.

Problem O. XORwice

Time limit 1000 ms
Mem limit 262144 kB

Problem Link

In order to celebrate Twice's 5th anniversary, Tzuyu and Sana decided to play a game.

Tzuyu gave Sana two integers a and b and a really important quest.

In order to complete the quest, Sana has to output the smallest possible value of $(a \oplus x)$ + $(b \oplus x)$ for any given x, where $\oplus$ denotes the bitwise XOR operation.

Input

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

The only line of each test case contains two integers a and b ($1 \le a, b \le 10^9$).

Output

For each testcase, output the smallest possible value of the given expression.

Sample 1

Input	Output
6 6 12 4 9 59 832 28 14 4925 2912 1 1	10 13 891 18 6237 0

Note

For the first test case Sana can choose $x = 4$ and the value will be $(6 \oplus 4)$ + $(12 \oplus 4)$ =

$2 + 8 = 10$. It can be shown that this is the smallest possible value.

Problem P. Roof Construction

Time limit 1000 ms
Mem limit 262144 kB
Problem Link

It has finally been decided to build a roof over the football field in School 179. Its construction will require placing n consecutive vertical pillars. Furthermore, the headmaster wants the heights of all the pillars to form a permutation p of integers from 0 to $n-1$, where p_i is the height of the i-th pillar from the left $(1 \le i \le n)$.

As the chief, you know that the cost of construction of consecutive pillars is equal to **the maximum value of the bitwise XOR** of heights of all pairs of adjacent pillars. In other words, the cost of construction is equal to $\max\limits_{1 \le i \le n-1} p_i \oplus p_{i+1}$, where $\oplus$ denotes the bitwise XOR operation.

Find any sequence of pillar heights p of length n with the smallest construction cost.

In this problem, a permutation is an array consisting of n distinct integers from 0 to $n-1$ in arbitrary order. For example, $[2, 3, 1, 0, 4]$ is a permutation, but $[1, 0, 1]$ is not a permutation (1 appears twice in the array) and $[1, 0, 3]$ is also not a permutation ($n = 3$, but 3 is in the array).

Input

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

The only line for each test case contains a single integer n ($2 \le n \le 2 \cdot 10^5$) — the number of pillars for the construction of the roof.

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

Output

For each test case print n integers $p_1, p_2, \ldots, p_n$ — the sequence of pillar heights with the smallest construction cost.

If there are multiple answers, print any of them.

Sample 1

Input	Output
4 2 3 5 10	0 1 2 0 1 3 2 1 0 4 4 6 3 2 0 8 9 1 7 5

Note

For $n = 2$ there are 2 sequences of pillar heights:

- $[0, 1]$ — cost of construction is $0 \oplus 1 = 1$.
- $[1, 0]$ — cost of construction is $1 \oplus 0 = 1$.

For $n = 3$ there are 6 sequences of pillar heights:

- $[0, 1, 2]$ — cost of construction is $\max(0 \oplus 1, 1 \oplus 2) = \max(1, 3) = 3$.
- $[0, 2, 1]$ — cost of construction is $\max(0 \oplus 2, 2 \oplus 1) = \max(2, 3) = 3$.
- $[1, 0, 2]$ — cost of construction is $\max(1 \oplus 0, 0 \oplus 2) = \max(1, 2) = 2$.
- $[1, 2, 0]$ — cost of construction is $\max(1 \oplus 2, 2 \oplus 0) = \max(3, 2) = 3$.
- $[2, 0, 1]$ — cost of construction is $\max(2 \oplus 0, 0 \oplus 1) = \max(2, 1) = 2$.
- $[2, 1, 0]$ — cost of construction is $\max(2 \oplus 1, 1 \oplus 0) = \max(3, 1) = 3$.

Problem Q. Mocha and Math

Time limit 1000 ms
Mem limit 262144 kB
Problem Link

Mocha is a young girl from high school. She has learned so much interesting knowledge from her teachers, especially her math teacher. Recently, Mocha is learning about binary system and very interested in bitwise operation.

This day, Mocha got a sequence a of length n. In each operation, she can select an arbitrary interval $[l, r]$ and for all values i $(0 \leq i \leq r - l)$, replace a_{l+i} with a_{l+i} $\&$ a_{r-i} at the same time, where $\&$ denotes the bitwise AND operation. This operation can be performed **any number of times**.

For example, if $n = 5$, the array is $[a_1, a_2, a_3, a_4, a_5]$, and Mocha selects the interval $[2, 5]$, then the new array is $[a_1, a_2 \,\&\, a_5, a_3 \,\&\, a_4, a_4 \,\&\, a_3, a_5 \,\&\, a_2]$.

Now Mocha wants to minimize the maximum value in the sequence. As her best friend, can you help her to get the answer?

Input

Each test contains multiple test cases.

The first line contains a single integer t $(1 \leq t \leq 100)$ — the number of test cases. Each test case consists of two lines.

The first line of each test case contains a single integer n $(1 \leq n \leq 100)$ — the length of the sequence.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ $(0 \leq a_i \leq 10^9)$.

Output

For each test case, print one integer — the minimal value of the maximum value in the sequence.

Sample 1

Input	Output
4 2 1 2 3 1 1 3 4 3 11 3 7 5 11 7 15 3 7	0 1 3 3

Note

In the first test case, Mocha can choose the interval $[1, 2]$, then the sequence becomes $[0, 0]$, where the first element is $1 \,\&\, 2$, and the second element is $2 \,\&\, 1$.

In the second test case, Mocha can choose the interval $[1, 3]$, then the sequence becomes $[1, 1, 1]$, where the first element is $1 \,\&\, 3$, the second element is $1 \,\&\, 1$, and the third element is $3 \,\&\, 1$.

Problem R. Absolute Maximization

Time limit 1000 ms
Mem limit 262144 kB
[Problem Link]

You are given an array a of length n. You can perform the following operation several (possibly, zero) times:

- Choose i, j, b: Swap the b-th digit in the binary representation of a_i and a_j.

Find the maximum possible value of $\max(a) - \min(a)$.

In a binary representation, bits are numbered from right (least significant) to left (most significant). Consider that there are an infinite number of leading zero bits at the beginning of any binary representation.

For example, swap the 0-th bit for $4 = 100_2$ and $3 = 11_2$ will result $101_2 = 5$ and $10_2 = 2$. Swap the 2-nd bit for $4 = 100_2$ and $3 = 11_2$ will result $000_2 = 0_2 = 0$ and $111_2 = 7$.

Here, $\max(a)$ denotes the maximum element of array a and $\min(a)$ denotes the minimum element of array a.

The binary representation of x is x written in base 2. For example, 9 and 6 written in base 2 are 1001 and 110, respectively.

Input

The first line contains a single integer t ($1 \leq t \leq 128$) — the number of testcases.

The first line of each test case contains a single integer n ($3 \leq n \leq 512$) — the length of array a.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($0 \leq a_i < 1024$) — the elements of array a.

It's guaranteed that the sum of n over all testcases does not exceed 512.

Output

For each testcase, print one integer — the maximum possible value of $\max(a) - \min(a)$.

Sample 1

Input	Output
4 3 1 0 1 4 5 5 5 5 5 1 2 3 4 5 7 20 85 100 41 76 49 36	1 0 7 125

Note

In the first example, it can be shown that we do not need to perform any operations — the maximum value of $\max(a) - \min(a)$ is $1 - 0 = 1$.

In the second example, no operation can change the array — the maximum value of $\max(a) - \min(a)$ is $5 - 5 = 0$.

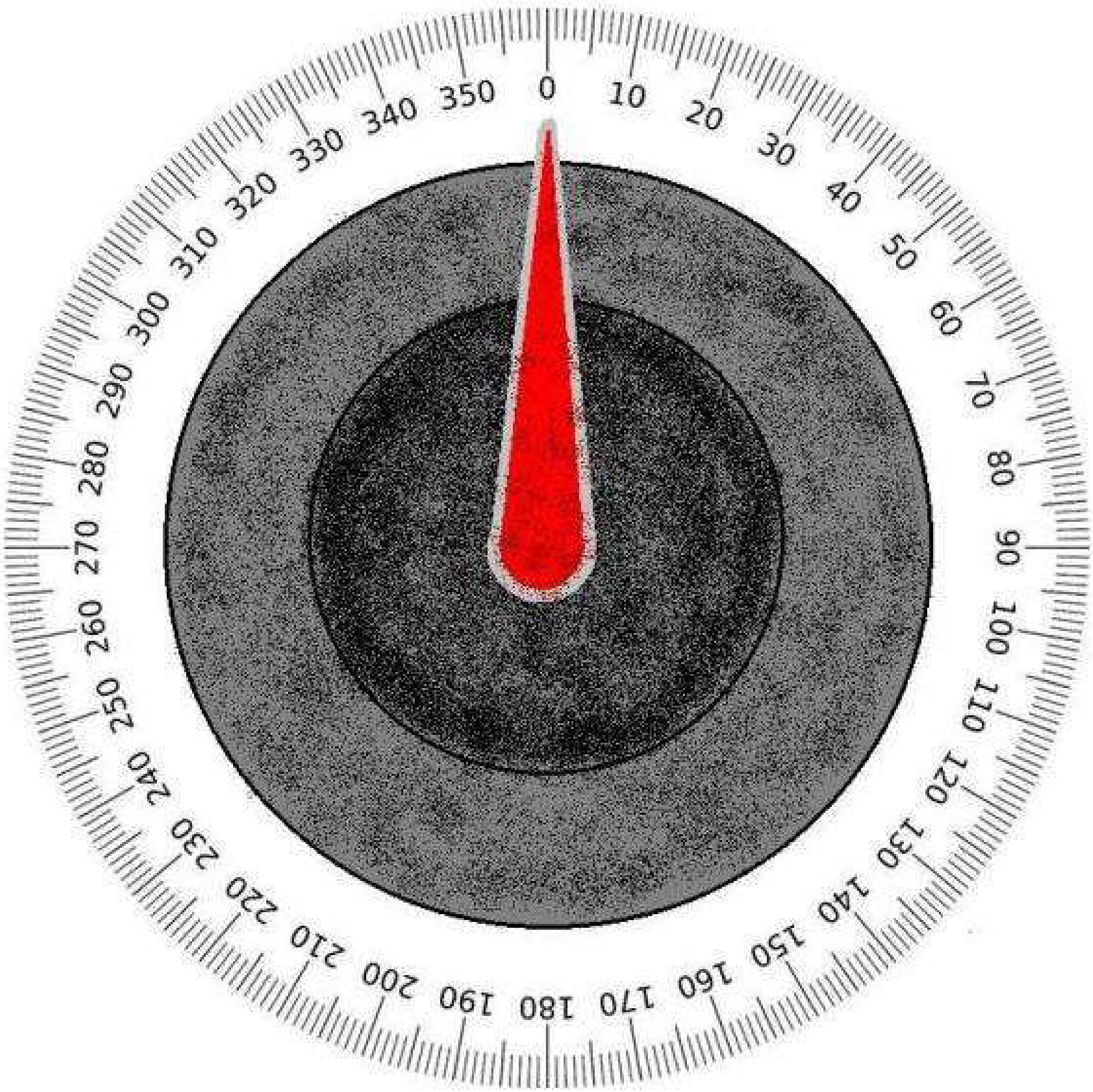
In the third example, initially $a = [1, 2, 3, 4, 5]$, we can perform one operation taking $i = 2, j = 5, b = 1$. The array now becomes $a = [1, 0, 3, 4, 7]$. It can be shown that any further operations do not lead to a better answer — therefore the answer is $\max(a) - \min(a) = 7 - 0 = 7$.

Problem S. Petr and a Combination Lock

Time limit 1000 ms
Mem limit 262144 kB

[Problem Link]

Petr has just bought a new car. He's just arrived at the most known Petersburg's petrol station to refuel it when he suddenly discovered that the petrol tank is secured with a combination lock! The lock has a scale of 360 degrees and a pointer which initially points at zero:



Petr called his car dealer, who instructed him to rotate the lock's wheel exactly n times. The i-th rotation should be a_i degrees, either clockwise or counterclockwise, and after all n rotations the pointer should again point at zero.

This confused Petr a little bit as he isn't sure which rotations should be done clockwise and which should be done counterclockwise. As there are many possible ways of rotating the lock, help him and find out whether there exists at least one, such that after all n rotations the pointer will point at zero again.

Input

The first line contains one integer n $(1 \leq n \leq 15)$ — the number of rotations.

Each of the following n lines contains one integer a_i $(1 \leq a_i \leq 180)$ — the angle of the i-th rotation in degrees.

Output

If it is possible to do all the rotations so that the pointer will point at zero after all of them are performed, print a single word "`YES`". Otherwise, print "`NO`". Petr will probably buy a new car in this case.

You can print each letter in any case (upper or lower).

Sample 1

Input	Output
3 10 20 30	YES

Sample 2

Input	Output
3 10 10 10	NO

Sample 3

Input	Output
3 120 120 120	YES

Note

In the first example, we can achieve our goal by applying the first and the second rotation clockwise, and performing the third rotation counterclockwise.

In the second example, it's impossible to perform the rotations in order to make the pointer point at zero in the end.

In the third example, Petr can do all three rotations clockwise. In this case, the whole wheel will be rotated by 360 degrees clockwise and the pointer will point at zero again.

Problem T. Fedor and New Game

Time limit	1000 ms
Mem limit	262144 kB
Input file	stdin
Output file	stdout

Problem Link

After you had helped George and Alex to move in the dorm, they went to help their friend Fedor play a new computer game «Call of Soldiers 3».

The game has $(m + 1)$ players and n types of soldiers in total. Players «Call of Soldiers 3» are numbered form 1 to $(m + 1)$. Types of soldiers are numbered from 0 to n - 1. Each player has an army. Army of the i-th player can be described by non-negative integer x_i. Consider binary representation of x_i: if the j-th bit of number x_i equal to one, then the army of the i-th player has soldiers of the j-th type.

Fedor is the $(m + 1)$-th player of the game. He assume that two players can become friends if their armies differ in at most k types of soldiers (in other words, binary representations of the corresponding numbers differ in at most k bits). Help Fedor and count how many players can become his friends.

Input

The first line contains three integers n, m, k $(1 \le k \le n \le 20;\ 1 \le m \le 1000)$.

The i-th of the next $(m + 1)$ lines contains a single integer x_i $(1 \le x_i \le 2^n$ - $1)$, that describes the i-th player's army. We remind you that Fedor is the $(m + 1)$-th player.

Output

Print a single integer — the number of Fedor's potential friends.

Sample 1

Input	Output
7 3 1 8 5 111 17	0

Sample 2

Input	Output
3 3 3 1 2 3 4	3