Kulyash is given an integer *N*. His task is to break *N* into some number of (integer) powers of 2.

To achieve this, he can perform the following operation several times (possibly, zero):

- Choose an integer *X* which he already has, and break *X* into 2 integer parts (*Y* and *Z*) such that *X*=*Y*+*Z*.

Find the minimum number of operations required by Kulyash to accomplish his task.

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 of input.
    - The first and only line of each test case contains an integer *N* — the original integer with Kulyash.

Output Format

For each test case, output on a new line the minimum number of operations required by Kulyash to break *N* into powers of 2.

Sample Input

2

3

4

Sample Output

1

0

```
import java.util.*;
import java.lang.*;
import java.io.*;
```

```java
class Main {
	static boolean isPowerOfTwo(int x) {
		return x != 0 && ((x & (x - 1)) == 0);
	}

	public static void main(String[] args) throws java.lang.Exception {
		Scanner sc = new Scanner(System.in);
		int t = sc.nextInt();
		while (t-- > 0) {
			int n = sc.nextInt();
			int count = 0;
			while (n > 0) {
				count += n & 1;
				n >>= 1;
			}
			System.out.println(count - 1);
		}
	}
}
```