F. And It's Non-Zero

time limit per test: 2 seconds

memory limit per test: 256 megabytes

You are given an array consisting of all integers from $[l, r]$ inclusive. For example, if $l = 2$ and $r = 5$, the array would be $[2, 3, 4, 5]$. What's the minimum number of elements you can delete to make the bitwise AND of the array non-zero?

A *bitwise AND* is a binary operation that takes two equal-length binary representations and performs the *AND* operation on each pair of the corresponding bits.

Input

The first line contains one integer t ($1 \le t \le 10^4$) — the number of test cases. Then t cases follow.

The first line of each test case contains two integers l and r ($1 \le l \le r \le 2 \cdot 10^5$) — the description of the array.

Output

For each test case, output a single integer — the answer to the problem.

Example

input Copy

```
5
1 2
2 8
4 5
1 5
100000 200000
```

output Copy

```
1
3
0
2
31072
```

Note

In the first test case, the array is $[1, 2]$. Currently, the *bitwise AND* is 0, as $1 \ \& \ 2 = 0$. However, after deleting 1 (or 2), the array becomes $[2]$ (or $[1]$), and the *bitwise AND* becomes 2 (or 1). This can be proven to be the optimal, so the answer is 1.

In the second test case, the array is $[2, 3, 4, 5, 6, 7, 8]$. Currently, the *bitwise AND* is 0. However, after deleting 4, 5, and 8, the array becomes $[2, 3, 6, 7]$, and the *bitwise AND* becomes 2. This can be proven to be the optimal, so the answer is 3. Note that there may be other ways to delete 3 elements.

ICPC Assiut University Training - Juniors Phase 1 Sheets-2022

Public

Participant

→ Group Contests

- Juniors Phase 1 Practice #5 (Bitmask, Bitset, Bits)
- Juniors Phase 1 Practice #4 (Binary search , Two pointers)
- Juniors Phase 1 Practice #3 (STL 2)
- Juniors Phase 1 Practice #2 (STL 1)
- Juniors Phase 1 Practice #1 (Prefix sum , Frequency Array)

Juniors Phase 1 Practice #5 (Bitmask, Bitset, Bits)

Finished

Practice

→ About Time Scaling

This contest uses time limits scaling policy (depending on a programming language). The system automatically adjusts time limits by the following multipliers for some languages. Despite scaling (adjustment), the time limit cannot be more than 30 seconds. Read the details by the link.

→ Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

Start virtual contest

→ Submit?

Language: GNU G++20 13.2 (64 bit, win

Choose file: Choose File No file chosen

Submit

→ Last submissions

Submission	Time	Verdict
231449332	Nov/05/2023 13:49	Accepted
231449074	Nov/05/2023 13:46	Accepted
231448779	Nov/05/2023 13:43	Accepted