ArrayList and Vector

# Practice

* Write a function which takes an ArrayList of characters and removes all ‘e’ characters in it without changing the order of other elements.(removeAll() method is not allowed)

*Main method:*

public static void main(String[] args){  
 ArrayList<Character> arl = new ArrayList<>();  
 for(int i=0; i<10; i++)  
 arl.add((char)  
 Math.round(Math.random()\*3+100) );  
 System.out.println(arl);  
 cleanse(arl);  
 System.out.println(arl);  
}

*solution:*

static void cleanse(ArrayList<Character> arl) {  
 while(arl.remove(Character.valueOf('e')));  
}

* Write a function which takes an ArrayList of integers and a rotation amount (integer) and rotates the ArrayList by the amount given.
* example
  + rotate([3,1,5,7], 0) –> [3,1,5,7]
  + rotate([2,7,4,4,4], 2) –> [4,4,2,7,4]
  + rotate([2,5,7,2,1,3], -1) –> [5,7,2,1,3,2]

*Main method:*

public static void main(String[] args){  
 ArrayList<Integer> arl = new ArrayList<>();  
 for(int i=0; i<10; i++)  
 arl.add((int)(Math.random()\*10));  
 System.out.println(arl);  
 rotate(arl, 2);  
 System.out.println(arl);  
}

*solution1:*

static void rotate(ArrayList<Integer> arl, int k) {  
 if(k < 0)  
 for(int i=0; i<-k; i++)  
 arl.add(arl.remove(0));  
 else  
 for(int i=0; i<k; i++)  
 arl.add(0, arl.remove(arl.size()-1));  
}

*solution2:*

static void rotate(ArrayList<Integer> arl, int k) {  
// This is also OK but harder to understand  
 for(int i=0; i<-k; i++)  
 arl.add(arl.remove(0));  
 for(int i=0; i<k; i++)  
 arl.add(0, arl.remove(arl.size()-1));  
}

* Write a program which takes integers from user until user enters a 0 and displays them in ascending order (0 won’t be displayed).

*solution:*

public static void main(String[] args){  
 Scanner input = new Scanner(System.in);  
 // get numbers  
 ArrayList<Integer> arl = new ArrayList<>();  
 int next;  
 while((next = input.nextInt()) != 0)  
 arl.add(next);  
 // sort them  
 Collections.sort(arl);  
 // display  
 for(int num: arl)  
 System.out.println(num);  
}

# Project

* Write a function which takes an ArrayList of Boolean and fills its last half with null references. Note that return-type of your function must be void. You can assume that the size of the parameter list is always even.

*Main method:*

public static void main(String[] args){  
 ArrayList<Boolean> arl = new ArrayList<>();  
 for(int i=0; i<10; i++)  
 arl.add(Math.random() < 0.5);  
 System.out.println(arl);  
 fill(arl);  
 System.out.println(arl);  
}

*solution:*

static void fill(ArrayList<Boolean> arl) {  
 for(int i=arl.size()/2; i<arl.size(); i++)  
 arl.set(i, null);  
}

* Write a function which takes an ArrayList of Strings words and adds just enough strings at the end of it to ensure that the following condition is satisfied: “For every string w in words the reverse of w is also in words.” If the condition is already satisfied, no action is needed.
* example
  + complete([“aa”, “aca”, “ba”, “ab”]) would not change anything.
  + complete([“ab”, “qe”, “eq”]) appends “ba”.

*Main method:*

public static void main(String[] args){  
 ArrayList<String> arl = new ArrayList<>();  
 arl.add("ab");  
 arl.add("qe");  
 arl.add("eq");  
 System.out.println(arl);  
 complete(arl);  
 System.out.println(arl);  
}

*solution:*

public static void complete(ArrayList<String> words) {  
 int size = words.size();  
 for (int i = 0; i < size; i++) {  
 String word = words.get(i);  
 String reverse = new StringBuilder(word).reverse().toString();  
 if (!words.contains(reverse)) {  
 words.add(reverse);  
 size++;  
 }  
 }  
}

* Write a Java method that takes two ArrayLists of integers as input and returns a new ArrayList that contains only the integers that appear in both input lists, in the order they first appear in the first input list.
* example
  + common([1, 2, 3, 4, 5], [2, 4, 6, 8, 10]) returns [2, 4]
  + common([1, 2, 3, 4, 5], [6, 7, 8, 9, 10]) returns []

*Main method:*

public static void main(String[] args){  
 ArrayList<Integer> arl1 = new ArrayList<>();  
 ArrayList<Integer> arl2 = new ArrayList<>();  
 ArrayList<Integer> arl3 = new ArrayList<>();  
 for(int i=0; i<5; i++){  
 arl1.add(i+1);  
 arl2.add(i+6);  
 arl3.add(i+3);  
 }  
 System.out.println(arl1);  
 System.out.println(arl2);  
 System.out.println(arl3);  
 System.out.println(common(arl1, arl2));  
 System.out.println(common(arl1, arl3));  
}

*solution:*

public static ArrayList<Integer> common(ArrayList<Integer> arl1, ArrayList<Integer> arl2) {  
 ArrayList<Integer> res = new ArrayList<>();  
 for (int elem : arl1) {  
 if (arl2.contains(elem)) {  
 res.add(elem);  
 }  
 }  
 return res;  
}