

# CMP-4008Y Coursework 1 - Electronic Component Inventory System

100237847 (ccg18fdu)

Tue, 3 Dec 2019 12:08

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

<b>StockItem.java</b>	<b>2</b>
<b>Inventory.java</b>	<b>3</b>
<b>StockProgram.java</b>	<b>7</b>
<b>Application</b>	<b>10</b>
Compiler Invocation . . . . .	10
Compiler Messages . . . . .	10
Application Invocation . . . . .	10
Messages to STDOUT . . . . .	10
Messages to STDERR . . . . .	15

## StockItem.java

```
1 package chartlins;

3 class StockItem {
    //This class will be used to import the information from the inventory class and
    //sort the information.

5    //Fields

7    private String product;
9    private String code;
11   private String qty;
11   private String price;
11   private String sp;

13    //Constructor class to create item object

15    StockItem(String productName, String productCode, String quantity, String
        itemPrice, String special){

17        product = productName;
19        code = productCode;
19        qty = quantity;
21        price = itemPrice;
21        sp = special;

23    }

25    //Methods to return object values

27    String getProduct(){
29        return product;
31    }
31    String getCode(){
33        return code;
33    }
33    String getQty(){
35        return qty;
35    }
35    String getPrice(){
37        return price;
37    }
37    String getSpecial(){
39        return sp;
39    }
41    }
43 }
```

## Inventory.java

```

package chartlins;

import java.math.BigDecimal;
import java.util.*;
import java.util.stream.Collectors;

class Inventory {

    //ArrayList for storing the entire inventory

    private ArrayList<StockItem> invStock;

    Inventory(){
        invStock = new ArrayList<>();
    }

    void addItem(StockItem newItem){

        //Method for adding an Item to the inventory

        invStock.add(newItem);

        //Collections.sort()

    }

    Map<String, Integer> sortPriceHigh(){

        //Method for sorting the prices of the inventory by the highest price

        Map<String, Integer> happy = new HashMap<>();

        System.out.println(invStock.size() + " Items to sort by Price");

        for (StockItem item : invStock) {
            //Turn price into int

            String strItem = item.getPrice();

            String trimStrItem = strItem.substring(1);

            int intItem = Integer.parseInt(trimStrItem);

            //Added product name and price to hashmap to be sorted

            happy.put(item.getCode(), intItem);
        }

        //I can also do a collections sort now as I have learnt this, however this
        //was the first way I did it :)

        return happy.entrySet()
            //This is ugly but IntelliJ made me do it for optimisation reasons.
            .stream()
            .sorted(Map.Entry.comparingByValue())
            .collect(Collectors.toMap(Map.Entry::getKey, Map.Entry::getValue, (e1
                , e2) -> e1, LinkedHashMap::new));
    }
}

```

```
60 String[] largestNum(String selection){
62     Map<String, Integer> unSorted = new HashMap<>();
64     String[] returnArray = new String[2];

66     for (StockItem item : invStock) {
67         //This if statement could be expanded to include other selections
68
69         if (selection.equals("quantity")) {
70             String strQty = item.getQty();
72
73             String trimStrItem = strQty.substring(1);
74
75             int intItem = Integer.parseInt(trimStrItem);
76
77             //Added product name and price to hashmap to be sorted
78
79             unSorted.put(item.getCode(), intItem);
80
81         }

82         Map<String, Integer> sorted = unSorted.entrySet()
83             .stream()
84             .sorted(Map.Entry.comparingByValue())
85             .collect(Collectors.toMap(Map.Entry::getKey, Map.Entry::getValue,
86                 (e1, e2) -> e1, LinkedHashMap::new));
87
88         boolean count = false;
89         String returnKey = "";
90         int returnValue = 0;
91
92         while (!count) {
93
94             for (Map.Entry<String, Integer> entry : sorted.entrySet()) {
95                 returnKey = entry.getKey();
96                 returnValue = entry.getValue();
97
98                 count = true;
99             }
100         }

101         returnArray[0] = returnKey;
102         returnArray[1] = Integer.toString(returnValue);
103     }
104     return returnArray;
105 }

106
107 int specialQty(){
108
109     int NPNtotal = 0;
110
111
112
113
114     for (StockItem item : invStock) {
115         String strSp = item.getSpecial();
116
117         if (strSp.length() < 1) {
118             continue;
119         }
120
121         String trimStrItem = strSp.substring(1);
```

```
122         if (trimStrItem.equals("NPN")) {
124             String multi = item.getQty();
126             String multiTrim = multi.substring(1);
128             int multiInt = Integer.parseInt(multiTrim);
130             NPNtotal = NPNtotal + multiInt;
132         }
134     }
136     return NPNtotal;
137 }
138
139 BigDecimal totalRes(){
140     double returnRes = 0;
142     for (StockItem item : invStock) {
144         String strName = item.getProduct();
146         if (strName.equals("resistor")) {
148             String multi = item.getQty();
150             String multiTrim = multi.substring(1);
152             int multiInt = Integer.parseInt(multiTrim);
154             String currItem = item.getSpecial();
155             String trimCurrItem = currItem.substring(1);
156             BigDecimal bd = new BigDecimal(trimCurrItem);
157             double val = bd.doubleValue();
159             double multiVal = val * multiInt;
161
162             returnRes = returnRes + multiVal;
164         }
166     }
168     return new BigDecimal(returnRes);
169 }
170
171 ArrayList<String> sortPrice(int price){
172     ArrayList<String> returnList = new ArrayList<>();
174     for (StockItem item : invStock) {
176         String itemPrice = item.getPrice();
178         String itemPriceSub = itemPrice.substring(1);
180         int itemPriceInt = Integer.parseInt(itemPriceSub);
182         if (itemPriceInt > price) {
184             String itemName = item.getCode();
```

```
186         returnList.add(itemName);  
188     }  
190     return returnList;  
192 }  
194 }
```

## StockProgram.java

```

1  /* *
   *
3  *   100237847
   *   5/12/2019
5  *   Program to parse file inventory.txt and answer 5 questions relating to the
   *   inventory.
   *
7  * */

9  package chartlins;

11 import java.io.File;
    import java.io.FileNotFoundException;
13 import java.math.BigDecimal;
    import java.text.DecimalFormat;
15 import java.util.*;

17 public class StockProgram {
    public static void main(String[] args) throws Exception {

19         System.out.println("Parsing inventory.txt to create Inventory");

21         Inventory inv = new Inventory();

23         parseFile(inv);

25         //Question 1

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

31         // Print a list of the inventory, sorted in order of increasing price.

33         Map<String, Integer> sortedMap;

35         sortedMap = inv.sortPriceHigh();

37         for(Map.Entry<String, Integer> entry : sortedMap.entrySet()){
39             System.out.println("Product Code : " + entry.getKey() + ", Price in pence
                : " + entry.getValue());
        }

41         //Question 2

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

47         // What is the component with the largest number of components in stock

49         String selection;

51         selection = "quantity";

53         String[] q2Arr = (inv.largestNum(selection));

55         System.out.println("Product with the highest " + selection + " is : ");

57         System.out.println("Product : " + q2Arr[0]);
        System.out.println(selection + " : " + q2Arr[1]);
59

```

```
61      //Question 3

63      System.out.println("-----");
64      System.out.println("Question 3");
65      System.out.println("-----");

67      // How many NPN transistors does Chartlins have in stock?

69      int q3Num;

71      q3Num = inv.specialQty();

73      System.out.println("The number of NPN transistors in stock is : " + q3Num);

75      //Question 4

77      System.out.println("-----");
78      System.out.println("Question 4");
79      System.out.println("-----");

81      // What is the total resistance of all the resistors in stock?

83      BigDecimal q4Num;

85      DecimalFormat df = new DecimalFormat("0.0000");

87      q4Num = inv.totalRes();

89      //Should have a unicode ohms character at the end of this statement

91      System.out.println("The total resistance of all of the resistors in stock is
      : " + df.format(q4Num) + " \u2126");

93      //Question 5

95      System.out.println("-----");
96      System.out.println("Question 5");
97      System.out.println("-----");

99      // How many stock items have unit prices above 10p?

101     ArrayList<String> q5List;

103     int price = 10;

105     q5List = inv.sortPrice(price);

107     System.out.println(q5List.size() + " Items have a price above " + price + "
      pence");
109 }

111 private static void parseFile(Inventory inv) throws FileNotFoundException {

113     //Method for parsing inventory.txt and retrieving the data in an and adding
      all items to the inventory

115     File file = new File("inventory.txt");

117     Scanner scanner = new Scanner(file);

119     //Check for next line in inventory.txt
```



```
121     while(scanner.hasNextLine()){
122         String line = scanner.nextLine();
123
124         ArrayList<String> lineInfo = new ArrayList<>(Arrays.asList(line.split(", "
125             )))
126
127         //Check to see if line has a minimum of 5 elements
128
129         if(lineInfo.size() < 5){
130
131             lineInfo.add(4, "");
132
133         }
134
135         //Assign elements of line to variables
136
137         //Product Name [0]
138         String name = lineInfo.get(0);
139         //Product Code [1]
140         String code = lineInfo.get(1);
141         //Product Quantity [2]
142         String qty = lineInfo.get(2);
143         //Product Price [3]
144         String price = lineInfo.get(3);
145         //Product Special Value [4]
146         String sp = lineInfo.get(4);
147
148         //Construct new object
149
150         StockItem stockItem = new StockItem(name, code, qty, price, sp);
151
152         //Add item to inventory
153
154         inv.addItem(stockItem);
155     }
156
157     System.out.println("All Items imported from inventory.txt");
158
159 }
160
161 }
```

## Application

### Compiler Invocation

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

### Compiler Messages

None.

### Application Invocation

```
javaw.exe chartlins.StockProgram
```

### Messages to STDOUT

```
Parsing inventory.txt to create Inventory  
All Items imported from inventory.txt
```

```
-----  
Question 1
```

```
-----  
304 Items to sort by Price  
Product Code : RES_910K, Price in pence : 1  
Product Code : RES_3K9, Price in pence : 1  
Product Code : RES_300K, Price in pence : 1  
Product Code : RES_16K, Price in pence : 1  
Product Code : RES_16M, Price in pence : 1  
Product Code : RES_910R, Price in pence : 1  
Product Code : RES_150K, Price in pence : 1  
Product Code : RES_16R, Price in pence : 1  
Product Code : RES_300R, Price in pence : 1  
Product Code : RES_270R, Price in pence : 1  
Product Code : RES_6R2, Price in pence : 1  
Product Code : RES_6R8, Price in pence : 1  
Product Code : RES_39K, Price in pence : 1  
Product Code : RES_27K, Price in pence : 1  
Product Code : RES_15K, Price in pence : 1  
Product Code : RES_39M, Price in pence : 1  
Product Code : RES_27M, Price in pence : 1  
Product Code : RES_15M, Price in pence : 1  
Product Code : RES_39R, Price in pence : 1  
Product Code : RES_27R, Price in pence : 1  
Product Code : RES_15R, Price in pence : 1  
Product Code : RES_270K, Price in pence : 1  
Product Code : RES_390K, Price in pence : 1  
Product Code : RES_160R, Price in pence : 1  
Product Code : RES_390R, Price in pence : 1  
Product Code : RES_3K0, Price in pence : 1  
Product Code : RES_4M3, Price in pence : 1  
Product Code : RES_3K3, Price in pence : 1  
Product Code : RES_3K6, Price in pence : 1  
Product Code : RES_4M7, Price in pence : 1  
Product Code : RES_4K7, Price in pence : 1  
Product Code : RES_750K, Price in pence : 1  
Product Code : RES_750R, Price in pence : 1  
Product Code : RES_82K, Price in pence : 1  
Product Code : RES_82M, Price in pence : 1  
Product Code : RES_82R, Price in pence : 1  
Product Code : RES_7R5, Price in pence : 1  
Product Code : RES_13K, Price in pence : 1  
Product Code : RES_13M, Price in pence : 1  
Product Code : RES_13R, Price in pence : 1  
Product Code : RES_150R, Price in pence : 1
```

Product Code : RES\_5M1, Price in pence : 1  
Product Code : RES\_4K3, Price in pence : 1  
Product Code : RES\_5M6, Price in pence : 1  
Product Code : RES\_9K1, Price in pence : 1  
Product Code : RES\_4R3, Price in pence : 1  
Product Code : RES\_4R7, Price in pence : 1  
Product Code : RES\_820K, Price in pence : 1  
Product Code : RES\_820R, Price in pence : 1  
Product Code : RES\_75K, Price in pence : 1  
Product Code : RES\_100R, Price in pence : 1  
Product Code : RES\_330R, Price in pence : 1  
Product Code : RES\_51K, Price in pence : 1  
Product Code : RES\_560R, Price in pence : 1  
Product Code : RES\_75M, Price in pence : 1  
Product Code : RES\_180K, Price in pence : 1  
Product Code : RES\_51M, Price in pence : 1  
Product Code : RES\_75R, Price in pence : 1  
Product Code : RES\_2M0, Price in pence : 1  
Product Code : RES\_1K0, Price in pence : 1  
Product Code : RES\_51R, Price in pence : 1  
Product Code : RES\_1K1, Price in pence : 1  
Product Code : RES\_180R, Price in pence : 1  
Product Code : RES\_2M2, Price in pence : 1  
Product Code : RES\_1K2, Price in pence : 1  
Product Code : RES\_1K3, Price in pence : 1  
Product Code : RES\_2M4, Price in pence : 1  
Product Code : RES\_1K5, Price in pence : 1  
Product Code : RES\_1K6, Price in pence : 1  
Product Code : RES\_2M7, Price in pence : 1  
Product Code : RES\_1K8, Price in pence : 1  
Product Code : RES\_3M9, Price in pence : 1  
Product Code : RES\_18K, Price in pence : 1  
Product Code : RES\_18M, Price in pence : 1  
Product Code : RES\_18R, Price in pence : 1  
Product Code : RES\_160K, Price in pence : 1  
Product Code : RES\_62K, Price in pence : 1  
Product Code : RES\_430R, Price in pence : 1  
Product Code : RES\_62M, Price in pence : 1  
Product Code : RES\_5R1, Price in pence : 1  
Product Code : RES\_62R, Price in pence : 1  
Product Code : RES\_5R6, Price in pence : 1  
Product Code : RES\_200K, Price in pence : 1  
Product Code : RES\_430K, Price in pence : 1  
Product Code : RES\_200R, Price in pence : 1  
Product Code : RES\_3M0, Price in pence : 1  
Product Code : RES\_2K0, Price in pence : 1  
Product Code : RES\_2K2, Price in pence : 1  
Product Code : RES\_3M3, Price in pence : 1  
Product Code : RES\_2K4, Price in pence : 1  
Product Code : RES\_3M6, Price in pence : 1  
Product Code : RES\_2K7, Price in pence : 1  
Product Code : RES\_7K5, Price in pence : 1  
Product Code : RES\_110K, Price in pence : 1  
Product Code : RES\_68K, Price in pence : 1  
Product Code : RES\_56K, Price in pence : 1  
Product Code : RES\_68M, Price in pence : 1  
Product Code : RES\_56M, Price in pence : 1  
Product Code : RES\_20K, Price in pence : 1  
Product Code : RES\_20M, Price in pence : 1  
Product Code : RES\_2R0, Price in pence : 1  
Product Code : RES\_68R, Price in pence : 1  
Product Code : RES\_56R, Price in pence : 1  
Product Code : RES\_2R2, Price in pence : 1

Product Code : RES\_20R, Price in pence : 1  
Product Code : RES\_2R4, Price in pence : 1  
Product Code : RES\_2R7, Price in pence : 1  
Product Code : RES\_43K, Price in pence : 1  
Product Code : RES\_43M, Price in pence : 1  
Product Code : RES\_120R, Price in pence : 1  
Product Code : RES\_43R, Price in pence : 1  
Product Code : RES\_8M2, Price in pence : 1  
Product Code : RES\_8K2, Price in pence : 1  
Product Code : CAP\_30pF, Price in pence : 1  
Product Code : RES\_100K, Price in pence : 1  
Product Code : RES\_560K, Price in pence : 1  
Product Code : RES\_330K, Price in pence : 1  
Product Code : RES\_30K, Price in pence : 1  
Product Code : RES\_220R, Price in pence : 1  
Product Code : RES\_30M, Price in pence : 1  
Product Code : RES\_680R, Price in pence : 1  
Product Code : RES\_3R0, Price in pence : 1  
Product Code : RES\_3R3, Price in pence : 1  
Product Code : RES\_30R, Price in pence : 1  
Product Code : RES\_3R6, Price in pence : 1  
Product Code : RES\_3R9, Price in pence : 1  
Product Code : RES\_220K, Price in pence : 1  
Product Code : RES\_680K, Price in pence : 1  
Product Code : RES\_110R, Price in pence : 1  
Product Code : RES\_1M0, Price in pence : 1  
Product Code : RES\_1M1, Price in pence : 1  
Product Code : RES\_1M2, Price in pence : 1  
Product Code : RES\_1M3, Price in pence : 1  
Product Code : RES\_1M5, Price in pence : 1  
Product Code : RES\_1M6, Price in pence : 1  
Product Code : RES\_1M8, Price in pence : 1  
Product Code : RES\_9M1, Price in pence : 1  
Product Code : RES\_5K6, Price in pence : 1  
Product Code : RES\_6M8, Price in pence : 1  
Product Code : RES\_510K, Price in pence : 1  
Product Code : RES\_36K, Price in pence : 1  
Product Code : RES\_24K, Price in pence : 1  
Product Code : RES\_12K, Price in pence : 1  
Product Code : RES\_36M, Price in pence : 1  
Product Code : RES\_130K, Price in pence : 1  
Product Code : RES\_24M, Price in pence : 1  
Product Code : RES\_360K, Price in pence : 1  
Product Code : RES\_12M, Price in pence : 1  
Product Code : RES\_510R, Price in pence : 1  
Product Code : RES\_36R, Price in pence : 1  
Product Code : RES\_24R, Price in pence : 1  
Product Code : RES\_12R, Price in pence : 1  
Product Code : RES\_8R2, Price in pence : 1  
Product Code : RES\_47K, Price in pence : 1  
Product Code : RES\_47M, Price in pence : 1  
Product Code : RES\_11K, Price in pence : 1  
Product Code : RES\_11M, Price in pence : 1  
Product Code : RES\_47R, Price in pence : 1  
Product Code : RES\_11R, Price in pence : 1  
Product Code : RES\_91K, Price in pence : 1  
Product Code : RES\_91M, Price in pence : 1  
Product Code : RES\_91R, Price in pence : 1  
Product Code : RES\_5K1, Price in pence : 1  
Product Code : RES\_6M2, Price in pence : 1  
Product Code : RES\_7M5, Price in pence : 1  
Product Code : RES\_6K8, Price in pence : 1  
Product Code : RES\_120K, Price in pence : 1

Product Code : RES\_22K, Price in pence : 1  
Product Code : RES\_10K, Price in pence : 1  
Product Code : RES\_22M, Price in pence : 1  
Product Code : RES\_10M, Price in pence : 1  
Product Code : RES\_1R0, Price in pence : 1  
Product Code : RES\_1R1, Price in pence : 1  
Product Code : RES\_1R2, Price in pence : 1  
Product Code : RES\_1R3, Price in pence : 1  
Product Code : RES\_22R, Price in pence : 1  
Product Code : RES\_240R, Price in pence : 1  
Product Code : RES\_10R, Price in pence : 1  
Product Code : RES\_1R5, Price in pence : 1  
Product Code : RES\_470R, Price in pence : 1  
Product Code : RES\_1R6, Price in pence : 1  
Product Code : RES\_1R8, Price in pence : 1  
Product Code : RES\_9R1, Price in pence : 1  
Product Code : RES\_620K, Price in pence : 1  
Product Code : RES\_240K, Price in pence : 1  
Product Code : RES\_33K, Price in pence : 1  
Product Code : RES\_470K, Price in pence : 1  
Product Code : RES\_33M, Price in pence : 1  
Product Code : RES\_620R, Price in pence : 1  
Product Code : RES\_130R, Price in pence : 1  
Product Code : RES\_33R, Price in pence : 1  
Product Code : RES\_360R, Price in pence : 1  
Product Code : RES\_6K2, Price in pence : 1  
Product Code : CAP\_27pF, Price in pence : 2  
Product Code : CAP\_24pF, Price in pence : 3  
Product Code : CAP\_22pF, Price in pence : 4  
Product Code : 1N4001, Price in pence : 4  
Product Code : 1N4148, Price in pence : 5  
Product Code : CAP\_20pF, Price in pence : 5  
Product Code : CAP\_18pF, Price in pence : 6  
Product Code : 1N4004, Price in pence : 6  
Product Code : CAP\_16pF, Price in pence : 7  
Product Code : CAP\_15pF, Price in pence : 8  
Product Code : CAP\_13pF, Price in pence : 9  
Product Code : CAP\_12pF, Price in pence : 10  
Product Code : BC107, Price in pence : 10  
Product Code : CAP\_11pF, Price in pence : 11  
Product Code : TL741, Price in pence : 12  
Product Code : CAP\_10pF, Price in pence : 12  
Product Code : BY126, Price in pence : 12  
Product Code : BY127, Price in pence : 12  
Product Code : BC108, Price in pence : 12  
Product Code : CAP\_300pF, Price in pence : 13  
Product Code : CAP\_270pF, Price in pence : 14  
Product Code : CAP\_240pF, Price in pence : 15  
Product Code : CAP\_220pF, Price in pence : 16  
Product Code : CAP\_200pF, Price in pence : 17  
Product Code : NE555, Price in pence : 17  
Product Code : 2N2369A, Price in pence : 18  
Product Code : CAP\_180pF, Price in pence : 18  
Product Code : CAP\_160pF, Price in pence : 19  
Product Code : CAP\_150pF, Price in pence : 20  
Product Code : CAP\_130pF, Price in pence : 21  
Product Code : CAP\_120pF, Price in pence : 22  
Product Code : CAP\_110pF, Price in pence : 23  
Product Code : CAP\_100pF, Price in pence : 24  
Product Code : CAP\_3000pf, Price in pence : 25  
Product Code : CAP\_2700pf, Price in pence : 26  
Product Code : CAP\_2400pf, Price in pence : 27  
Product Code : CAP\_2200pf, Price in pence : 28

Product Code : CAP\_2000pf, Price in pence : 29  
Product Code : CAP\_1800pf, Price in pence : 30  
Product Code : CAP\_1600pf, Price in pence : 31  
Product Code : CAP\_1500pf, Price in pence : 32  
Product Code : CAP\_1300pf, Price in pence : 33  
Product Code : CAP\_1200pf, Price in pence : 34  
Product Code : CAP\_1100pf, Price in pence : 35  
Product Code : AC125, Price in pence : 35  
Product Code : CAP\_1000pf, Price in pence : 36  
Product Code : AC126, Price in pence : 37  
Product Code : CAP\_30nF, Price in pence : 37  
Product Code : CAP\_27nF, Price in pence : 38  
Product Code : CAP\_24nF, Price in pence : 39  
Product Code : CAP\_22nF, Price in pence : 40  
Product Code : CAP\_20nF, Price in pence : 41  
Product Code : CAP\_18nF, Price in pence : 42  
Product Code : CAP\_16nF, Price in pence : 43  
Product Code : CAP\_15nF, Price in pence : 44  
Product Code : LF356, Price in pence : 45  
Product Code : CAP\_13nF, Price in pence : 45  
Product Code : CAP\_12nF, Price in pence : 46  
Product Code : CAP\_11nF, Price in pence : 47  
Product Code : CAP\_10nF, Price in pence : 48  
Product Code : CAP\_300nF, Price in pence : 49  
Product Code : CAP\_270nF, Price in pence : 50  
Product Code : TIS88A, Price in pence : 50  
Product Code : OC44, Price in pence : 50  
Product Code : CAP\_240nF, Price in pence : 51  
Product Code : CAP\_220nF, Price in pence : 52  
Product Code : CAP\_200nF, Price in pence : 53  
Product Code : CAP\_180nF, Price in pence : 54  
Product Code : CAP\_160nF, Price in pence : 55  
Product Code : CAP\_150nF, Price in pence : 56  
Product Code : CAP\_130nF, Price in pence : 57  
Product Code : CAP\_120nF, Price in pence : 58  
Product Code : CAP\_110nF, Price in pence : 59  
Product Code : CAP\_100nF, Price in pence : 60  
Product Code : CAP\_3000nF, Price in pence : 61  
Product Code : CAP\_2700nF, Price in pence : 62  
Product Code : CAP\_2400nF, Price in pence : 63  
Product Code : CAP\_2200nF, Price in pence : 64  
Product Code : CAP\_2000nF, Price in pence : 65  
Product Code : CAP\_1800nF, Price in pence : 66  
Product Code : CAP\_1600nF, Price in pence : 67  
Product Code : CAP\_1500nF, Price in pence : 68  
Product Code : CAP\_1300nF, Price in pence : 69  
Product Code : CAP\_1200nF, Price in pence : 70  
Product Code : CAP\_1100nF, Price in pence : 71  
Product Code : CAP\_1000nF, Price in pence : 72  
Product Code : CAP\_30uF, Price in pence : 73  
Product Code : CAP\_27uF, Price in pence : 74  
Product Code : CAP\_24uF, Price in pence : 75  
Product Code : CAP\_22uF, Price in pence : 76  
Product Code : CAP\_20uF, Price in pence : 77  
Product Code : CAP\_18uF, Price in pence : 78  
Product Code : CAP\_16uF, Price in pence : 79  
Product Code : CAP\_15uF, Price in pence : 80  
Product Code : CAP\_13uF, Price in pence : 81  
Product Code : CAP\_12uF, Price in pence : 82  
Product Code : CAP\_11uF, Price in pence : 83  
Product Code : CAP\_10uF, Price in pence : 84  
Product Code : CAP\_300uF, Price in pence : 85  
Product Code : CAP\_270uF, Price in pence : 86

Product Code : CAP\_240uF, Price in pence : 87  
Product Code : CAP\_220uF, Price in pence : 88  
Product Code : CAP\_200uF, Price in pence : 89  
Product Code : CAP\_180uF, Price in pence : 90  
Product Code : CAP\_160uF, Price in pence : 91  
Product Code : CAP\_150uF, Price in pence : 92  
Product Code : CAP\_130uF, Price in pence : 93  
Product Code : CAP\_120uF, Price in pence : 94  
Product Code : CAP\_110uF, Price in pence : 95  
Product Code : CAP\_100uF, Price in pence : 96  
Product Code : Z80A, Price in pence : 800  
-----

#### Question 2

-----

Product with the highest quantity is :  
Product : RES\_82M  
quantity : 999  
-----

#### Question 3

-----

The number of NPN transistors in stock is : 66  
-----

#### Question 4

-----

The total resistance of all of the resistors in stock is : 579955286788.6000 ?  
-----

#### Question 5

-----

98 Items have a price above 10 pence

### Messages to STDERR

None.