

CMP-4008Y Coursework 1 - Electronic Component Inventory System

100246306 (wtn18vcu)

Tue, 3 Dec 2019 14:02

PDF prepared using PASS version 1.17 running on Windows 10 10.0 (amd64).

☒ I agree that by submitting a PDF generated by PASS I am confirming that I have checked the PDF and that it correctly represents my submission.



Contents

StockItem.java	2
Inventory.java	3
StockProgram.java	6
Application	7
Compiler Invocation	7
Compiler Messages	7
Application Invocation	7
Messages to STDOUT	7
Messages to STDERR	12

StockItem.java

```
1  package com.company.inventoryprogram;
   import java.util.*;
3
5
6  class StockItem {
7
8      private String componentType;
9      private String stockCode;
10     private Integer numberOfItemsInStock;
11     private Integer unitPriceInPence;
12     private String additionalInformation;
13
14
15     StockItem(String componentType, String stockCode, Integer
        numberOfItemsInStock, Integer unitPriceInPence, String
        additionalInformation) {
16
17         this.componentType = componentType;
18         this.stockCode = stockCode;
19         this.numberOfItemsInStock = numberOfItemsInStock;
20         this.unitPriceInPence = unitPriceInPence;
21         this.additionalInformation = additionalInformation;
22     }
23
24     String getComponentType() {
25         return componentType;
26     }
27
28     String getStockCode() {
29         return stockCode;
30     }
31
32     Integer getNumberOfItemsInStock() {
33         return numberOfItemsInStock;
34     }
35
36     Integer getUnitPriceInPence() {
37         return unitPriceInPence;
38     }
39
40     String getAdditionalInformation() {
41         return additionalInformation;
42     }
43
44
45     static class compareByPriceInPence implements Comparator<StockItem> {
46
47         public int compare(StockItem a, StockItem b) {
48             return a.unitPriceInPence - b.unitPriceInPence;
49         }
50     }
51
52     static class compareByLargestNumInStock implements Comparator<StockItem> {
53
54         public int compare(StockItem a, StockItem b) {
55             return a.numberOfItemsInStock - b.numberOfItemsInStock;
56         }
57     }
58 }
59 }
```

Inventory.java

```
1  package com.company.inventoryprogram;
   import java.io.*;
3  import java.math.BigDecimal;
   import java.text.DecimalFormat;
5  import java.util.*;

7  class Inventory {

9      private ArrayList<StockItem> units;

11     Inventory() {

13         units = new ArrayList<>();
14     }

15

17     void readData() {

19         try {

21             File fileName = new File("inventory.txt");
22             FileReader fReader = new FileReader(fileName); //Creation of File
23                 Reader object
24             BufferedReader bReader = new BufferedReader(fReader); //Creation of
25                 BufferedReader object
26             String line;

27             StockItem unit; //creates component with all the fields inside which
                are separated with comas.

29             while ((line = bReader.readLine()) != null) //Reading the content
                line by line
30             {

32                 ArrayList<String> lineList = new ArrayList<>(Arrays.asList(line.
33                     split(",")));

34                 if (lineList.size() < 5) {
35                     lineList.add(4, "lastValue");
36                 }

37                 String componentItemsSub = lineList.get(2);
38                 String componentItemsTrim = componentItemsSub.substring(1);

40                 String componentPriceSub = lineList.get(3);
41                 String componentPriceTrim = componentPriceSub.substring(1);

43                 String componentName = (lineList.get(0));
44                 String componentCode = (lineList.get(1));
45                 Integer componentItems = Integer.parseInt(componentItemsTrim); //
46                     Removes the space
47                 Integer componentPrice = Integer.parseInt(componentPriceTrim);
48                 String componentInformation = (lineList.get(4));

50                 unit = new StockItem(componentName, componentCode, componentItems
51                     , componentPrice, componentInformation);
52                 units.add(unit);
53             }
```

```

55         fReader.close(); //Close File Reader
    } catch (Exception e) {
57         System.out.println("error" + e.getMessage());
    }
59 }

61 //QUESTION 1

63 void sortingPrice() {

65     units.sort(new StockItem.compareByPriceInPence()); //Used collection sort
        method to sort price in pence by ascending order

67     for (StockItem unit : units) {
        //System.out.println(unit.getComponentType() + " " + unit.
            getStockCode() + " |" + unit.getNumberOfItemsInStock() + "|" +
            unit.getAdditionalInformation() + " " + " | Price In Pence |: " +
            unit.getUnitPriceInPence()); //prints the method of the object
69     System.out.println(unit.getComponentType() + "," + unit.
            getStockCode() + "," + unit.getNumberOfItemsInStock() + " , " +
            unit.getUnitPriceInPence() + "," + unit.getAdditionalInformation
            ()); //prints the method of the object

71     }
73 }

75 //QUESTION 2
void largestNumOfComponents() {

77     units.sort(Collections.reverseOrder(new StockItem.
        compareByLargestNumInStock())); //Used Collection.reverseOrder method
        to sort number of components in descending order

79     System.out.println("The component with the largest number of components
        in stock is : " + units.get(0).getComponentType() + " " + units.get(0)
        .getStockCode() + "| with : " + units.get(0).getNumberOfItemsInStock()
        +" Components");

81 }

83 //QUESTION 3
void npnTransistors() {

85     int total = 0;
87     for (StockItem unit : units) {

89         if (unit.getAdditionalInformation().substring(1).equals("NPN")) {

91             total = total + unit.getNumberOfItemsInStock();
            }
93     }
    System.out.println("Chartlins has | " + total + " | NPN transistors in
        stock ");

95 }

97 //QUESTION 4
void totalResistance() {

99     double grandTotal = 0;
101     for (StockItem unit : units) {

103         if (unit.getComponentType().equals("resistor")) {

```

```
105         double additionalInformationDouble = Double.parseDouble(unit.
            getAdditionalInformation().substring(1));
        double tot = unit.getNumberOfItemsInStock() *
            additionalInformationDouble;

107         grandTotal = grandTotal + tot;

109     }

111 }
BigDecimal bigDecimal = new BigDecimal(grandTotal); //Takes out
    scientific notation and puts back into big decimal

113 DecimalFormat decimalFormat = new DecimalFormat("0.00"); //Rounds the
    number into 2 decimal points
115 System.out.println("Total resistors of all resistors in stock is |: " +
    decimalFormat.format(bigDecimal) + " |rounded up to two decimal places
    "); //Adds the pattern
}

117 //QUESTION 5
119 void sortByPrice() {

121     int countNumber = 0;
    for (StockItem unit : units) {

123         if (unit.getUnitPriceInPence() > 10) {
            countNumber = countNumber + 1;

125         }

127     }
    System.out.println(" | "+ countNumber + " | " + " Stock items have unit
        price above 10p ");

129 }
}
```

StockProgram.java

```
package com.company.inventoryprogram;

2

4

public class StockProgram
6 {
    public static void main(String[] args) {
8        Inventory inv = new Inventory(); //an object of the inventory class
        inv.readData();

10        //Question 1 (Print a list of the inventory, sorted in order of
            increasing price.)
12        System.out.println("*****");
        System.out.println("* QUESTION NUMBER : 1 *");
14        System.out.println("*****");
        inv.sortingPrice();

16        //Question 2 (What is the component with the largest number of components
            in stock?)
18        System.out.println("*****");
        System.out.println("* QUESTION NUMBER : 2 *");
20        System.out.println("*****");
        inv.largestNumOfComponents();

22        //Question 3 (How many NPN transistors does Chartlins have in stock?)
24        System.out.println("*****");
        System.out.println("* QUESTION NUMBER : 3 *");
26        System.out.println("*****");
        inv.npnTransistors();

28        //Question 4 (What is the total resistance of all of the resistors in
            stock?)
30        System.out.println("*****");
        System.out.println("* QUESTION NUMBER : 4 *");
32        System.out.println("*****");
        inv.totalResistance();

34        //Question 5 ( How many stock items have unit prices above 10p? )
36        System.out.println("*****");
        System.out.println("* QUESTION NUMBER : 5 *");
38        System.out.println("*****");
        inv.sortByPrice();

40    }
42 }
```

Application

Compiler Invocation

```
javac.exe -Xlint:unchecked -Xlint:deprecation -d classes StockItem.java Inventory  
        .java StockProgram.java
```

Compiler Messages

None.

Application Invocation

```
javaw.exe com.company.inventoryprogram.StockProgram
```

Messages to STDOUT

* QUESTION NUMBER : 1 *

```
resistor, RES_1R0,41 , 1, 1  
resistor, RES_10R,467 , 1, 10  
resistor, RES_100R,334 , 1, 100  
resistor, RES_1K0,500 , 1, 1000  
resistor, RES_10K,169 , 1, 10000  
resistor, RES_100K,724 , 1, 100000  
resistor, RES_1M0,478 , 1, 1e+06  
resistor, RES_10M,358 , 1, 1e+07  
resistor, RES_1R1,962 , 1, 1.1  
resistor, RES_11R,464 , 1, 11  
resistor, RES_110R,705 , 1, 110  
resistor, RES_1K1,145 , 1, 1100  
resistor, RES_11K,281 , 1, 11000  
resistor, RES_110K,827 , 1, 110000  
resistor, RES_1M1,961 , 1, 1.1e+06  
resistor, RES_11M,491 , 1, 1.1e+07  
resistor, RES_1R2,995 , 1, 1.2  
resistor, RES_12R,942 , 1, 12  
resistor, RES_120R,827 , 1, 120  
resistor, RES_1K2,436 , 1, 1200  
resistor, RES_12K,391 , 1, 12000  
resistor, RES_120K,604 , 1, 120000  
resistor, RES_1M2,902 , 1, 1.2e+06  
resistor, RES_12M,153 , 1, 1.2e+07  
resistor, RES_1R3,292 , 1, 1.3  
resistor, RES_13R,382 , 1, 13  
resistor, RES_130R,421 , 1, 130  
resistor, RES_1K3,716 , 1, 1300  
resistor, RES_13K,718 , 1, 13000  
resistor, RES_130K,895 , 1, 130000  
resistor, RES_1M3,447 , 1, 1.3e+06  
resistor, RES_13M,726 , 1, 1.3e+07  
resistor, RES_1R5,771 , 1, 1.5  
resistor, RES_15R,538 , 1, 15  
resistor, RES_150R,869 , 1, 150  
resistor, RES_1K5,912 , 1, 1500  
resistor, RES_15K,667 , 1, 15000  
resistor, RES_150K,299 , 1, 150000  
resistor, RES_1M5,35 , 1, 1.5e+06  
resistor, RES_15M,894 , 1, 1.5e+07  
resistor, RES_1R6,703 , 1, 1.6  
resistor, RES_16R,811 , 1, 16  
resistor, RES_160R,322 , 1, 160  
resistor, RES_1K6,333 , 1, 1600
```

resistor, RES_16K,673 , 1, 16000
resistor, RES_160K,664 , 1, 160000
resistor, RES_1M6,141 , 1, 1.6e+06
resistor, RES_16M,711 , 1, 1.6e+07
resistor, RES_1R8,253 , 1, 1.8
resistor, RES_18R,868 , 1, 18
resistor, RES_180R,547 , 1, 180
resistor, RES_1K8,644 , 1, 1800
resistor, RES_18K,662 , 1, 18000
resistor, RES_180K,757 , 1, 180000
resistor, RES_1M8,37 , 1, 1.8e+06
resistor, RES_18M,859 , 1, 1.8e+07
resistor, RES_2R0,723 , 1, 2
resistor, RES_20R,741 , 1, 20
resistor, RES_200R,529 , 1, 200
resistor, RES_2K0,778 , 1, 2000
resistor, RES_20K,316 , 1, 20000
resistor, RES_200K,35 , 1, 200000
resistor, RES_2M0,190 , 1, 2e+06
resistor, RES_20M,842 , 1, 2e+07
resistor, RES_2R2,288 , 1, 2.2
resistor, RES_22R,106 , 1, 22
resistor, RES_220R,40 , 1, 220
resistor, RES_2K2,942 , 1, 2200
resistor, RES_22K,264 , 1, 22000
resistor, RES_220K,648 , 1, 220000
resistor, RES_2M2,446 , 1, 2.2e+06
resistor, RES_22M,805 , 1, 2.2e+07
resistor, RES_2R4,890 , 1, 2.4
resistor, RES_24R,729 , 1, 24
resistor, RES_240R,370 , 1, 240
resistor, RES_2K4,350 , 1, 2400
resistor, RES_24K,6 , 1, 24000
resistor, RES_240K,101 , 1, 240000
resistor, RES_2M4,393 , 1, 2.4e+06
resistor, RES_24M,548 , 1, 2.4e+07
resistor, RES_2R7,629 , 1, 2.7
resistor, RES_27R,623 , 1, 27
resistor, RES_270R,84 , 1, 270
resistor, RES_2K7,954 , 1, 2700
resistor, RES_27K,756 , 1, 27000
resistor, RES_270K,840 , 1, 270000
resistor, RES_2M7,966 , 1, 2.7e+06
resistor, RES_27M,376 , 1, 2.7e+07
resistor, RES_3R0,931 , 1, 3
resistor, RES_30R,308 , 1, 30
resistor, RES_300R,944 , 1, 300
resistor, RES_3K0,439 , 1, 3000
resistor, RES_30K,626 , 1, 30000
resistor, RES_300K,323 , 1, 300000
resistor, RES_3M0,537 , 1, 3e+06
resistor, RES_30M,538 , 1, 3e+07
resistor, RES_3R3,118 , 1, 3.3
resistor, RES_33R,82 , 1, 33
resistor, RES_330R,929 , 1, 330
resistor, RES_3K3,541 , 1, 3300
resistor, RES_33K,833 , 1, 33000
resistor, RES_330K,115 , 1, 330000
resistor, RES_3M3,639 , 1, 3.3e+06
resistor, RES_33M,658 , 1, 3.3e+07
resistor, RES_3R6,704 , 1, 3.6
resistor, RES_36R,930 , 1, 36
resistor, RES_360R,977 , 1, 360

resistor, RES_3K6,306 , 1, 3600
resistor, RES_36K,673 , 1, 36000
resistor, RES_360K,386 , 1, 360000
resistor, RES_3M6,21 , 1, 3.6e+06
resistor, RES_36M,745 , 1, 3.6e+07
resistor, RES_3R9,924 , 1, 3.9
resistor, RES_39R,72 , 1, 39
resistor, RES_390R,270 , 1, 390
resistor, RES_3K9,829 , 1, 3900
resistor, RES_39K,777 , 1, 39000
resistor, RES_390K,573 , 1, 390000
resistor, RES_3M9,97 , 1, 3.9e+06
resistor, RES_39M,512 , 1, 3.9e+07
resistor, RES_4R3,986 , 1, 4.3
resistor, RES_43R,290 , 1, 43
resistor, RES_430R,161 , 1, 430
resistor, RES_4K3,636 , 1, 4300
resistor, RES_43K,355 , 1, 43000
resistor, RES_430K,767 , 1, 430000
resistor, RES_4M3,655 , 1, 4.3e+06
resistor, RES_43M,574 , 1, 4.3e+07
resistor, RES_4R7,31 , 1, 4.7
resistor, RES_47R,52 , 1, 47
resistor, RES_470R,350 , 1, 470
resistor, RES_4K7,150 , 1, 4700
resistor, RES_47K,941 , 1, 47000
resistor, RES_470K,724 , 1, 470000
resistor, RES_4M7,966 , 1, 4.7e+06
resistor, RES_47M,430 , 1, 4.7e+07
resistor, RES_5R1,107 , 1, 5.1
resistor, RES_51R,191 , 1, 51
resistor, RES_510R,7 , 1, 510
resistor, RES_5K1,337 , 1, 5100
resistor, RES_51K,457 , 1, 51000
resistor, RES_510K,287 , 1, 510000
resistor, RES_5M1,753 , 1, 5.1e+06
resistor, RES_51M,383 , 1, 5.1e+07
resistor, RES_5R6,945 , 1, 5.6
resistor, RES_56R,909 , 1, 56
resistor, RES_560R,209 , 1, 560
resistor, RES_5K6,758 , 1, 5600
resistor, RES_56K,221 , 1, 56000
resistor, RES_560K,588 , 1, 560000
resistor, RES_5M6,422 , 1, 5.6e+06
resistor, RES_56M,946 , 1, 5.6e+07
resistor, RES_6R2,506 , 1, 6.2
resistor, RES_62R,30 , 1, 62
resistor, RES_620R,413 , 1, 620
resistor, RES_6K2,168 , 1, 6200
resistor, RES_62K,900 , 1, 62000
resistor, RES_620K,591 , 1, 620000
resistor, RES_6M2,762 , 1, 6.2e+06
resistor, RES_62M,655 , 1, 6.2e+07
resistor, RES_6R8,410 , 1, 6.8
resistor, RES_68R,359 , 1, 68
resistor, RES_680R,624 , 1, 680
resistor, RES_6K8,537 , 1, 6800
resistor, RES_68K,548 , 1, 68000
resistor, RES_680K,483 , 1, 680000
resistor, RES_6M8,595 , 1, 6.8e+06
resistor, RES_68M,41 , 1, 6.8e+07
resistor, RES_7R5,602 , 1, 7.5
resistor, RES_75R,350 , 1, 75

resistor, RES_750R,291 , 1, 750
resistor, RES_7K5,836 , 1, 7500
resistor, RES_75K,374 , 1, 75000
resistor, RES_750K,20 , 1, 750000
resistor, RES_7M5,596 , 1, 7.5e+06
resistor, RES_75M,21 , 1, 7.5e+07
resistor, RES_8R2,348 , 1, 8.2
resistor, RES_82R,199 , 1, 82
resistor, RES_820R,668 , 1, 820
resistor, RES_8K2,484 , 1, 8200
resistor, RES_82K,281 , 1, 82000
resistor, RES_820K,734 , 1, 820000
resistor, RES_8M2,53 , 1, 8.2e+06
resistor, RES_82M,999 , 1, 8.2e+07
resistor, RES_9R1,418 , 1, 9.1
resistor, RES_91R,938 , 1, 91
resistor, RES_910R,900 , 1, 910
resistor, RES_9K1,788 , 1, 9100
resistor, RES_91K,127 , 1, 91000
resistor, RES_910K,467 , 1, 910000
resistor, RES_9M1,728 , 1, 9.1e+06
resistor, RES_91M,893 , 1, 9.1e+07
capacitor, CAP_30pF,861 , 1, 0.03
capacitor, CAP_27pF,892 , 2, 0.027
capacitor, CAP_24pF,881 , 3, 0.024
capacitor, CAP_22pF,362 , 4, 0.022
diode, 1N4001,208 , 4,lastValue
capacitor, CAP_20pF,156 , 5, 0.02
diode, 1N4148,201 , 5,lastValue
capacitor, CAP_18pF,38 , 6, 0.018
diode, 1N4004,194 , 6,lastValue
capacitor, CAP_16pF,589 , 7, 0.016
capacitor, CAP_15pF,448 , 8, 0.015
capacitor, CAP_13pF,195 , 9, 0.013
capacitor, CAP_12pF,798 , 10, 0.012
transistor, BC107,20 , 10, NPN
capacitor, CAP_11pF,309 , 11, 0.011
capacitor, CAP_10pF,648 , 12, 0.01
diode, BY126,118 , 12,lastValue
diode, BY127,45 , 12,lastValue
transistor, BC108,9 , 12, NPN
IC, TL741,20 , 12, "op-amp"
capacitor, CAP_300pF,688 , 13, 0.3
capacitor, CAP_270pF,389 , 14, 0.27
capacitor, CAP_240pF,998 , 15, 0.24
capacitor, CAP_220pF,886 , 16, 0.22
capacitor, CAP_200pF,511 , 17, 0.2
IC, NE555,8 , 17, "Timer"
capacitor, CAP_180pF,179 , 18, 0.18
transistor, 2N2369A,37 , 18, NPN
capacitor, CAP_160pF,798 , 19, 0.16
capacitor, CAP_150pF,200 , 20, 0.15
capacitor, CAP_130pF,485 , 21, 0.13
capacitor, CAP_120pF,303 , 22, 0.12
capacitor, CAP_110pF,616 , 23, 0.11
capacitor, CAP_100pF,483 , 24, 0.1
capacitor, CAP_3000pf,401 , 25, 3
capacitor, CAP_2700pf,75 , 26, 2.7
capacitor, CAP_2400pf,322 , 27, 2.4
capacitor, CAP_2200pf,875 , 28, 2.2
capacitor, CAP_2000pf,202 , 29, 2
capacitor, CAP_1800pf,190 , 30, 1.8
capacitor, CAP_1600pf,9 , 31, 1.6

capacitor, CAP_1500pf,458 , 32, 1.5
 capacitor, CAP_1300pf,93 , 33, 1.3
 capacitor, CAP_1200pf,224 , 34, 1.2
 capacitor, CAP_1100pf,935 , 35, 1.1
 transistor, AC125,13 , 35, PNP
 capacitor, CAP_1000pf,807 , 36, 1
 capacitor, CAP_30nF,789 , 37, 30
 transistor, AC126,40 , 37, PNP
 capacitor, CAP_27nF,712 , 38, 27
 capacitor, CAP_24nF,651 , 39, 24
 capacitor, CAP_22nF,433 , 40, 22
 capacitor, CAP_20nF,634 , 41, 20
 capacitor, CAP_18nF,657 , 42, 18
 capacitor, CAP_16nF,157 , 43, 16
 capacitor, CAP_15nF,618 , 44, 15
 capacitor, CAP_13nF,343 , 45, 13
 IC, LF356,1 , 45, "JFET op-amp"
 capacitor, CAP_12nF,8 , 46, 12
 capacitor, CAP_11nF,451 , 47, 11
 capacitor, CAP_10nF,421 , 48, 10
 capacitor, CAP_300nF,255 , 49, 300
 capacitor, CAP_270nF,600 , 50, 270
 transistor, TIS88A,42 , 50, FET
 transistor, OC44,57 , 50, PNP
 capacitor, CAP_240nF,21 , 51, 240
 capacitor, CAP_220nF,869 , 52, 220
 capacitor, CAP_200nF,272 , 53, 200
 capacitor, CAP_180nF,958 , 54, 180
 capacitor, CAP_160nF,472 , 55, 160
 capacitor, CAP_150nF,580 , 56, 150
 capacitor, CAP_130nF,523 , 57, 130
 capacitor, CAP_120nF,844 , 58, 120
 capacitor, CAP_110nF,600 , 59, 110
 capacitor, CAP_100nF,310 , 60, 100
 capacitor, CAP_3000nF,423 , 61, 3000
 capacitor, CAP_2700nF,510 , 62, 2700
 capacitor, CAP_2400nF,699 , 63, 2400
 capacitor, CAP_2200nF,142 , 64, 2200
 capacitor, CAP_2000nF,55 , 65, 2000
 capacitor, CAP_1800nF,191 , 66, 1800
 capacitor, CAP_1600nF,622 , 67, 1600
 capacitor, CAP_1500nF,796 , 68, 1500
 capacitor, CAP_1300nF,587 , 69, 1300
 capacitor, CAP_1200nF,609 , 70, 1200
 capacitor, CAP_1100nF,249 , 71, 1100
 capacitor, CAP_1000nF,617 , 72, 1000
 capacitor, CAP_30uF,2 , 73, 30000
 capacitor, CAP_27uF,3 , 74, 27000
 capacitor, CAP_24uF,557 , 75, 24000
 capacitor, CAP_22uF,844 , 76, 22000
 capacitor, CAP_20uF,328 , 77, 20000
 capacitor, CAP_18uF,815 , 78, 18000
 capacitor, CAP_16uF,538 , 79, 16000
 capacitor, CAP_15uF,798 , 80, 15000
 capacitor, CAP_13uF,314 , 81, 13000
 capacitor, CAP_12uF,989 , 82, 12000
 capacitor, CAP_11uF,519 , 83, 11000
 capacitor, CAP_10uF,813 , 84, 10000
 capacitor, CAP_300uF,585 , 85, 300000
 capacitor, CAP_270uF,869 , 86, 270000
 capacitor, CAP_240uF,476 , 87, 240000
 capacitor, CAP_220uF,416 , 88, 220000
 capacitor, CAP_200uF,646 , 89, 200000

```

capacitor, CAP_180uF,888 , 90, 180000
capacitor, CAP_160uF,292 , 91, 160000
capacitor, CAP_150uF,281 , 92, 150000
capacitor, CAP_130uF,503 , 93, 130000
capacitor, CAP_120uF,702 , 94, 120000
capacitor, CAP_110uF,556 , 95, 110000
capacitor, CAP_100uF,514 , 96, 100000
IC, Z80A,0 , 800, "CPU"
*****
* QUESTION NUMBER : 2 *
*****
The component with the largest number of components in stock is : resistor RES_82M| with : 999 Components
*****
* QUESTION NUMBER : 3 *
*****
Chartlins has | 66 | NPN transistors in stock
*****
* QUESTION NUMBER : 4 *
*****
Total resistors of all resistors in stock is |: 579955286788.60 |rounded up to two decimal places
*****
* QUESTION NUMBER : 5 *
*****
| 98 | Stock items have unit price above 10p

```

Messages to STDERR

None.