211

Equivalent Arrays – Take Two

You solved this problem before. But, for this version, the arrays may have values that appear more than once.

So the two arrays below are "shift equivalent".

Consider the two arrays a and b.

a:	3	1	3	2	3

b:	3	2	3	3	1

It is possible to transform array a into array b by **right shifting** each element of a to the "right" three places. If an element "falls off" the back of the array have it come around the front and keep counting positions. That is how 3 in array ended up in the first position of array b. One way to look at this is to imagine that we are moving the element around in a circular manner.

In the example above, we have right shifted the array a 3 positions to the right to get array b.

Definition: Let a and b be two integer arrays of the same length. We say that they are "shift equivalent" if array a can be right shifted to create array b.

Problem

Write a function

bool equivalent(int a[], int b[], int n)

which takes two arrays a and b of length n and returns **true** is they are shift equivalent and **false** otherwise.