



CANDIDATE NAME: Sean Riedl

Date: 05/05/2022

Are you looking for a new opportunity to have your work transform a business? Then look no further than Armedia

Please review the following questions and Provide All Requested Items of information for each question.

We understand that this may be a difficult evaluation test and take some time to complete.

We ask you provide answers to **all** questions. If you are struggling, complete as much as possible for the question. It is imperative we understand your ability and we will review them with you at your in-person interview.

We at Armedia are looking for exceptional talent with folks that take pride in their work and strive to go above and beyond to reach the next level of excellence; as they will have the opportunity to work with some of the finest minds in this industry on cutting edge technology.

Thank you for considering Armedia!
A great place to work to make a difference

1. Show the exact output produced by the following code segment.

```
char[][] pic = new char[6][6];
for (int i = 0; i < 6; i++)
    for (int j = 0; j < 6; j++) {
        if (i == j || i == 0 || i == 5)
            pic[i][j] = '*';
        else
            pic[i][j] = '.';
    }
for (int i = 0; i < 6; i++) {
    for (int j = 0; j < 6; j++)
        System.out.print(pic[i][j]);
    System.out.println();
}
```

The code needs to be enclosed within a class and method to run. Once done it outputs a "Z" in *'s

```
*****
*....
*...
*..
*...
*..
*...
*****
```

2. What letters get written to the standard output with the following code?

```
class anyClass {
    public static void main(String[] args) {
        try {
            clsMethod1();
        } catch (Exception e) {}
    }

    static void clsMethod1() {
        try {
            clsMethod2();

            System.out.println("a");

        } catch (ArithmeticException e) { System.out.println("b"); }
        Finally {System.out.println("c"); }

        System.out.println("d");
    }

    Static void clsMethod2() {
        Throw new NullPointerException();
    }
}
```

The output will be 'c'. This occurs because clsMethod2 throws an error sending the code to the catch block. The catch block will only execute its code section if the error thrown is an "ArithmeticException". Since the error thrown in the try section was a NullPointerException from clsMethod2, the catch block isn't executed. The finally block of code will always execute, printing 'c' to the console. The code outside of the try-catch-finally does not execute in this case because an unhandled error occurred.

3. What output would you get when you invoke the following main classes?

```
class anyClass1 {
    public static void main(String[] args) {
        String s1 = new String ("hello");
        String s2 = new String ("hello");

        If (s1.equals(s2)) System.out.println("equal");
        Else System.out.println("not equal");
    }
}

class anyClass2 {
    public static void main(String[] args) {
        StringBuffer sb1 = new StringBuffer ("hello");
        StringBuffer sb2 = new StringBuffer ("hello");

        If (sb1.equals(sb2)) System.out.println("equal");
        Else System.out.println("not equal");
    }
}
```

anyClass1 will output "equal" because .equals for Strings checks the contents of the string, not the object reference.

anyClass2 will output "not equal" because the .equals method for StringBuffer is not overridden from object instead it references equality (==).

Programming Tests – Please completed the Following Exercises

4. EX1

Write a complete subroutine that finds the largest value in an array of ints. The subroutine should have one parameter, which is an array of type int[]. The largest number in the array should be returned as the value of the subroutine.



5. EX2

```
import java.util.*;
```

```
public class BetterProgrammerTask {

    public static int[] retainPositiveNumbers(int[] a) {
        /*
         Please implement this method to
         return a new array with only positive numbers from the given array.
         The elements in the resulting array shall be sorted in the ascending order.
        */
    }
}
```



6. EX3

What output is produced by the following program segment? **Why?**

```
String name;
int i;
boolean startWord;

name = "Richard M. Nixon";
startWord = true;
for (i = 0; i < name.length(); i++) {
    if (startWord)
        System.out.println(name.charAt(i));
    if (name.charAt(i) == ' ')
        startWord = true;
    else
        startWord = false;
}
```



7. EX4

I have a customer who trades stocks and would like to create a model which gives him information on the optimal buying and selling points for a stock. For this exercise, we'll assume that we are just collecting and processing the data to determine the best model. Also, for this exercise, we'll assume that the data is from a single day of trading.

Please write a program which examines the data provided and provides the best (or best) solution(s) to maximize profit. Use the data provided below in the table:

Date	Time	Start
Oct 01 2018	9:30:00 AM	227.95
Oct 01 2018	9:40:00 AM	224.79
Oct 01 2018	9:50:00 AM	223.82
Oct 01 2018	10:00:00 AM	221
Oct 01 2018	10:10:00 AM	219.75
Oct 01 2018	10:20:00 AM	216.82
Oct 01 2018	10:30:00 AM	220.78
Oct 01 2018	10:40:00 AM	220.24
Oct 01 2018	10:50:00 AM	218.5
Oct 01 2018	11:00:00 AM	217.79
Oct 01 2018	11:10:00 AM	222.15
Oct 01 2018	11:20:00 AM	225.75
Oct 01 2018	11:30:00 AM	223.52
Oct 01 2018	11:40:00 AM	224.94
Oct 01 2018	11:50:00 AM	218.01
Oct 01 2018	12:00:00 PM	220.95
Oct 01 2018	12:10:00 PM	221.85
Oct 01 2018	12:20:00 PM	226.23
Oct 01 2018	12:30:00 PM	228.99
Oct 01 2018	12:40:00 PM	228.41
Oct 01 2018	12:50:00 PM	226.51
Oct 01 2018	1:00:00 PM	226.51
Oct 01 2018	1:10:00 PM	223.25
Oct 01 2018	1:20:00 PM	220.15
Oct 01 2018	1:30:00 PM	219.01
Oct 01 2018	1:40:00 PM	217.15
Oct 01 2018	1:50:00 PM	216.6
Oct 01 2018	2:00:00 PM	214.65
Oct 01 2018	2:10:00 PM	214.1
Oct 01 2018	2:20:00 PM	216.8
Oct 01 2018	2:30:00 PM	218.1
Oct 01 2018	2:40:00 PM	213.44
Oct 01 2018	2:50:00 PM	211.75
Oct 01 2018	3:00:00 PM	209.22
Oct 01 2018	3:10:00 PM	210.155
Oct 01 2018	3:20:00 PM	207.7
Oct 01 2018	3:30:00 PM	207.36
Oct 01 2018	3:40:00 PM	207.28
Oct 01 2018	3:50:00 PM	206.05
Oct 01 2018	4:00:00 PM	209.32

In this table you have the price of a stock at that time. For this model, let's only use the price at 10 minute intervals.

Develop an algorithm and print the best times to buy the stock and sell it to maximize the profit. Quantity

```
package ProblemSeven;
import java.io.*;
import java.text.NumberFormat;
import java.util.Scanner;
import java.util.ArrayList;
public class ProblemSeven {
    // Data type to store time and price as object.
    public class StockData {
        public String time;
        public Double price;
        public StockData(String time, Double price) {
            this.time = time;
            this.price = price;
        }
    }
    // Method to read data from the table of times and stock prices and create StockData objects.
    public static ArrayList<StockData> readData() throws Exception {
        String line = "";
        ArrayList<StockData> stockDataList = new ArrayList<StockData>();
        File file = new File("./ProblemSeven/stockData.csv");
        Scanner scanner = new Scanner(file);
        scanner.useDelimiter(",");
        while(scanner.hasNext()) {
            line = scanner.nextLine();
            String[] currentLine = line.split(",");
            String time = currentLine[3] + " " + currentLine[4];
            Double price = NumberFormat.getInstance().parse(currentLine[5]).doubleValue();
            StockData stockData = new ProblemSeven().new StockData(time, price);
            stockDataList.add(stockData);
        }
        scanner.close();
        return stockDataList;
    }
    public static StockData[] pickStocks(ArrayList<StockData> stockData) {
        int n1 = 0;
        int n2 = 1;
        Double maxProfit = 0.0;
        Double currentProfit = 0.0;
        StockData[] buyAndSell = new StockData[2];
        for(int i = 0; i < stockData.size(); i++) {
            n1 = i;
            currentProfit = stockData.get(n1).price - stockData.get(n2).price;
            if(currentProfit > maxProfit) {
                maxProfit = currentProfit;
                // Replace the first element in the array for when to buy
                buyAndSell[0] = stockData.get(n2);
                // Replace the second element in the array for when to sell
                buyAndSell[1] = stockData.get(n1);
            }
            if(stockData.get(n1).price < stockData.get(n2).price) {
                n2 = n1;
            }
        }
        System.out.println("Buy at " + buyAndSell[0].time + " for $" + buyAndSell[0].price);
        System.out.println("Sell at " + buyAndSell[1].time + " for $" + buyAndSell[1].price);
        return buyAndSell;
    }
    public static void main(String[] args) throws Exception {
        ArrayList<StockData> stockDataList = new ArrayList<StockData>();
```

8. EX5

Create a function with the following input parameters to perform the task defined

```
findMatch(int[] list, int total, int numToUse) {  
    /* list = list of integers  
       total = a single integer  
       numToUse = how many numbers from the list to use as a maximum
```

This function should print all combinations of values in 'list', when summed up return the value in 'total'

The value provided in 'numToUse' indicates the exact values to use from 'list' to calculate the 'total'. If the value is 2 then use exactly 2 elements from 'list'. If the value is 3, exactly 3 elements from 'list'. In all cases, there should be No Duplicate set reported.

For example:

list[1,5,6,3,6,7]

total = 13

numToUse = 2

Then the following should be printed:

Element 3 + Element 6 = 13

Element 5 + Element 6 = 13

If numToUse = 3 with the same inputs, then

Element 1 + Element 2 + Element 6 = 13

*/

}

Execute for the following inputs:

list = {3,4,6,7,10,3,9,15,17,17, -5, 10,7, -1, 21}

total = 20

numToUse = 2, 3, and 4

Please provide the output from the execution of the code.

```
package ProblemEight;  
public class ProblemEight {  
    public void findMatch(int[] list, int total, int numToUse) {  
  
    }  
}
```

9. EX. 6

We have a process where we receive data in a file which are required to process manually. We would like to automate this through the use of a program that will automatically read the data and process it, and then display the results to us in a report. **Please provide the output from the execution of the code.**

The format of the input file and the sample data to use is shown below.

```
14
*2 2
2 3
4 5
*2 2
3 11
11 3
*4 2
4 4
5 5
6 6
55 99
*3 3
3 5 a
4 7 a
3 4 b
*3 2
3 3
4 5 c a
3 5 b
*3 3
4 2 d
3 4 a
3 3 b
*3 3
4 2 a
3 3 a
4
*4 2
2 2
3 3
4 4
*2 3
-1 -2 a
-3 -3 b
*4 2
0 0
1 -1
-1 1
1111111 222222
*3 3
*2 3
*3 6
3 005 a
4 .007 a
3 4 b
7 2 e
8 4 ffffff
-1 -1 aaaaaaa
*2 2
a 0
b 01
```

The data is structured as follows.

- The first record in the file will indicate the number of data samples to read.
- The subsequent records will provide the specific data to process
 - The first record must contain two items that identify the data group to follow – this can be considered the control record – the items are separated by spaces
 - The first character of the control record must be a ‘*’
 - The first number indicates the number of data records in the data group
 - The second number indicates the number of elements in the data group
 - The next set of records will be the actual data that needs to be processed and will contain any number of elements separated by spaces. You only need to use the applicable values.
 - If there is a discrepancy in the data counts, the entire set of data is to be rejected and an error message printed indicated which data group the error is in.
 - If there is any error, generate an error message and reject the entire data group.
 - Blank lines should not be processed

You are to process the data as follows:

If the data contains two parameters, perform the following calculations

- Calculate the average of the two values and display with the following label “AVG = “
- Calculate the results of the following formula and display with the following label “FORM = “

Result = pi * pi * (first value) + psi * (2.44 + (second value)^{3/2}) * 3

- Calculate the average of the 'Result' calculated above from all the data points
- The results must be displayed to 3 significant digits – nnnn.000
- Display the total number of records reads

If the data contains three parameters

- Calculate the average of the three values and display with the following label "AVG 33 = "
- Calculate the results of the following formula and display with the following label "FORM 33 = "

Result = pi * pi * (first value) * pi + psi * (2.44 + (second value)^{3/2}) * 3 + ((zeta (lookup using the following mapping)) / (second value)) * log(lookup using the following mapping from third parameter of input data)

a → 3.21
b → 4.1
c → 6,8

- If a translation is not found, then use 'a' as the default.
- Calculate the average of the 'Result' calculated above from all the data points
- The results must be displayed to 3 significant digits – nnnn.000
- Display the total number of records reads

The constants pi,psi, and zeta are defined as follows:

Pi = 3.14
Psi = 6.84845
Zeta = 4.26

You are more than welcome to add additional data to validate other error and test conditions. All code and output should be sent back so it can be reviewed and verified. If there are any questions, please feel free to contact us.

**Some values are cut off in the text box due to the length of the output.

```
Reading Sample: 1
*2 2
[2.0, 3.0]
[4.0, 5.0]
[2.0, 3.0]
AVG = 2.500
FORM = 176.607
[4.0, 5.0]
AVG = 4.500
FORM = 319.273
The resulting average of the form value is 247.940
Reading Sample: 2
*2 2
[3.0, 11.0]
[11.0, 3.0]
[3.0, 11.0]
AVG = 7.000
FORM = 829.263
[11.0, 3.0]
AVG = 7.000
FORM = 265.343
The resulting average of the form value is 547.303
Reading Sample: 3
*4 2
[4.0, 4.0]
[5.0, 5.0]
[6.0, 6.0]
[55.0, 99.0]
[4.0, 4.0]
AVG = 4.000
FORM = 253.932
[5.0, 5.0]
AVG = 5.000
FORM = 329.133
[6.0, 6.0]
AVG = 6.000
FORM = 411.242
[55.0, 99.0]
AVG = 77.000
FORM = 20830.350
The resulting average of the form value is 5456.164
Reading Sample: 4
*3 3
[3.0, 5.0, 3.21]
[4.0, 7.0, 3.21]
[3.0, 4.0, 4.1]
[3.0, 5.0, 3.21]
AVG 33 = 3.737
FORM 33 = 397.160
[4.0, 7.0, 3.21]
AVG 33 = 4.737
FORM 33 = 571.935
[3.0, 4.0, 4.1]
AVG 33 = 3.700
FORM 33 = 441.660
The resulting average of the form value is 470.252
Reading Sample: 5
*3 2
[3.0, 3.0]
[4.0, 5.0, 6.8, 3.21]
[3.0, 5.0, 4.1]
```