The structure of NewList:

public class NewList<T> extends ArrayList<T> {

private Object[] data;

private int manyItems;

}

Task 1

public <T> int remove1(T target) {

manyItems = data.length

for index=0 to manyItems do

begin

if data[index] == target then

for j = index to manyItems – 1 do

begin

data[j] = data[j + 1];

end

manyItems--

index--

end

data = Arrays.copyOf(data, manyItems);

return data.length

}

Task 2

public <T> int remove2() {

manyItems = data.length

Map<Object, Integer> helper = new HashMap

for each Object o in data do

begin

helper.put(o, helper.getOrDefault(o, 0) +1)

end

List<Object> target = new ArrayList

for each Map.Entry<Object, Integer> entry in helper.entrySet do

begin

if entry.getValue() == 3 then

target.add(entry.getKey())

end

for each Object o in target do

begin

for index = 0 to manyItems do

begin

if data[index] == o then

for j = index to manyItems – 1do

begin

data[j] = data[j + 1];

end

manyItems--;

index--;

end

end

data = Arrays.copyOf(data, manyItems)

println("The new List: " + Arrays.toString(data))

return data.length

}

Task 3

public <T> List<List<String>> anagrams() {

String[] str = new String[data.length]

for i = 0 to data.length do

begin

str[i] = data[i].toString

end

if data.length == 0 then

return new ArrayList

Map<String, Set> helper = new HashMap<String, Set>

for each String s in str do

begin

char[] ca = s.toCharArray

Arrays.sort(ca)

String K = String.valueOf(ca)

if !helper.containsKey(K) then

helper.put(K, new HashSet())

helper.get(K).add(s)

end

List<List<String>> res = new ArrayList

for each Set set in helper.values do

begin

List<String> list = new ArrayList(set)

if list.size() > 1 then

res.add(list)

end

return res;

}