Magic Fractions (Programming)
A Magic Fraction for N is one that has the following properties:
It is a proper fraction (The value is < 1)
It cannot be reduced further (The GCD of the numerator and the denominator is 1)
The product of the numerator and the denominator is factorial of N (a/b is the fraction, then a*b = N!)
Examples of Magic Fractions are:
1/2 [gcd(1,2) = 1 and 1*2=2!]
2/3 [gcd(2,3) = 1 and 2*3=3!]
3/8
[gcd(3,8) = 1 and 3*8=4!]
2/12 is not a magic fraction, as even though 2*12=4!, gcd(2,12) != 1
And Magic fractions for number 3 are: 2/3 and 1/6 (since both of them satisfy the above criteria, are of the form a/b where a*b = 3!)
Now given a number N, you need to print the number of magic fractions that exists, for all numbers between 1 and N (include magic fractions for N, too).
Note:
The number N will be in the range [1, 500]. (1 and 500 inclusive)
You'll have to read the input from STDIN, and print the output to STDOUT

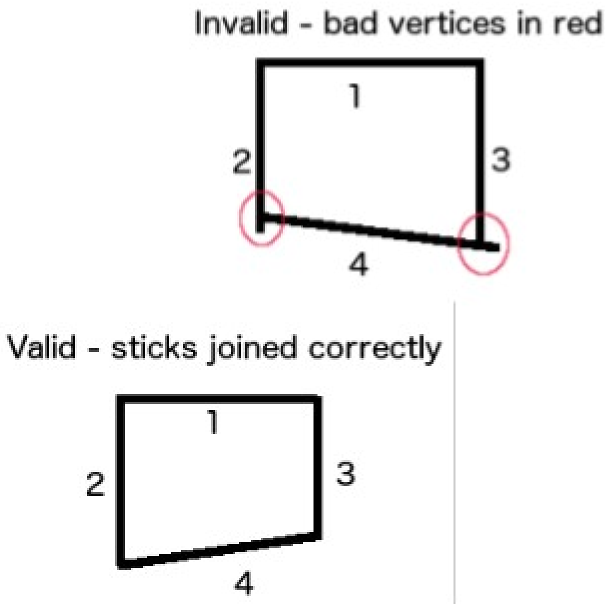
Examples:
1)
Input: 1
Output: 0
Explanation: There is no fraction < 1 whose numerator * denominator = 1!

2)
Input: 3
Output: 3
Explanation: 0 for 1 + 1 for 2 [1/2] + 2 for 3 [1/6, 2/3]

3)
Input: 5
Output: 9

Choosy Guests (Programming)
There is a party in which not every guest is interested in talking to every other guest. Given who is comfortable talking to whom, the host needs an arrangement so that the maximum number of guests are kept engaged.
So given an input matrix which has Y in the cell (i, j) if person i is comfortable talking to person j, our task is to figure out what are the maximum number of people that can be kept engaged amongst themselves. Note that:
A guest can talk only to one guest at a time, effectively keeping both of them engaged
We need to decide only one such guest to guest mapping, and that will remain throughout the party. i.e., the guest dont need to switch the people they are talking to mid-way in the party
All the cells in the matrix have values Y or N
For simplicity, assume that if a Guest i is interested in talking to guest j, then the guest j is interested in talking to guest i too.
The first line in input gives the number of guests G
The next G lines represent the cells in Matrix with values as described above
Note: The number of guests is a positive number <= 50
Example:
(Explanations are based on 1 based index in matrix)
Input:
4
NYNN
YNYY
NYNN
NYNN
Output:
2
Explanation:
All the chatter can happen only via Guest 2, but he can talk to only one other person
Input:
5
NYNN
YNNN
NNNY
NNYN
YNNN
Y
N
N
N
N
Output:
4
Explanation:
The best case is when the pairing is [(1,2), (3,4)] or [(1,5), (3,4)].

Sticky Problem (Programming)
As a teacher for a 1st grade class, you plan an activity for the kids to introduce them to what a rectangle is.
In an attempt to do that, you decide to cut 4 equal sized sticks, and stick them together as a rectangle. After cutting 3 sticks and pasting them at right angles to each other (one base and 2 sides), you realize that there was some error in cutting the first 3 sticks. What you do know though, is the range within which each of the sticks length falls. Lets say the ith stick falls in the range (li, ui). Now, when the 4th stick - you want to be sure of the range in which the 4th stick should fall for it to be stuck on the remaining side, so that no part of it comes out of the rectangle.



In other words, given l1, u1, l2, u2, l3, u3 and knowing that the sticks are stuck at right angles, determine the minimum and maximum possible length for the 4th stick to be cut, assuming we are talking of scenarios when none of the edges go beyond the vertices thus formed.
Input will be 6 integers, 2 in each line representing the the range for base (l i, u i) and the 2 sides respectively.
The output should be 2 comma separated integers, representing the rounded off range for the 4th side (lower number first, then higher number)
Note: both the base and the sides will be a positive number less than 500 units
Example:
1)
Input:
4,4

3,3
3,3
Output:
4,4
Explanation:
The base (4) and the 2 sides (3) have no error, so the new side also will not have any error, and will be equal to the base.
2)
Input:
10,12
4,8
3,6
Output:
10, 13
Explanation:
The best case is when the base is of size 10, and the sides are equal (possible in the range (4,6)). Worst is when the base is 12, and the difference in 2 sides is 5 (the sides have size 8 and 3)