Consider the following skeletal class

**import java.util.ArrayList;**

**import java.util.Arrays;**

**import java.util.Collections;**

**public class ArrayListExercise2 {**

**private ArrayList<String> names ;**

**public ArrayListExercise2() {**

**String [] list = {“ABIGAIL”, “ADAM”, “ADILE”, “ADRIAN-GABRIEL”,**

**“ALAN”, “ALEX”, “ALEXANDRA”, “ALYSON”, “ANDREI”, “AOIFE”, “BABAFEMI”, “BARRY”, “BAYAN”, “BEN”, “BENJAMIN”, “BLAKE”, “CAEL”, “CALUM”, “CAOIMHE”, “CARLOS”, “CATHAL”, “CIAN”, “CLANCY”, “CLEM”, “CLODAGH”, “CONOR”, “CORMAC”, “DANIEL”, “DARRAGH”, “DARREN”, “DAWID”, “DEAN”, “DEBBIE”, “DENNIS”, “DERMOT”, “DOMINION”, “DWAIN”, “DYLAN”, “EDDIE”, “EDISON”, “EDWARD”, “EL AMEEN”, “ELLIOT”, “EMILY”, “EOIN”, “ETHAN”, “EVAN”, “FAROUQ”, “FARRAN”, “FERGAL”, “GABRIELA”, “GAVIN”, “HAO”, “HEMSANKHA”, “HONG TAO”, “IMELDA”, “JABLONSKI”, “JACK”, “JAMES”, “JASON”, “JIM”, “JOHN”, “JOSH”, “KAMIL”, “KATE”, “KIM”, “KONAN”, “KORATHAN”, “LABHAOISE”, “LAKEISHA”, “LEANNA”, “LEON”, “LIAM”, “LIU”, “LUKE”, “LUXIN”, “MAHIR”, “MARK”, “MATTHEW”, “MICHAEL”, “MORRISSEY”, “MOUTHER”, “NAN”, “NATHAN”, “NATHANIEL”, “NEEMA”, “NIALL”, “OLAYINKA”, “OUSSOUBI”, “PAULIS”, “QUINN”, “RAINERS”, “RAJAT”, “REBECCA”, “REPATH”, “RIAN”, “RICHARD”, “ROHAN”, “RUAIRI”, “SARAH”, “SAW”, “SCOTT”, “SEAN”, “SERGIU”, “SIMON”, “SIWEI”, “SOHAM”, “SOPHIA”, “STEPHEN”, “SULTAN”, “TEMILOLUWA”, “TENDAI”, “THOMAS”, “TIANXING”, “TIMOTHY”, “TOM”, “TOMASZ”, “VALERII”, “YONGXIN”, “ZEBA”, “JAI”};**

**names = new ArrayList<String>(Arrays.asList(list));**

**}**

**}**

1. Write the code for a method with the following header

**public String capitalise() {**

The method alters each name in the list so that the first character of the name is uppercase and the remaining letters are lowercase.

**public void capitalise() {**

**String capitalisedName;**

**for(int i=0; i < names.size(); i++) {**

**capitalisedName = names.get(i);**

**capitalisedName = Character.toUpperCase(capitalisedName.charAt(0)) +**

**capitalisedName.substring(1).toLowerCase();**

**names.set(i,capitalisedName);**

**}**

**}**

1. Write the code for a method with the following header

**public ArrayList<String> removeWordsStartingWith(String portion)**

The method deletes all the entries in the list that start with the text specified in the String parameter portion. The method returns an ArrayList of the names deleted.

**// Deleting as we go from the end to the beginning of the list**

**public ArrayList<String> removeWordsStartingWithV0(String portion) {**

**System.out.println(names.toString());**

**ArrayList<String> namesRemoved = new ArrayList<String>();**

**for(int i = names.size()-1 ; i >= 0; i--) {**

**if(names.get(i).startsWith(portion)) {**

**namesRemoved.add(names.get(i));**

**names.remove(i) ;**

**// Alternatively**

**// namesRemoved.add(names.remove(i));**

**}**

**}**

**System.out.println(names.toString());**

**return namesRemoved;**

**}**

**//**

**// Alternatively, deleting as we go from the beginning to the end**

**// of the list**

**public ArrayList<String> removeWordsStartingWithV1(String portion) {**

**System.out.println(names.toString());**

**ArrayList<String> namesRemoved = new ArrayList<String>();**

**int i ;**

**i = 0 ;**

**while(i < names.size()) {**

**if(names.get(i).startsWith(portion)) {**

**namesRemoved.add(names.get(i));**

**names.remove(i) ;**

**// Alternatively**

**// namesRemoved.add(names.remove(i));**

**} else {**

**i++ ;**

**}**

**}**

**System.out.println(names.toString());**

**return namesRemoved;**

**}**

1. Write the code for a method with the following header

**public int vowelCount()**

The method returns a count of the number of vowels in the names contained in the list.

**public int vowelCount() {**

**int vowelTotal = 0;**

**for(String s : names) {**

**vowelTotal += vowelCount(s);**

**}**

**return vowelTotal;**

**}**

**private int vowelCount(String s) {**

**int count=0;**

**for(char ch : s.toCharArray()) {**

**if("aeiouAEIOU".indexOf(ch) != -1) {**

**count++;**

**}**

**}**

**return count;**

**}**

1. Write the code for a method with the following header

**public String firstNameAlphabetically() {**

The method returns the name that would be listed first if the names were sorted alphabetically. TIP: Find the Java Collections class page using your preferred search engine and look at the **min** and **max** methods provided.

**public String firstNameAlphabetically() {**

**return Collections.min(names);**

**}**

1. Write the code for a method with the following header

**public String lastNameAlphabetically()**

The method returns the name that would be listed last if the names were sorted alphabetically. TIP: Find the Java Collections class page using your preferred search engine and look at the min and max methods provided.

**public String lastNameAlphabetically() {**

**return Collections.max(names);**

**}**