

Revisi Pre Test PBO C

Brendan Timothy Mannuel

5025221177

BT69 - BitPlay69

You are given 2 integers **N** and **M**.

Print the smallest **K**, such that $N \oplus K > M$. Here, \oplus is the Bitwise XOR Operator.

Pada soal ini kita diperlukan untuk mencari angka K terkecil yang dapat dibentuk dari operasi N XOR K tetapi masih harus lebih besar dari M

Berdasarkan soal diberikan batasan sebanyak $1 \leq N, M \leq 10^{17}$. Kita dapat menggunakan operasi bitwise sehingga batasan terkecil adalah 2^0 dan batasan terbesar adalah 2^{64} .

```
Input:
4
3 5
3 2
69 696
696 96

Output:
4
0
640
0
```

Kita akan mencoba mengulas berdasarkan testcase pertama yaitu 3 sebagai n dan 5 sebagai m dengan jawaban 4, karena di soal diperlukan yang lebih besar ($hasil > m$), maka 5 akan kita tambah 1 menjadi 6.

Angka	Basis 2		
3	-	1	1
6	1	1	0
4	1	0	0

Berdasarkan table diatas maka bisa dilihat bahwa angka 3 memiliki 2^2 kemungkinan sedangkan angka 6 dan 4 memiliki 2^8 kemungkinan. Setelah kita lihat pada table, kita mulai dari table paling kanan, kita bisa lihat bahwa pada bit angka 3 dan 6 yaitu 1 dan 0, bit paling kiri dari 4 adalah 0, sehingga kita perlu cara untuk membuat angka 0 dan menyimpannya, selain itu juga kita perlu melakukan iterasi untuk semua bit n dan m sehingga dapat menemukan jawabannya.

Jawaban

```
int cases;  
scanf ("%d", &cases);
```

Kita akan menginput berapa banyak iterasi yang diminta soal

```
while (cases--){  
    ll n, m, res = 0, b = 1;  
  
    scanf ("%lld %lld", &n, &m);  
    m++;
```

Kita akan mengiterasi sebanyak “cases” sekaligus mendeklarasikan n dan m, serta res sebagai hasil akhir dan, b sebagai suatu variable yang akan digunakan sebagai tempat menaruh bit yang telah di proses (kenapa b dideklarasikan menjadi 1 karena 2^0 adalah 1 dan karena berapapun hasilnya nanti bit paling kanan akan dimulai angka 1), semua variable akan dideklarasikan sebagai long long karena angka terbesar nya adalah 10^{17} .

Angka	b		
Biner	2^2	2^1	2^0
1	-	-	1

Kita akan menginput angka n dan m, dan sekaligus menambah m satu angka karena disoal meminta agar hasil $> m$.

```
while(n || m){  
    ll N = n & 1, M = m & 1;  
    if (!N && M) res |= b;  
    else if (N && !M) res = 0;  
  
    b <<= 1; n >>= 1; m >>= 1;  
}
```

Dengan menggunakan contoh testcase 1 dimana $n = 3$ dan $m = 5 + 1 = 6$, sehingga terbentuk table berikut.

Angka	Basis 2		
3	-	1	1
6	1	1	0

Kita akan menyimpan bit paling kiri dari n dan m dengan melakukan operasi AND pada bit paling kiri dan menyimpannya pada variable N dan M. sehingga tersimpan N = 1 dan M = 0. Pada iterasi ini akan memasuki kondisi kedua dimana N = TRUE karena 1 dan M = False karena 0, sehingga res = 0 agar hasil menjadi paling kecil, tentunya kita harus menyimpan nya pada variable b, kita akan melakukan operasi shift left sehingga table terupdate sebagai berikut.

Angka	b		
Biner	2^2	2^1	2^0
2	-	1	0

Kemudian kita akan melakukan operasi shift right pada n dan m yang mengubah banyak kemungkinan yang pada awalnya $m = 2^3$ dan $n = 2^2$ menjadi $m = 2^2$ dan $n = 2^1$ karena pada intinya shift right membagi angka dengan 2, sehingga table menjadi.

Angka	Basis 2		
1	-	-	1
3	-	1	1

Pada loop while, kondisi masih terpenuhi karena n dan m dua duanya tidak 0, sehingga kita akan mengupdate N = 1 dan M = 1, karena tidak ada case N dan M sama sama 1 maka kita hanya kan melakukan serangkaian operasi shift pada variable b, n, dan m, sehingga table terupdate menjadi.

Angka	b		
Biner	2^2	2^1	2^0
4	1	0	0
Angka	Basis 2		
0	-	-	-
1	-	-	1

Pada loop while, kondisi masih terpenuhi karena hanya n yang berangka 0, sehingga kita akan mengupdate N = 0 dan M = 1, pada case if pertama , syarat terpenuhi sehingga kita kan melakukan operasi OR pada res dan b sekaligus mengupdate res, sehingga res = 4. Setelah itu kita akan Kembali melakukan serangkaian operasi shift pada variable b, n, dan m.

Angka	b			
Biner	2^3	2^2	2^1	2^0
4	1	0	0	0

Angka	Basis 2		
0	-	-	-
0	-	-	-

Pada loop while, kondisi sudah tidak terpenuhi karena kedua n dan m bernilai 0, sehingga kita akan mengoutput hasil yaitu res yang berangka 4 sebagai jawaban akhir.

Kesimpulan

Pada soal ini kita melakukan pendekatan secara bitwise, kita memanipulasi banyaknya kemungkinan menjadi hanya 64 (Worst Case) kemungkinan dari serangkaian angka 1 dan 0, kita dapat menggunakan operasi shift right dan shift left untuk menyimpan hasil dan juga membagi angka, kita juga menggunakan operasi OR dan juga AND. Source Code lengkap dibawah:

```

1  #include <stdio.h>
2  typedef long long ll;
3
4  int main(){
5
6      int cases;
7      scanf ("%d", &cases);
8
9      while (cases--){
10         ll n, m, res = 0, b = 1;
11
12         scanf ("%lld %lld", &n, &m);
13         m++;
14
15         while(n || m){
16             ll N = n & 1, M = m & 1;
17             if (!N && M) res |= b;
18             else if (N && !M) res = 0;
19
20             b <<= 1; n >>= 1; m >>= 1;
21         }
22
23         printf ("%lld\n", res);
24     }
25     return 0;
26

```