```
foobar:~/queue-to-do cihan.goksu.88$ cat readme.txt
```

Queue To Do
==========

You're almost ready to make your move to destroy the LAMBCHOP doomsday device, but the security checkpoints that guard the underlying systems of the LAMBCHOP are going to be a problem. You were able to take one down without tripping any alarms, which is great! Except that as Commander Lambda's assistant, you've learned that the checkpoints are about to come under automated review, which means that your sabotage will be discovered and your cover blown -- unless you can trick the automated review system.

To trick the system, you'll need to write a program to return the same security checksum that the bunny trainers would have after they would have checked all the workers through. Fortunately, Commander Lambda's desire for efficiency won't allow for hours-long lines, so the trainers at the checkpoint have found ways to quicken the pass-through rate. Instead of checking each and every worker coming through, the bunny trainers instead go over everyone in line while noting their worker IDs, then allow the line to fill back up. Once they've done that they go over the line again, this time leaving off the last worker. They continue doing this, leaving off one more worker from the line each time but recording the worker IDs of those they do check, until they skip the entire line, at which point they XOR the IDs of all the workers they noted into a checksum and then take off for lunch. Fortunately, the workers' orderly nature causes them to always line up in numerical order without any gaps.

For example, if the first worker in line has ID 0 and the security checkpoint line holds three workers, the process would look like this:
0 1 2 /
3 4 / 5
6 / 7 8
where the trainers' XOR (^) checksum is 0^1^2^3^4^6 == 2.

Likewise, if the first worker has ID 17 and the checkpoint holds four workers, the process would look like:
17 18 19 20 /
21 22 23 / 24
25 26 / 27 28
29 / 30 31 32
which produces the checksum 17^18^19^20^21^22^23^25^26^29 == 14.

All worker IDs (including the first worker) are between 0 and

2000000000 inclusive, and the checkpoint line will always be at least 1 worker long.

With this information, write a function solution(start, length) that will cover for the missing security checkpoint by outputting the same checksum the trainers would normally submit before lunch. You have just enough time to find out the ID of the first worker to be checked (start) and the length of the line (length) before the automatic review occurs, so your program must generate the proper checksum with just those two values.

Languages
=========

To provide a Java solution, edit Solution.java
To provide a Python solution, edit solution.py

Test cases
==========
Your code should pass the following test cases.
Note that it may also be run against hidden test cases not shown here.

-- Java cases --
Input:
Solution.solution(0, 3)
Output:
    2

Input:
Solution.solution(17, 4)
Output:
    14

-- Python cases --
Input:
solution.solution(0, 3)
Output:
    2

Input:
solution.solution(17, 4)
Output:
    14

Use verify [file] to test your solution and see how it does. When you

are finished editing your code, use submit [file] to submit your answer. If your solution passes the test cases, it will be removed from your home folder.