CS 314 Exam One Review — Remove Odd Indices

Array-based Lists

Implement an instance method for the `GenericList` class which will remove all elements at odd indices. This method will then return the number of elements removed from the list. None of the elements in the list will be `null`.

Complete the following method.

```
// Removes all elements at odd indices
// pre: none
// post: returns number of elements removed from list
public int removeOddIndices()
```

Here are some example calls to `removeOddIndices()`:
[0, 1, 2, 3, 4, 5].`removeOddIndices()`→ [0, 2, 4], returns 3
["A", "B", "C", "D", "E"].`removeOddIndices()` → ["B", "D"], returns 3
[314].`removeOddIndices()` → [314], returns 0
[].`removeOddIndices()` → [], returns 0

Your method will be in the following `GenericList` class:

```
public class GenericList<E>{
  private int size;
  private E[] con;
  // ...
}
```

Do not use or assume there are any provided methods in the GenericList class. Do not use any other Java classes or methods.

```java
// Removes all elements at odd indices
// pre: none
// post: returns number of elements removed from list
public int removeOddIndices() {
    int numToRemove = size / 2;
    if(numToRemove == 0)
        return 0;

    int indexToReplace = 1;
    //Loop through elements we want to keep
    for(int i = 2; i < size; i += 2){
        con[indexToReplace] = con[i];
        indexToReplace++;
    }
    //Null out elements past new size
    int newSize = size - numToRemove;
    for(int i = newSize; i < size; i++){
        con[i] = null;
    }

    //Set size variable
    size = newSize;
    return numToRemove;
}
```

This is a relatively simple problem about removing elements from an array based list. However, when dealing with array based lists there are a lot of details which can be easy to forget. For instance, it is important to update the size instance variable and to null out the elements past the new size of the list to prevent memory leaks.