Exercise 2.2.21: Triplicates. Given three lists of N names each, devise a linearithmic algorithm to determine if there is any name common to all three lists, and if so, return the first such name.

Solution: First, sort the array in ascending order so that it is easier to identify values/elements without repetition.

Then compare the values/elements of each array while increasing the index to be compared at, until finds a match. Kind of similar to the concept of how you try to merge arrays together in mergesort going little by little, incrementing each array index.

```
//Pseudocode Triplicates
//By Anando Zaman

function triplicate(a[], b[], c[])

    //Sort the arrays so that comparisons are easier
    Arrays.sort(a);
    Arrays.sort(b);
    Arrays.sort(c);

    //index trackers for each array. Used for value comparisons
    index_a = 0;
    index_b = 0;
    index_c = 0;

    while ((index_a < a.length) && (index_b < a.length)):

        if(a[index_a] > a[index_b]):
            index_b++

        else if(a[index_a] < a[index_b]):
            index_a++

        else: //If a & b have the same value, then check c-array
            while(index_c < c.length):

                //If value of index at c is smaller than the matching a & b values found
                //then check next index of c until we find a match
                if(c[index_c] < a[index_a]):
                    index_c++

                //if value is larger than the matching a & b array values
                //then terminate loop as the value does not exist in c-array
                else if(c[index_c] > a[index_a]):
                    break

                //Return value if it matches that of a & back
                //This value represents the value that is common
                //in all three arrays, known as triplicate
                else:
                    return c[index_c]
```