Let d(n) be defined as the sum of proper divisors of n (numbers less than n which divide evenly into n).
If d(a) = b and d(b) = a, where a ≠ b, then a and b are an amicable pair and each of a and b are called amicable numbers.

For example, the proper divisors of 220 are 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 and 110; therefore d(220) = 284.

The proper divisors of 284 are 1, 2, 4, 71 and 142; so d(284) = 220.

Therefore, amicable numbers are a pair of numbers, each of which is the sum of the factors of the other.

Write a multithreaded client-server application whereby a client must accept two numbers as input from the user and then pass them to a server application. The server must then determine if the two numbers are amicable or not and communicate this to the client. The client application must then display a message to the end user indicating if the original input represented an amicable pair. Two sample runs of the program are illustrated below.

The responsibilities of the **client** are as follows:

1. Accept two (numeric) values from the user.
2. These numbers must be passed to the server as two **int** values.
3. Receive a **boolean** value from the server.
4. If the **boolean** value returned from server is **false**, the client must display a message indicating that the original input did not represent an amicable pair.
5. If the **boolean** value returned from server is **true**, the client must display a message indicating that the original input represented an amicable pair.

The responsibilities of the **server** are as follows:

1. Accept two (numeric) values from the client.
2. Determine of the two numbers represent an amicable pair.
3. Return a **boolean** value to the client.

4. The server must return a value of **false** if the numbers passed from the client application do not represent an amicable pair.
5. The server must return a value of **true** if the numbers passed from the client application represent an amicable pair.

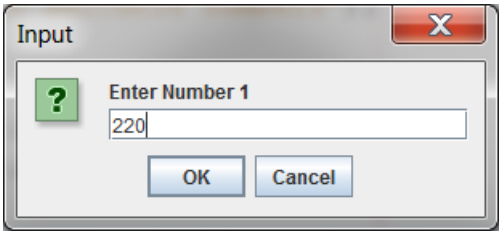
Note 1: *The server must be multi-threaded.*

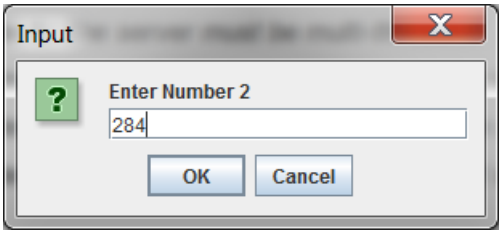
Note 2: *All necessary exceptions must be caught and declared.*

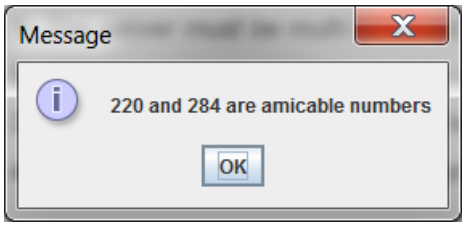
Note 3: *There is no input/output from/to the user in the server.*

Note 4: *The server must continually accept input from the same client.*

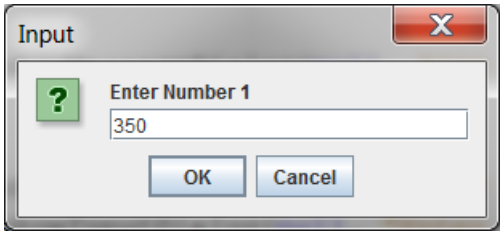
Sample Run 1

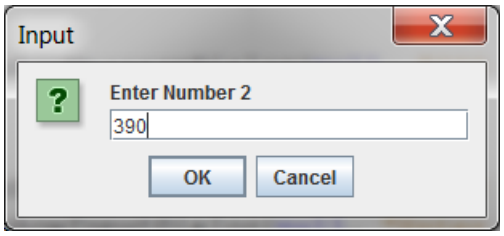


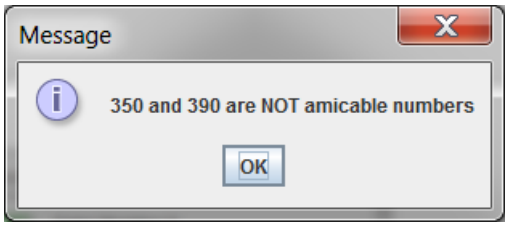




Sample Run 2


Input
Enter Number 1
350
OK
Cancel


Input
Enter Number 2
390
OK
Cancel


Message
350 and 390 are NOT amicable numbers
OK