

Sofdor Ali, the mad scientist, is a well known figure for his great inventions. He sent me a telegram saying, ‘urgent need, write the code and send me the output!’ And there was a pseudo code in the telegram. Finally, I managed to write it in C++ (my apology to non C programmers). Here is the code:

```
#include <cstdio>
#include <algorithm>
using namespace std;

typedef long long i64;

const i64 MOD = (1 << 30);

void solve( i64 n, i64 m ) {
    i64 res1 = 0, res2 = 0, k, i;

    for( k = 0; k <= m; k++ ) {
        for( i = 0; ; i++ )
            if( ( n ^ i ) == k ) // n ^ i means n xor i
                break;
        res1 = max( res1, i );
    }
    for( i = 0; i <= res1; i++ ) {
        if( ( n ^ i ) > m ) {
            res2 += ( n ^ i );
            res2 %= MOD;
        }
    }
    printf("%lld %lld\n", res1, res2);
}

int main() {
    int T, t;
    i64 n, m;

    scanf("%d", &T);
    for( t = 1; t <= T; t++ ) {
        scanf("%lld %lld", &n, &m);
        printf("Case %d: ", t);
        solve( n, m );
    }
    return 0;
}
```

But soon, I found that for inputs, as specified in the telegram, the code takes huge time to compute the result. And according to the structure of the code, it’s clear that for some cases it may take years to compute the output.

So, I am asking you to help me optimizing the code such that it can generate the output in reasonable amount of time.

Input

Input starts with an integer T (≤ 10000), denoting the number of test cases.

Each case starts with a line containing two integers n, m ($0 \leq n, m \leq 2^{60}$).

Output

For each case, print the output generated by the code.

Sample Input

```
3
0 0
5 3
10 13
```

Sample Output

```
Case 1: 0 0
Case 2: 7 22
Case 3: 15 29
```