```
::::::::::::::
Lab1Status.txt
::::::::::::::
Problem 1:
Status update 1:
Output separates correctly due to for loop, but only displays total number of peop
le rather than separating i.e. "name number 1" "name number 2" etc.

Status update 2:
Name number lines now separated, but "shouts Hello" line is missing, only first na
me displays.

Status update 3:
Names all combined with no "and" for ex: "Tony and Joe".

Status update 4:
All names besides last are concatenated if more than 3.

Status update 5:
Accidental extra space between commas.

Status update 6:
Statement repeats itself, incorrect output.

Status update 7:
Output is correct with 2 people.

Status update 8:
Output is correct entirely.

Problem 2:
Status update 1:
Output fully functional for when user input equals zero, everything else non funct
ional.

Status update 2:
Output correctly interprets number of pairs to amount of lines needed for number a
nd weight.

Status update 3:
Program is fully functional with correct output except that weighted mean is not d
ivided by total of all weight(s). Simple mean is correct.

Problem 3:
Status update 1:
Everything is functional except for Each + method ref and palindrome

Status update 2:
Palindrome still not operating

Status update 3:
ERROR RESOLVED: missing break lines

Output fully functional. ::::::::::::::
Lab1Conclusions.txt
::::::::::::::
In this lab, I learned how to iterate through an ArrayList in multiple different w
ays i.e. (reading through in reverse, reading the third index), and the way to per
form those iterations can be done in even MORE ways. I found doing the same thing
in a different way to be useful and has definitely gained me some experience as a
computer scientist. ::::::::::::::
Lab1P1Driver.java
::::::::::::::
/*

 * Purpose: Data Structure and Algorithms Lab 1 Problem 1
 * Status: Complete and thoroughly tested
 * Last update: 1/21/23
 * Submitted:  1/22/23
 * Comment: test suite and sample run attached
 * Comment: I declare that this is entirely my own work
 * @author: Antonio Rosado
 * @version: 2023.01.21

 */

import java.io.*;

class Lab1P1Driver
{
    static BufferedReader stdin = new BufferedReader (new InputStreamReader(System
.in)); // input for user
    public static void main(String args[]) throws IOException
    {
        // Problem 1
        // parse int then
        // for loop for output stream
        // build string while reading in

        System.out.print("\nEnter number of people: ");
        int n = Integer.parseInt(stdin.readLine()); // parse num given by user
        System.out.println(n);
        String[] names = new String[n]; // array of String[] based on number

        for (int index = 0; index < n; index++)
        {
            System.out.print("Enter name number " + (index + 1) + ": "); // index
+ 1 because index starts at 0
            names[index] = stdin.readLine(); // name at desired index is input
            System.out.println(names[index]);
        }
        // formatting for different possible outcomes
        if(names.length == 1)
        {
            System.out.print(names[0] + " shouts Hello Class!!");
        }

        else if(names.length == 2)
        {
            System.out.print(names[0] + " and " + names[1] + " shouts Hello Class!
!");
        }

        else {
            for(int l = 0; l < (names.length - 2); l++)
            {
                System.out.print(names[l] + ", ");
            }
            System.out.print(names[n-2] + "," + " and " + names[n-1] + " shouts "
+ "Hello Class!!");
        }
    }
}
::::::::::::::
```

Lab1P1Sampleruns.txt
::::::::::::::

```
Enter number of people: 1
Enter name number 1: Joe
Joe shouts Hello Class!!
Enter number of people: 2
Enter name number 1: John
Enter name number 2: Jane
John and Jane shouts Hello Class!!
Enter number of people: 3
Enter name number 1: John
Enter name number 2: Jane
Enter name number 3: Joe
John, Jane, and Joe shouts Hello Class!!
Enter number of people: 4
Enter name number 1: John
Enter name number 2: Jane
Enter name number 3: Joe
Enter name number 4: Jack
John, Jane, Joe, and Jack shouts Hello Class!!
Enter number of people: 5
Enter name number 1: John
Enter name number 2: Jane
Enter name number 3: Joe
Enter name number 4: Jack
Enter name number 5: Jim
John, Jane, Joe, Jack, and Jim shouts Hello Class!!:::::::::::::::
```

Lab1P2Driver.java
::::::::::::::

```
/*
 * Purpose: Data Structure and Algorithms Lab 1 Problem 2
 * Status: Complete and thoroughly tested
 * Last update: 1/21/23
 * Submitted:  1/22/23
 * Comment: test suite and sample run attached
 * Comment: I declare that this is entirely my own work
 * @author: Antonio Rosado
 * @version: 2023.01.21
 */

import java.io.*;

class Lab1P2Driver {
    static BufferedReader stdin = new BufferedReader (new InputStreamReader(System
.in)); // input for user
    public static void main(String args[]) throws IOException
    {
        System.out.print("Enter number of pairs: ");
        int pairs = Integer.parseInt(stdin.readLine()); // parse num given by user
        System.out.println(pairs);

        // sums for final results
        double sum_weighted = 0.0;
        double sum_simple = 0.0;
        // arrays for number of pairs and weight
        double[] weightArray = new double[pairs];
        double[] numOfPairs = new double[pairs];
        double weight_total = 0.0; // total weight(s) added

        if(pairs == 0) // if user inputs 0, program will cease at this point
        {
            System.out.print("Simple arithmetic mean of these " + pairs + " number
s is: " + pairs + "\n");
            System.out.print("Weighted arithmetic mean of these " + pairs + " pair
s is: " + pairs);
        }

        else {

            for (int index = 0; index < pairs; index++)
            {
                System.out.print("Enter number: ");
                double number = Double.parseDouble(stdin.readLine()); // parse num
 given by user
                System.out.println(number);

                System.out.print("Enter weight: ");
                double weight = Double.parseDouble(stdin.readLine()); // parse dou
ble given by user
                System.out.println(weight);

                sum_simple += number; // sum for simple mean
                numOfPairs[index] = number; // index of number of pairs aligns wit
h number given by user
                weight_total += weight;
                sum_weighted += (number * weight); // sum for weighted mean
                weightArray[index] = weight; // index of array of weight should al
ign with weight given by user

            }

            double weightedMean = sum_weighted / weight_total; // final result for
 weightedMean
            double simpleMean = sum_simple / pairs; // final result for simpleMean

            System.out.print("Simple arithmetic of these " + pairs + " numbers is:
 " + simpleMean + "\n");
            System.out.print("Weighted arithmetic mean of these " + pairs + " pair
s is: " + weightedMean);
        }

    }
}:::::::::::::::
```

Lab1P2Sampleruns.txt
::::::::::::::

```
Enter number of pairs: 6
Enter number: 9.0
Enter weight: 3.4
Enter number: 24.0
Enter weight: 46.6
Enter number: 67.0
Enter weight: 9.5
Enter number: 9.0
Enter weight: 23.1
Enter number: 84.0
Enter weight: 45.0
Enter number: 32.0
Enter weight: 3.1
Simple arithmetic of these 6 numbers is: 37.5
Weighted arithmetic mean of these 6 pairs is: 44.93190512624331Enter number of pai
rs: 6
Enter number: 9.0
Enter weight: 3.4
```

```
Enter number: 24.0
Enter weight: 46.6
Enter number: 67.0
Enter weight: 9.5
Enter number: 9.0
Enter weight: 23.1
Enter number: 84.0
Enter weight: 45.0
Enter number: 32.0
Enter weight: 3.1
Simple arithmetic of these 6 numbers is: 37.5
Weighted arithmetic mean of these 6 pairs is: 44.93190512624331Enter number of pai
rs: Enter number of pairs: 0
Simple arithmetic mean of these 0 numbers is: 0
Weighted arithmetic mean of these 0 pairs is: 0::::::::::::::
Lab1P3Driver.java
::::::::::::::
/*
 * Purpose: Data Structure and Algorithms Lab 1 Problem 3
 * Status: Complete and thoroughly tested
 * Last update: 1/22/23
 * Submitted:  1/23/23
 * Comment: test suite and sample run attached
 * Comment: I declare that this is entirely my own work
 * @author: Antonio Rosado
 * @version: 2023.01.23
 */

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.ListIterator;

class Lab1P3Driver {
    ArrayList<Character> data = new ArrayList<Character>(); // ArrayList for Chara
cter(s)
    static BufferedReader stdin = new BufferedReader(new InputStreamReader(System.
in));

    public static void main(String args[]) throws IOException {
        Lab1P3Driver charProgram = new Lab1P3Driver();
        int input = 0;
        while (input != 6) {
            System.out.print("What would you like to do? \n"
                             + "1. Add a character \n"
                             + "2. Display a list (in order entered) \n"
                             + "3. Display a list (reversed) \n"
                             + "4. Display every 3rd element of list \n"
                             + "5. Test if list is a palindrome \n"
                             + "6. Quit. \n");
            input = Integer.parseInt(stdin.readLine());
            System.out.println("You chose: " + input + "\n");

            // possible cases for initial input
            switch (input) {
            case 1:
                System.out.print("Enter the character you would like to add to the
 list: ");
                char c = stdin.readLine().charAt(0);
                System.out.println(c);
                charProgram.add(c);
                break;

            case 2:
                int second_choice = 0;
                System.out.print("How would you like to display a list? (in order)
 \n"
                                 + "1. Direct Index Access \n"
                                 + "2. With An Iterator \n"
                                 + "3. For-Each Loop \n"
                                 + "4. For-Each w/ Lambda \n"
                                 + "5. For-Each w/ Method Reference \n");

                // possible cases for second input
                second_choice = Integer.parseInt(stdin.readLine());
                switch (second_choice) {
                case 1:
                    charProgram.displayDirectIndexAccess();
                    break;

                case 2:
                    charProgram.displayIterators();
                    break;

                case 3:
                    charProgram.displayForEach();
                    break;

                case 4:
                    charProgram.displayForEachLambda();
                    break;

                case 5:
                    charProgram.displayForEachMethodRef();
                    break;

                }
                break;

            // possible cases for third input
            case 3:
                int third_choice = 0;
                System.out.print("How would you like to display a list? (in revers
e) \n"
                                 + "1. Iterator/ListIterator \n"
                                 + "2. For-Loop \n");
                third_choice = Integer.parseInt(stdin.readLine());
                switch (third_choice) {
                case 1:
                    charProgram.reverseWithIterator();
                    break;

                case 2:
                    charProgram.reverseWithForLoop();
                    break;
                }
                break;

            case 4:
                charProgram.displayEveryThirdItemForLoop();
                break;
```

```
            case 5:
                charProgram.testIfPalindrome();
                break;
            }
        }
    }

    /**
     * adds char to ArrayList
     *
     * @param add
     */
    public void add(char add) {
        data.add(add);
    }

    /**
     * Process ArrayList via direct index access
     */
    public void displayDirectIndexAccess() {
        int dataSize = data.size();
        for (int index = 0; index < dataSize; index++) {
            System.out.println(data.get(index));
        }
    }

    /**
     * <
     * Process ArrayList via an iterator
     */
    public void displayIterators() {
        Iterator<Character> iterator = data.iterator();
        while (iterator.hasNext()) {
            System.out.println(iterator.next().toString());
        }
    }

    /**
     * Process ArrayList via For Each loop
     */
    public void displayForEach() {
        for (char Character : data) {
            System.out.println(data.get(Character));
        }
    }

    /**
     * Process ArrayList via forEach method + lambda expression
     */
    public void displayForEachLambda() {
        data.forEach(c -> System.out.println(c));
    }

    /**
     * Process ArrayList via forEach method + method reference
     */
    public void displayForEachMethodRef() {
        data.forEach(System.out::println);
    }

    // all reverse methods below

    /**
     * Process ArrayList in reverse with listIterator
     */
    public void reverseWithIterator() {
        int dataSize = data.size();
        ListIterator iterator = data.listIterator(dataSize);
        while (iterator.hasPrevious()) {
            // previous method returns previous element
            System.out.println(iterator.previous().toString());
        }
    }

    /**
     * Process ArrayList in reverse with for loop
     */
    public void reverseWithForLoop() {
        int dataSize = data.size();
        for (int index = dataSize - 1; index >= 0; index--) // decrement
        {
            System.out.println(data.get(index));
        }
    }

    // third index method(s) below

    /**
     * Process ArrayList by every 3rd index
     */
    public void displayEveryThirdItemForLoop() {
        int dataSize = data.size();
        System.out.println(data.get(0));
        for (int index = 0; index < dataSize; index += 3) {
            System.out.println(data.get(index));
        }
    }

    // all palindrome method(s) below

    /**
     * Process ArrayList to check if it's a palindrome
     */
    boolean testIfPalindrome() {
        int dataSize = data.size();
        for (int index = 0; index < dataSize / 2; index++) {
            if (data.get(index) != data.get((dataSize - index) - 1)) {
                System.out.println("Given list is NOT a palindrome");
                return false;
            }
        }
        System.out.println("Given list IS a palindrome");
        return true;

    }
}:::::::::::::::
Lab1P3Sampleruns.txt
:::::::::::::::
What would you like to do?
1. Add a character
2. Display a list (in order entered)
3. Display a list (reversed)
4. Display every 3rd element of list
5. Test if list is a palindrome
```

```
6. Quit.
You chose: 1

Enter the character you would like to add to the list: r
What would you like to do?
1. Add a character
2. Display a list (in order entered)
3. Display a list (reversed)
4. Display every 3rd element of list
5. Test if list is a palindrome
6. Quit.
You chose: 1

Enter the character you would like to add to the list: a
What would you like to do?
1. Add a character
2. Display a list (in order entered)
3. Display a list (reversed)
4. Display every 3rd element of list
5. Test if list is a palindrome
6. Quit.
You chose: 1

Enter the character you would like to add to the list: c
What would you like to do?
1. Add a character
2. Display a list (in order entered)
3. Display a list (reversed)
4. Display every 3rd element of list
5. Test if list is a palindrome
6. Quit.
You chose: 1

Enter the character you would like to add to the list: e
What would you like to do?
1. Add a character
2. Display a list (in order entered)
3. Display a list (reversed)
4. Display every 3rd element of list
5. Test if list is a palindrome
6. Quit.
You chose: 1

Enter the character you would like to add to the list: c
What would you like to do?
1. Add a character
2. Display a list (in order entered)
3. Display a list (reversed)
4. Display every 3rd element of list
5. Test if list is a palindrome
6. Quit.
You chose: 1

Enter the character you would like to add to the list: a
What would you like to do?
1. Add a character
2. Display a list (in order entered)
3. Display a list (reversed)
4. Display every 3rd element of list
5. Test if list is a palindrome
6. Quit.
You chose: 1

Enter the character you would like to add to the list: r
What would you like to do?
1. Add a character
2. Display a list (in order entered)
3. Display a list (reversed)
4. Display every 3rd element of list
5. Test if list is a palindrome
6. Quit.
You chose: 2

How would you like to display a list? (in order)
1. Direct Index Access
2. With An Iterator
3. For-Each Loop
4. For-Each w/ Lambda
5. For-Each w/ Method Reference
r
a
c
e
c
a
r
What would you like to do?
1. Add a character
2. Display a list (in order entered)
3. Display a list (reversed)
4. Display every 3rd element of list
5. Test if list is a palindrome
6. Quit.
You chose: 3

How would you like to display a list? (in reverse)
1. Iterator/ListIterator
2. For-Loop
r
a
c
e
c
a
r
What would you like to do?
1. Add a character
2. Display a list (in order entered)
3. Display a list (reversed)
4. Display every 3rd element of list
5. Test if list is a palindrome
6. Quit.
You chose: 4

r
r
e
r
What would you like to do?
1. Add a character
2. Display a list (in order entered)
3. Display a list (reversed)
4. Display every 3rd element of list
5. Test if list is a palindrome
```

```
6. Quit.
You chose: 5

Given list IS a palindrome
What would you like to do?
1. Add a character
2. Display a list (in order entered)
3. Display a list (reversed)
4. Display every 3rd element of list
5. Test if list is a palindrome
6. Quit.
You chose: 6

What would you like to do?
1. Add a character
2. Display a list (in order entered)
3. Display a list (reversed)
4. Display every 3rd element of list
5. Test if list is a palindrome
6. Quit.
You chose: 1

Enter the character you would like to add to the list: r
What would you like to do?
1. Add a character
2. Display a list (in order entered)
3. Display a list (reversed)
4. Display every 3rd element of list
5. Test if list is a palindrome
6. Quit.
You chose: 1

Enter the character you would like to add to the list: a
What would you like to do?
1. Add a character
2. Display a list (in order entered)
3. Display a list (reversed)
4. Display every 3rd element of list
5. Test if list is a palindrome
6. Quit.
You chose: 1

Enter the character you would like to add to the list: g
What would you like to do?
1. Add a character
2. Display a list (in order entered)
3. Display a list (reversed)
4. Display every 3rd element of list
5. Test if list is a palindrome
6. Quit.
You chose: 2

How would you like to display a list? (in order)
1. Direct Index Access
2. With An Iterator
3. For-Each Loop
4. For-Each w/ Lambda
5. For-Each w/ Method Reference
r
a
g
What would you like to do?
1. Add a character
2. Display a list (in order entered)
3. Display a list (reversed)
4. Display every 3rd element of list
5. Test if list is a palindrome
6. Quit.
You chose: 3

How would you like to display a list? (in reverse)
1. Iterator/ListIterator
2. For-Loop
g
a
r
What would you like to do?
1. Add a character
2. Display a list (in order entered)
3. Display a list (reversed)
4. Display every 3rd element of list
5. Test if list is a palindrome
6. Quit.
You chose: 4

r
r
What would you like to do?
1. Add a character
2. Display a list (in order entered)
3. Display a list (reversed)
4. Display every 3rd element of list
5. Test if list is a palindrome
6. Quit.
You chose: 5

Given list is NOT a palindrome
What would you like to do?
1. Add a character
2. Display a list (in order entered)
3. Display a list (reversed)
4. Display every 3rd element of list
5. Test if list is a palindrome
6. Quit.
You chose: 6
```