Digital Pakistan Speed Programing Competition Online Qualifier Round

Instructions

- Do not open the booklet unless you are explicitly told to do so. You can only read these instructions below.
- If you have any question regarding the problems, seek a clarification from the judges using DOMJudge.
- Before submitting a run, make sure that it is executable via command line. For Java, it must be executable via "javac" and for GNU C++ via "g++". Java programmers need to remove any "package" statements and source code's file name must be the same as of main class. C++ programmers need to remove any getch() / system("pause") like statements.
- Do not attach input files while submitting a run, only submit/attach source code files, i.e., *.java or *.cpp or *.py.
- Language supported: C/C++, Java and Python3
- Source code file name should not contain white space or special characters.
- You must take input from Console i.e.: Standard Input Stream (stdin in C, cin in C++, System.in in Java, stdin in Python)
- You must print your output to Console i.e.: Standard Output Stream (stdout in C, cout in C++, System.out in Java)
- Please, don't create/open any file for input or output.
- Please strictly meet the output format requirements as described in problem statements, because your program will be auto judged by computer. Your output will be compared with judge's output byte-by-byte and not tolerate even a difference of single byte. So, be aware! **Pay special attention to spaces, commas, dots, newlines, decimal places, case sensitivity etc.**
- All your programs must meet the time constraint specified.
- The decision of judges will be absolutely final.

Problem 08: Garebi

Time limit: 1 seconds

Tired after a long stretch of programming practice, you decide to take a break and head to SP Hotel for some nihari. You order n $(n > 1)$ naans and p $(p \neq n)$ plates of nihari. You can finish one plate of nihari with one naan. If any naan or plates are left over, SP Hotel's famous *garebi* game begins.

If you are out of naans but still have unfinished plates, SP Hotel will additionally serve you half the number (rounded down) of naans from the previous round. If you are out of plates but still have naans, SP Hotel will additionally serve you half the number (rounded down) of plates from the previous round. If you run out of both together, you win the game and walk away without paying. The rounds continue like this until you win the game or you are unable to win. In the latter case, you have to join SP Hotel's growing kitchen staff for the rest of the month and miss the upcoming speed programming contest.

In a sample garebi game, you order $n = 7$ naans and $p = 11$ plates. This is represented as the state, $(7, 11)$. After eating, you are left with $n = 0$ naans and $p = 4$ plates (you eat 7 naans and 7 plates). The new state is $(0, 4)$. Round 1 is denoted as $(7, 11) \rightarrow (0, 4)$. As you are out of naan, you are served $7 / 2 = 3$ (rounded down) naans. The new state is $(3, 4)$, and after eating, the state is $(0, 1)$. Round 2 is $(3, 4) \rightarrow (0, 1)$. Round 3 is then $(1, 1) \rightarrow (0, 0)$. You win in 3 rounds!

Below are some other sample games:
Example 2: Round 1: $(11, 7) \rightarrow (4, 0)$. Round 2: $(4, 3) \rightarrow (1, 0)$. Round 3: $(1, 1) \rightarrow (1, 0)$. You win in 3 rounds!
Example 3: Round 1: $(4, 5) \rightarrow (0, 1)$. Round 2: $(2, 1) \rightarrow (1, 0)$. Round 3: $(1, 0) \rightarrow (1, 0)$. Round 4: $(1, 0) \rightarrow (1, 0)$. The rounds never end. You cannot win and head to the kitchen.
Example 4: Round 1: $(3, 4) \rightarrow (0, 1)$. Round 2: $(1, 1) \rightarrow (0, 0)$. You win in 2 rounds!
Example 5: Round 1: $(3, 2) \rightarrow (1, 0)$. Round 2: $(1, 1) \rightarrow (0, 0)$. You win in 2 rounds!

Input
The single line of input contains 2 space-separated integers, $\boldsymbol{n}$ $(2 \leq \boldsymbol{n} \leq 10^{18})$ and $\boldsymbol{d}$ $(\boldsymbol{d} \in \{-1, 1\})$, the number of naans that you order and the direction for $\boldsymbol{p}$: if $\boldsymbol{d} = 1$, then $\boldsymbol{p} > \boldsymbol{n}$; otherwise $\boldsymbol{p} < \boldsymbol{n}$.

Output
Output a single integer, the value of $\boldsymbol{p}$, such that you win in 2 rounds for the given $\boldsymbol{n}$ and $\boldsymbol{d}$. Output -1 if no such $\boldsymbol{p}$ exists.

Sample Input	Sample Output
3 1	4
3 -1	2
5 -1	-1
2 1	3

Note
The first two test cases are explained as sample games 4 and 5 above.
In the third test case, $(n, d) = (5, -1)$. No value of p less than 5 leads to victory in 2 rounds.
In the fourth test case, the rounds are $(2, 3) \rightarrow (0, 1)$ and then $(1, 1) \rightarrow (0, 0)$.