

OR Preference

Input file: **standard input**
Output file: **standard output**
Time limit: 3 seconds
Memory limit: 768 megabytes

Note that the memory constraints are special.

You are given an integer N and an integer sequence $A = (A_1, A_2, \dots, A_N)$.

You repeatedly perform operations on the sequence until its length becomes 1. At each step, you choose two adjacent elements in the current sequence and merge them into a single element. After each merge, the resulting sequence is reindexed in the natural order. The following two types of operations are available.

- Operation AND: replace A_i and A_{i+1} with $(A_i \& A_{i+1})$.
- Operation OR: replace A_i and A_{i+1} with $(A_i | A_{i+1})$.

Here, the operators $\&$ and $|$ denote the bitwise AND and bitwise OR operations, respectively.

Since each operation reduces the length of the sequence by 1, exactly $(N - 1)$ operations are performed in total.

Among all possible sequences of operations such that the final remaining element is equal to 0, find the maximum possible number of times Operation OR is performed.

Input

The input consists of multiple test cases.

```
T
testcase 1
testcase 2
⋮
testcase T
```

Each test case is given in the following format:

```
N
A1, A2, ..., AN
```

- $1 \leq T \leq 2^{12}$
- $2 \leq N \leq 2^{13}$
- $0 \leq A_i < 2^{13}$
- The sum of N over all test cases does not exceed 2^{13} .
- All input values are integers.

Output

Print the maximum number of times Operation OR can be performed among all operation sequences that satisfy the condition that the final remaining element is equal to 0. If no sequence of operations satisfies the condition, print -1 .

Example

standard input	standard output
4	3
6	0
3 0 1 4 1 5	5
2	6
0 1	
8	
2 0 1 6 1 2 0 3	
10	
1 7 6 7 5 7 5 3 2 7	

Note

This input contains 4 test cases.

For the first test case, we can perform the second type of operation 3 times as follows:

- Initially, $A = (3, 0, 1, 4, 1, 5)$.
- Apply the OR operation to (A_3, A_4) . After the operation, $A = (3, 0, 5, 1, 5)$.
- Apply the AND operation to (A_1, A_2) . After the operation, $A = (0, 5, 1, 5)$.
- Apply the OR operation to (A_3, A_4) . After the operation, $A = (0, 5, 5)$.
- Apply the OR operation to (A_2, A_3) . After the operation, $A = (0, 5)$.
- Apply the AND operation to (A_1, A_2) . After the operation, $A = (0)$.

Since the final state satisfies $A_1 = 0$, the condition is met.

It is impossible to satisfy the condition with 4 or more OR operations, so the answer is 3.