Java

**Submit a solution for B**

|  |  |
| --- | --- |
| **Full score:** | 100 |
| **Input file name:** | input.txt |
| **Output file name:** | output.txt |
| **Time limit:** | 1 s |
| **Real time limit:** | 1 s |
| **Memory limit:** | 64M |

**Evolution**

Під час досліджень, присвячених появі життя на планеті Олімпія, вченими було зроблено декілька сенсаційних відкриттів: 1. Усі живі організми планети походять від однієї бактерії Bitozoria Programulis. 2. Еволюція проходила крок за кроком (за припущенням вчених – під час змін клімату на планеті). 3. На кожному кроці еволюції з кожного виду утворювалися рівно два підвиди, а попередній вид зникав. 4. Якщо вважати появу бактерії Bitozoria Programulis першим кроком еволюції, то нині існуючі живі організми знаходяться на N-му кроці. Щоб не вигадувати назви під час досліджень, вчені пронумерували всі види організмів, що будь-коли існували на планеті. Для цього вони намалювали дерево еволюції із коренем Bitozoria Programulis, яка отримала номер 1. Далі нумерували види кожного кроку еволюції зліва направо. Таким чином безпосередні підвиди Bitozoria Programulis отримали номери 2 та 3. Наступними були занумеровані види третього кроку еволюції – підвиди виду 2 отримали номери 4 та 5, а виду 3 – номери 6 та 7, і т.д. Завдання Напишіть програму EVO, яка за номерами двох видів обчислить номер виду їх найближчого спільного предка у дереві еволюції.

Input format

Перший рядок вхідного файлу EVO.DAT містить ціле число N (1≤N≤100) – кількість етапів еволюції, що відбулися на планеті Олімпія до теперішнього часу. Другий та третій рядки файлу містять по одному натуральному числу, що представляють номери видів, для яких потрібно знайти номер їх найближчого спільного предка.

Output format

Єдиний рядок вихідного файлу EVO.SOL має містити натуральне число – номер найближчого предка для двох видів.

Examples

|  |  |
| --- | --- |
| Input in input.txt | Output in output.txt |
| 4  15  12 | 3 |
| 18  233016  233008 | 14563 |

import java.util.\*;  
import java.io.\*;  
import java.math.BigInteger;;  
public class Main{  
public static void main(String[] argv) throws IOException{  
new Main().run();  
}  
PrintWriter pw;  
Scanner sc;  
public void run() throws IOException{  
sc = new Scanner(new File("input.txt"));  
pw = new PrintWriter(new File("output.txt"));  
int n=sc.nextInt();  
BigInteger a=sc.nextBigInteger();  
BigInteger b=sc.nextBigInteger();  
while (!a.equals(b)){  
        if (a.compareTo(b)==1)a=a.divide(BigInteger.valueOf(2));  
        else if (a.compareTo(b)==-1)b=b.divide(BigInteger.valueOf(2));  
}  
pw.println(a);  
pw.close();  
}  
}