Call two arms *equally strong* if the heaviest weights they each are able to lift are equal.

Call two people *equally strong* if their strongest arms are equally strong (the strongest arm can be both the right and the left), and so are their weakest arms.

Given your and your friend's arms' lifting capabilities find out if you two are equally strong.

Example

- For `yourLeft = 10`, `yourRight = 15`, `friendsLeft = 15`, and `friendsRight = 10`, the output should be
  `areEquallyStrong(yourLeft, yourRight, friendsLeft, friendsRight) = true`;
- For `yourLeft = 15`, `yourRight = 10`, `friendsLeft = 15`, and `friendsRight = 10`, the output should be
  `areEquallyStrong(yourLeft, yourRight, friendsLeft, friendsRight) = true`;
- For `yourLeft = 15`, `yourRight = 10`, `friendsLeft = 15`, and `friendsRight = 9`, the output should be
  `areEquallyStrong(yourLeft, yourRight, friendsLeft, friendsRight) = false`.

Input/Output

- **[execution time limit] 0.5 seconds (cpp)**

- **[input] integer yourLeft**

  A non-negative integer representing the heaviest weight you can lift with your left arm.

  Guaranteed constraints:
  `0 ≤ yourLeft ≤ 20`.

- **[input] integer yourRight**

  A non-negative integer representing the heaviest weight you can lift with your right arm.

  Guaranteed constraints:
  `0 ≤ yourRight ≤ 20`.

- **[input] integer friendsLeft**

  A non-negative integer representing the heaviest weight your friend can lift with his or her left arm.

  Guaranteed constraints:
  `0 ≤ friendsLeft ≤ 20`.

- **[input] integer friendsRight**

A non-negative integer representing the heaviest weight your friend can lift with his or her right arm.

Guaranteed constraints:
`0 ≤ friendsRight ≤ 20`.

- **[output] boolean**
    - `true` if you and your friend are equally strong, `false` otherwise.