

```

package java_week_04;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class assignment {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        /* 1.Create an array of int called ages that contains the following values:
        * (3, 9, 23, 64, 2, 8, 28, 93).
        *
        * a. Programmatically subtract the value of the first element in the array
        * from the value in the last element of the array (i.e. do not use ages
        * [7] in your code).Print the result to the console.
        *
        * b. Add a new age to your array and repeat the step above to ensure it is
        * dynamic (works for arrays of different lengths).
        *
        * c. Use a loop to iterate through the array and calculate the average age.
        * Print the result to the console.
        */

        List<Integer> ages = new ArrayList<>();
        ages.addAll(Arrays.asList(3, 9, 23, 64, 2, 8, 28, 93));
        System.out.println(ages);// add ages

        System.out.println(subtractFirstFromLast(ages)); // a

        ages.add(43);
        System.out.println(ages);
        System.out.println(subtractFirstFromLast(ages)); // b

        System.out.format("%.2f%n",agesAverage(ages)); //c

        System.out.println("1-----");

        /* 2.Create an array of String called names that contains the following
        * values: "Sam", "Tommy", "Tim", "Sally", "Buck", "Bob".
        *
        * a. Use a loop to iterate through the array and calculate the average number
        * of letters per name. Print the result to the console.

```

```

*
* b. Use a loop to iterate through the array again and concatenate all the
* names together, separated by spaces, and print the result to the console.
*/

List<String> names = new ArrayList<>();
names.addAll(Arrays.asList("Sam", "Tommy", "Tim", "Sally", "Buck", "Bob"));
System.out.println(names); //2

System.out.format("%.2f%n",averageNumberOfLettersPerName(names)); //a

System.out.println(concatenateAllName(names));

System.out.println("2-----");

/* 3.How do you access the last element of any array?
*/

System.out.println(names.get(names.size()-1));

System.out.println("3-----");

/* 4.How do you access the first element of any array?
*/

System.out.println(names.get(0));

System.out.println("4-----");

/* 5. Create a new array of int called nameLengths. Write a loop to iterate over
* the previously created names array and add the length of each name to the
* nameLengths array.
*/

List<Integer> nameLengths = new ArrayList<>();
nameLengths = addNameLength(names);
System.out.println(nameLengths);

System.out.println("5-----");

/* 6. Write a loop to iterate over the nameLengths array and calculate the sum of
* all he elements in the array. Print the result to the console.
*/

System.out.println(sumOfNamelength(nameLengths));

```

```

System.out.println("6-----");

/* 7. Write a method that takes a String, word, and an int, n, as arguments and
 * returns the word concatenated to itself n number of times.
 * (i.e. if I pass in "Hello" and 3, I expect the method to return "HelloHelloHello").
 */
System.out.println(severalTimes("Hello", 3));

System.out.println("7-----");

/* 8. Write a method that takes two Strings, firstName and lastName, and returns
 * a full name (the full name should be the first and the last name as a String
 * separated by a space).
 */
System.out.println(fullName("Tom", "Lee"));

System.out.println("8-----");
/* 9. Write a method that takes an array of int and returns true if the sum of all
 * the ints in the array is greater than 100.
 */

System.out.println(listSumGreaterThan100(ages));

System.out.println("9-----");

/* 10. Write a method that takes an array of double and returns the average
 * of all the elements in the array.
 */

List<Double> nums = new ArrayList<>();
nums.addAll(Arrays.asList(12.35, 29.77, 99.54, 34.92, 45.34));

System.out.println(listAverage(nums));

System.out.println("10-----");

/* 11. Write a method that takes two arrays of double and returns true if the
 * average of the elements in the first array is greater than the average
 * of the elements in the second array.
 */

```

```

List<Double> nums1 = new ArrayList<>();
nums1.addAll(Arrays.asList(66.24, 34.76, 9.67, 46.27, 89.33, 34.22));

System.out.println(ifList1AverageGreaterThanList2(nums, nums1));

System.out.println("11-----");

    /*12. Write a method called willBuyDrink that takes a boolean isHotOutside,
    * and a double moneyInPocket, and returns true if it is hot outside and
    * if moneyInPocket is greater than 10.50.
    */
System.out.println(willBuyDrink(true, 7.0));

System.out.println("12-----");

    /* 13. Create a method of your own that solves a problem. In comments,
    * write what the method does and why you created it.
    */

/*Create a array that contains monthly living cost of family, if cost greater
* than specific % of monthly income, return false.
*/

List<Double> costs = new ArrayList<>();
costs.add(500.00);
costs.add(30.45);
costs.add(1200.00);
costs.add(450.00);
costs.add(300.00);
double monthlyIncome = 4000.00;
double n = 0.8;
System.out.println(willCostsLessThanIncome(costs, monthlyIncome, n));

System.out.println("13-----");

}

public static int subtractFirstFromLast(List<Integer> list ) { // 1.a b
    int result = 0;
    result = list.get(list.size()-1)-list.get(0);
    return result;
}

public static double agesAverage(List<Integer> list) { // 1.c
    double result = 0;

```

```

        for (int i = 0; i < list.size(); i++) {
            result += list.get(i);
        }
        return result/list.size();
    }

    public static double averageNumberOfLettersPerName(List<String> list) { // 2.a
        double result = 0;
        for (int i = 0; i < list.size(); i++) {
            result += list.get(i).length();
        }
        return result/list.size();
    }

    public static String concatenateAllName(List<String> list) { // 2.b
        String name = "";
        for (int i = 0; i < list.size()-1; i++) {
            name += list.get(i) + " ";
        }

        return name + list.get(list.size()-1);
    }

    public static List<Integer> addNameLength(List<String> list1) { // 5
        List<Integer> lists = new ArrayList<>();
        int list = 0;
        for (int i = 0; i < list1.size(); i++) {
            list = list1.get(i).length();
            lists.add(list);
        }
        return lists;
    }

    public static int sumOfNamelength(List<Integer> list) { // 6
        int result = 0;
        for(int i = 0; i < list.size(); i++) {
            result += list.get(i);
        }
        return result;
    }

    public static String severalTimes(String str, int n) { // 7
        String strs = "";
        for(int i = 0; i < n; i++) {

```

```

        strs +=str;
    }
    return strs;
}

public static String fullName(String str1, String str2) {           // 8
    String strs = str1 + " " + str2;
    return strs;
}

public static boolean listSumGreaterThan100(List<Integer> lists) {    // 9
    int result = 0;
    for(int list : lists) {
        result += list;
    }
    if (result >=100)
        return true;
    else
        return false;
}

public static double listAverage(List<Double> lists) {              // 10
    double result = 0;
    for(double list : lists) {
        result += list / lists.size();
    }
    return result;
}

public static boolean ifList1AverageGreaterThanList2(List<Double> lists1, List<Double> lists2) {
    double result1 = 0;                                           // 11
    double result2 = 0;
    for (double list1 : lists1) {
        result1 += list1 / lists1.size();
    }
    for (double list2 : lists2) {
        result2 += list2 / lists2.size();
    }
    if (result1 > result2)
        return true;
    else
        return false;
}

```

```

}

public static boolean willBuyDrink(boolean isHotOutside, double num){    // 12
    if(isHotOutside && num > 10.50) {
        return true;
    }
    else
        return false;
}

public static boolean willCostsLessThanIncome(List<Double> lists, double income, double n ) {
    double result = 0;                                // 13
    for (double list : lists) {
        result += list;
    }
    if(result > income * n)
        return false;
    else
        return true;
}
}

```