```
::::::::::::::
MyRLHelloClass.java
::::::::::::::
/*
 * Purpose: Data Structure and Algorithms Lab 1 Problem 7
 * Status: Complete and thoroughly tested
 * Last update: 01/22/20
 * Submitted:  01/28/20
 * Comment: test suite and sample run attached
 * @author: Matthew Ryan
 * @version: 2020.28.1
 */

package myRLHelloClass;

import java.io.*;
import java.lang.StringBuilder;

public class MyRLHelloClass {

	static BufferedReader stdin = new BufferedReader (new InputStreamReader(Sy
stem.in));

	public static void main(String args[]) throws IOException
	{
		int userIn;
		String message;

		System.out.print("Enter number of people: ");
		userIn = Integer.parseInt(stdin.readLine());
		System.out.println(userIn);

		message = userInput(userIn);
		System.out.println(message);
	}

	public static String userInput(int userIn) throws IOException
	{
		int count = 1;
		String message = "";
		StringBuilder builder = new StringBuilder();

		while(count <= userIn)
		{
			System.out.print("Enter name number " + count + ": ");
			String name = stdin.readLine();
			System.out.println(name + "\n");
			message = processInput(builder, count, userIn, name).toStr
ing();
			count++;
		}

		return message;
	}

	public static StringBuilder processInput(StringBuilder message, int nameCo
unt, int userInTotal, String name)
	{

		if(userInTotal == 1)
		{
			{
				message.append(name + " shouts Hello Class!!!");
				return message;
			}
		}
		else if (nameCount == 1)
		{
			if (userInTotal == 2)
				message.append(name);
			else
				message.append(name + ", ");
		}


		if ((nameCount > 1) && (nameCount < userInTotal))
		{
			message.append(name + ", ");
		}

		else if (nameCount == userInTotal)
		{
			message.append("and " + name + " shout Hello Class!!!");
		}

		return message;
	}
}
::::::::::::::
lab1_P7.output
::::::::::::::
Enter number of people: 3
Enter name number 1: John

Enter name number 2: Jane

Enter name number 3: Joe

John, Jane, and Joe shout Hello Class!!!
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

Johnand Jane shout 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 shout 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 shout Hello Class!!!
::::::::::::::
Means.java
::::::::::::::
/*

 * Purpose: Data Structure and Algorithms Lab 1 Problem 8

 * Status: Complete and thoroughly tested

 * Last update: 27/1/20

 * Submitted:  28/1/20

 * Comment: test suite and sample run attached

 * @author: Matthew Ryan

 * @version: 2020.28.1

 */

import java.io.*;

public class Means {

    static BufferedReader stdin = new BufferedReader (new InputStreamReader(System
.in));

    public static void main(String args[]) throws IOException
    {
        System.out.print("Enter number of pairs: ");
        int numberOfPairs = Integer.parseInt(stdin.readLine().trim());
        System.out.println(numberOfPairs);
        inputHandling(numberOfPairs);
    }

    public static void inputHandling(int numberOfPairs) throws IOException
    {
        float userInTotal = 0;
        float weightTotal = 0;

        int counter = 1;

        float weightToCalculate = 0;

        while(counter <= numberOfPairs)
        {
            System.out.print("Enter number " + counter + ": ");
            int userIn = Integer.parseInt(stdin.readLine().trim());
            System.out.println(userIn);
            userInTotal += userIn;

            System.out.print("Enter weight " + counter + ": ");
            float weightIn = Float.parseFloat(stdin.readLine().trim());
            System.out.println(weightIn);
            weightTotal += weightIn;

            weightToCalculate += (userIn * weightIn);

            counter++;
        }
        if(numberOfPairs > 0)
        {
            arithmeticMean(numberOfPairs, userInTotal);
            weightedMean(weightToCalculate, weightTotal, numberOfPairs);
        }
        else
        {
            numberOfPairs = 0;
            System.out.println("Simple arithmetic mean of these " + numberOfPairs
+ " numbers is: " + numberOfPairs);
            System.out.println("Weighted arithmetic mean of these " + numberOfPair
s + " numbers is: " + numberOfPairs );
        }
    }

    public static void arithmeticMean(int numberOfPairs, float userInput)
    {
        float arithmeticMean = (userInput/numberOfPairs);
        System.out.println("Simple arithmetic mean of these " + numberOfPairs + "
numbers is: " + arithmeticMean);
    }

    public static void weightedMean(float weightToCalculate, float weightTotal, in
t numberOfPairs)
    {
        float weightedMean = (weightToCalculate/weightTotal);
        System.out.println("Weighted arithmetic mean of these " + numberOfPairs +
" numbers is: " + weightedMean);
    }
}
::::::::::::::
lab1_P8.output
::::::::::::::
Enter number of pairs: 6
Enter number 1: 9
Enter weight 1: 3.4
Enter number 2: 24
Enter weight 2: 46.6
Enter number 3: 67
Enter weight 3: 9.5
Enter number 4: 9
Enter weight 4: 23.1
Enter number 5: 84
Enter weight 5: 45.0
Enter number 6: 32
Enter weight 6: 3.1
Simple arithmetic mean of these 6 numbers is: 37.5
Weighted arithmetic mean of these 6 numbers is: 44.931908
Enter number of pairs: 0
Simple arithmetic mean of these 0 numbers is: 0
Weighted arithmetic mean of these 0 numbers is: 0
::::::::::::::
Collection_Processing.java
::::::::::::::
/*

 * Purpose: Data Structure and Algorithms Lab 1 Problem 9
```

```
 * Status: Complete and thoroughly tested

 * Last update: 1/28/20

 * Submitted:  1/28/20

 * Comment: test suite and sample run attached

 * @author: Matthew Ryan

 * @version: 2020.28.1

 */

import java.util.ArrayList;
import java.util.Iterator;
import java.util.ListIterator;
import java.io.*;

public class Collection_Processing {

    static BufferedReader stdin = new BufferedReader (new InputStreamReader(System
.in));

    /*
     * The following is a menu designed to handle testing on the dataset.
     */

    public static void main(String[] args) throws IOException {
        ArrayList<Character> data = new ArrayList<Character>();

        boolean switchOn = true;
        while(switchOn == true)
        {
            System.out.println("Please select from the following:\n 1. Add \n\nFOR
WARD PROCESSING: \n 2. for \n 3. for-each \n 4. while \n 5. do-while \n 6. iterato
r \n\nREVERSE PROCESSING: \n 7. for \n 8. while \n 9. do-while \n 10. iterator \n\
nPALINDROME TESTING \n 11. for \n 12. while \n 13. do-while \n 14. iterator \n\n 0
. Exit");
            int select = Integer.parseInt(stdin.readLine());
            System.out.println(select);
            switch(select)
            {
            case 1:
                data = add(data);
                break;
            case 2:
                displayForLoop(data);
                break;
            case 3:
                displayForEachLoop(data);
                break;
            case 4:
                displayWhileLoop(data);
                break;
            case 5:
                displayDoWhileLoop(data);
                break;
            case 6:
                displayIterator(data);
                break;
            case 7:
                displayReverseForLoop(data);
                break;
            case 8:
                displayReverseWhileLoop(data);
                break;
            case 9:
                displayReverseDoWhileLoop(data);
                break;
            case 10:
                displayReverseIterator(data);
                break;
            case 11:
                System.out.println("For loop palindrome test came back: " + testIf
PalindromeForLoop(data));
                break;
            case 12:
                System.out.println("While loop palindrome test came back: " + test
IfPalindromeWhileLoop(data));
                break;
            case 13:
                System.out.println(("Do While loop palindrome test came back: " +
testIfPalindromeDoWhileLoop(data)));
                break;
            case 14:
                System.out.println("Iterator palindrome test came back: " + testIf
PalindromeIterator(data));
                break;
            case 0:
                System.out.println("\nExiting! Farewell!");
                switchOn = false;
            default:
                System.out.println("Please select from the menu!\n");
                break;
            }
        }
    }
    public static ArrayList<Character> add(ArrayList<Character> data) throws IOExc
eption
    {
        System.out.println("\nEnter character to add: ");
        char toAdd = stdin.readLine().charAt(0);
        System.out.println(toAdd);
        data.add(toAdd);
        return data;
    }

    public static void displayForLoop(ArrayList<Character> data)
    {
        for(int i = 0; i < data.size(); i++)
        {
            System.out.println(data.get(i));
        }
    }

    public static void displayForEachLoop(ArrayList<Character> data)
    {
        data.forEach(Character -> System.out.println(Character));
    }

    public static void displayWhileLoop(ArrayList<Character> data)
    {
        int counter = 0;
```

```
        while(counter < data.size())
        {
            System.out.println(data.get(counter));
            counter++;
        }
    }

    public static void displayDoWhileLoop(ArrayList<Character> data)
    {
        int counter = 0;
        do
        {
            System.out.println(data.get(counter));
            counter++;
        } while(counter < data.size());
    }

    public static void displayIterator(ArrayList<Character> data)
    {
        Iterator<Character> it = data.iterator();
        while(it.hasNext())
        {
            System.out.println(it.next());
        }
    }

    /*
     * I found 5 different ways to process collections in Java, all with their own
 strengths and weaknesses.
     */

    public static void displayReverseForLoop(ArrayList<Character> data)
    {
        for(int i = data.size() - 1; i >= 0; i--)
        {
            System.out.println(data.get(i));
        }
    }

    public static void displayReverseWhileLoop(ArrayList<Character> data)
    {
        int counter = data.size() - 1;
        while(counter >= 0)
        {
            System.out.println(data.get(counter));
            counter--;
        }
    }

    public static void displayReverseDoWhileLoop(ArrayList<Character> data)
    {
        int counter = data.size() - 1;
        do
        {
            System.out.println(data.get(counter));
            counter--;
        } while (counter >= 0);
    }

    /*
     * Was unsure if this was doable; commented out to eventually come back to if
it is.
     */

    //public static void displayReverseForEachLoop(ArrayList<Character> data)
    //{

    //}

    public static void displayReverseIterator(ArrayList<Character> data)
    {
        ListIterator<Character> it = data.listIterator(data.size());
        while(it.hasPrevious())
        {
            System.out.println(it.previous());
        }
    }

    /*
     * 4 of the 5 proceesses I could reuse for backwards iteration; I'm still a bi
t iffy on forEach.
     */

    public static boolean testIfPalindromeForLoop(ArrayList<Character> data)
    {

        boolean isPalindrome = true;
        int front = 0;
        int back = data.size()-1;

        for(; front < data.size(); )
        {


            if(data.get(front) != data.get(back))
            {
                isPalindrome = false;
                front += data.size();
            }
            else
            {
                front++;
                back--;
            }
        }

        return isPalindrome;
    }

    public static boolean testIfPalindromeWhileLoop(ArrayList<Character> data)
    {
        boolean isPalindrome = true;

        int counter = 0;

        while((counter < data.size() && (isPalindrome == true)))
        {
            if (data.get(counter) != data.get(data.size()-1-counter))
            {
                isPalindrome = false;
            }
            else
            {
                counter++;
```

```
            }
        }

        return isPalindrome;
    }

    public static boolean testIfPalindromeDoWhileLoop(ArrayList<Character> data)
    {
        boolean isPalindrome = true;
        int counter = 0;

        do
        {
            if (data.get(counter) != data.get(data.size()-1-counter))
            {
                isPalindrome = false;
            }
            else
            {
                counter++;
            }
        } while((counter < data.size() && (isPalindrome == true)));

        return isPalindrome;
    }

    public static boolean testIfPalindromeIterator(ArrayList<Character> data)
    {
        boolean isPalindrome = true;

        Iterator<Character> it = data.iterator();
        ListIterator<Character> li = data.listIterator(data.size());

        while((it.hasNext() && li.hasPrevious()) && isPalindrome == true)
        {
            if(it.next() != li.previous())
            {
                isPalindrome = false;
            }
        }
        return isPalindrome;
    }

    /*
     * Likewise, I could only find 4 ways to check for a palindrome since it was d
irectly correlated with being able to process the array backwards.
     */

}

::::::::::::::
lab1_P9.output
::::::::::::::
Please select from the following:
 1. Add

FORWARD PROCESSING:
 2. for
 3. for-each
 4. while
 5. do-while
 6. iterator

REVERSE PROCESSING:
 7. for
 8. while
 9. do-while
 10. iterator

PALINDROME TESTING
 11. for
 12. while
 13. do-while
 14. iterator

 0. Exit
1

Enter character to add:
a
Please select from the following:
 1. Add

FORWARD PROCESSING:
 2. for
 3. for-each
 4. while
 5. do-while
 6. iterator

REVERSE PROCESSING:
 7. for
 8. while
 9. do-while
 10. iterator

PALINDROME TESTING
 11. for
 12. while
 13. do-while
 14. iterator

 0. Exit
1

Enter character to add:
b
Please select from the following:
 1. Add

FORWARD PROCESSING:
 2. for
 3. for-each
 4. while
 5. do-while
 6. iterator

REVERSE PROCESSING:
 7. for
 8. while
 9. do-while
 10. iterator

PALINDROME TESTING
```

```
 11. for
 12. while
 13. do-while
 14. iterator

 0. Exit
1

Enter character to add:
c
Please select from the following:
 1. Add

FORWARD PROCESSING:
 2. for
 3. for-each
 4. while
 5. do-while
 6. iterator

REVERSE PROCESSING:
 7. for
 8. while
 9. do-while
 10. iterator

PALINDROME TESTING
 11. for
 12. while
 13. do-while
 14. iterator

 0. Exit
2
a
b
c
Please select from the following:
 1. Add

FORWARD PROCESSING:
 2. for
 3. for-each
 4. while
 5. do-while
 6. iterator

REVERSE PROCESSING:
 7. for
 8. while
 9. do-while
 10. iterator

PALINDROME TESTING
 11. for
 12. while
 13. do-while
 14. iterator

 0. Exit
3
a
b
c
Please select from the following:
 1. Add

FORWARD PROCESSING:
 2. for
 3. for-each
 4. while
 5. do-while
 6. iterator

REVERSE PROCESSING:
 7. for
 8. while
 9. do-while
 10. iterator

PALINDROME TESTING
 11. for
 12. while
 13. do-while
 14. iterator

 0. Exit
4
a
b
c
Please select from the following:
 1. Add

FORWARD PROCESSING:
 2. for
 3. for-each
 4. while
 5. do-while
 6. iterator

REVERSE PROCESSING:
 7. for
 8. while
 9. do-while
 10. iterator

PALINDROME TESTING
 11. for
 12. while
 13. do-while
 14. iterator

 0. Exit
5
a
b
c
Please select from the following:
 1. Add

FORWARD PROCESSING:
 2. for
 3. for-each
```

```
 4. while
 5. do-while
 6. iterator

REVERSE PROCESSING:
 7. for
 8. while
 9. do-while
 10. iterator

PALINDROME TESTING
 11. for
 12. while
 13. do-while
 14. iterator

 0. Exit
6
a
b
c
Please select from the following:
 1. Add

FORWARD PROCESSING:
 2. for
 3. for-each
 4. while
 5. do-while
 6. iterator

REVERSE PROCESSING:
 7. for
 8. while
 9. do-while
 10. iterator

PALINDROME TESTING
 11. for
 12. while
 13. do-while
 14. iterator

 0. Exit
7
c
b
a
Please select from the following:
 1. Add

FORWARD PROCESSING:
 2. for
 3. for-each
 4. while
 5. do-while
 6. iterator

REVERSE PROCESSING:
 7. for
 8. while
 9. do-while
 10. iterator

PALINDROME TESTING
 11. for
 12. while
 13. do-while
 14. iterator

 0. Exit
8
c
b
a
Please select from the following:
 1. Add

FORWARD PROCESSING:
 2. for
 3. for-each
 4. while
 5. do-while
 6. iterator

REVERSE PROCESSING:
 7. for
 8. while
 9. do-while
 10. iterator

PALINDROME TESTING
 11. for
 12. while
 13. do-while
 14. iterator

 0. Exit
9
c
b
a
Please select from the following:
 1. Add

FORWARD PROCESSING:
 2. for
 3. for-each
 4. while
 5. do-while
 6. iterator

REVERSE PROCESSING:
 7. for
 8. while
 9. do-while
 10. iterator

PALINDROME TESTING
 11. for
 12. while
 13. do-while
 14. iterator
```

```
 0. Exit
10
c
b
a
Please select from the following:
 1. Add

FORWARD PROCESSING:
 2. for
 3. for-each
 4. while
 5. do-while
 6. iterator

REVERSE PROCESSING:
 7. for
 8. while
 9. do-while
 10. iterator

PALINDROME TESTING
 11. for
 12. while
 13. do-while
 14. iterator

 0. Exit
0

Exiting! Farewell!
Please select from the menu!

Please select from the following:
 1. Add

FORWARD PROCESSING:
 2. for
 3. for-each
 4. while
 5. do-while
 6. iterator

REVERSE PROCESSING:
 7. for
 8. while
 9. do-while
 10. iterator

PALINDROME TESTING
 11. for
 12. while
 13. do-while
 14. iterator

 0. Exit
1

Enter character to add:
a
Please select from the following:
 1. Add

FORWARD PROCESSING:
 2. for
 3. for-each
 4. while
 5. do-while
 6. iterator

REVERSE PROCESSING:
 7. for
 8. while
 9. do-while
 10. iterator

PALINDROME TESTING
 11. for
 12. while
 13. do-while
 14. iterator

 0. Exit
1

Enter character to add:
b
Please select from the following:
 1. Add

FORWARD PROCESSING:
 2. for
 3. for-each
 4. while
 5. do-while
 6. iterator

REVERSE PROCESSING:
 7. for
 8. while
 9. do-while
 10. iterator

PALINDROME TESTING
 11. for
 12. while
 13. do-while
 14. iterator

 0. Exit
1

Enter character to add:
a
Please select from the following:
 1. Add

FORWARD PROCESSING:
 2. for
 3. for-each
 4. while
 5. do-while
 6. iterator
```

```
REVERSE PROCESSING:
 7. for
 8. while
 9. do-while
 10. iterator

PALINDROME TESTING
 11. for
 12. while
 13. do-while
 14. iterator

 0. Exit
11
For loop palindrome test came back: true
Please select from the following:
 1. Add

FORWARD PROCESSING:
 2. for
 3. for-each
 4. while
 5. do-while
 6. iterator

REVERSE PROCESSING:
 7. for
 8. while
 9. do-while
 10. iterator

PALINDROME TESTING
 11. for
 12. while
 13. do-while
 14. iterator

 0. Exit
12
While loop palindrome test came back: true
Please select from the following:
 1. Add

FORWARD PROCESSING:
 2. for
 3. for-each
 4. while
 5. do-while
 6. iterator

REVERSE PROCESSING:
 7. for
 8. while
 9. do-while
 10. iterator

PALINDROME TESTING
 11. for
 12. while
 13. do-while
 14. iterator

 0. Exit
13
Do While loop palindrome test came back: true
Please select from the following:
 1. Add

FORWARD PROCESSING:
 2. for
 3. for-each
 4. while
 5. do-while
 6. iterator

REVERSE PROCESSING:
 7. for
 8. while
 9. do-while
 10. iterator

PALINDROME TESTING
 11. for
 12. while
 13. do-while
 14. iterator

 0. Exit
14
Iterator palindrome test came back: true
Please select from the following:
 1. Add

FORWARD PROCESSING:
 2. for
 3. for-each
 4. while
 5. do-while
 6. iterator

REVERSE PROCESSING:
 7. for
 8. while
 9. do-while
 10. iterator

PALINDROME TESTING
 11. for
 12. while
 13. do-while
 14. iterator

 0. Exit
0

Exiting! Farewell!
Please select from the menu!

::::::::::::::
Lab1Conclusions.txt
::::::::::::::
Lab 1 caught me off guard initially because my Java was incredibly rough going int
o DSA but I feel like I gained a lot out of this.
```

Arrays and iterating over them isn't nearly as big of a headache as I remember the
m being, it gave me an excuse to build a simple UI which was a fun time.

Overall, it just reminded me that I *can* **do** programming **if** I sit down, shut up, a
nd actually put my mind to it.