CLASS STOCK

```java
package stk;

public class Stock {
    private String identity;
    private String name;
    private int quantity;
    private double price;
    private String manufacturingDate;
    private String expiryDate;
    public Stock(String identity, String name, int quantity, double price,
            String manufacturingDate, String expiryDate) {
        super();
        this.identity = identity;
        this.name = name;
        this.quantity = quantity;
        this.price = price;
        this.manufacturingDate = manufacturingDate;
        this.expiryDate = expiryDate;
    }

    public String getIdentity() {
        return identity;
    }
    public void setIdentity(String identity) {
        this.identity = identity;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getQuantity() {
        return quantity;
    }
    public void setQuantity(int quantity) {
        this.quantity = quantity;
    }
    public double getPrice() {
        return price;
    }
    public void setPrice(double price) {
        this.price = price;
    }
    public String getManufacturingDate() {
        return manufacturingDate;
    }
    public void setManufacturingDate(String manufacturingDate) {
        this.manufacturingDate = manufacturingDate;
    }
    public String getExpiryDate() {
        return expiryDate;
    }
```

```
	public void setExpiryDate(String expiryDate) {
		this.expiryDate = expiryDate;
	}

	 public String toString() {
		 return String.format ("%-10s %-20s %-3d %5.2f %-10s %-10s",identity,name,quantity,price,manufacturingDate,expiryDate);
	    }



}
```

CLASS SOLUTION

```
package stk;


import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
//import java.util.Collections;
import java.util.LinkedList;


import stk.ExpiryDateComparator;
import stk.IdentityComparator;
import stk.ManufacturingDateComparator;
import stk.NameComparator;
import stk.PriceComparator;
import stk.QuantityComparator;
import stk.Stock;


public class Solution {
        public static void main(String[] args) throws IOException {
                // TODO Auto-generated method stub


                LinkedList <Stock> s1 = new LinkedList<Stock>();
                BufferedReader bf = new BufferedReader(new InputStreamReader(System.in));
                String ch;
Stock sObj;


                System.out.println("GIVE UR PRODUCT DETAILS");


    do
    {


       String identity= bf.readLine();
       String name =bf.readLine();
       int quantity=Integer.parseInt(bf.readLine());
       double price = Double.parseDouble(bf.readLine());
       String manufacturingDate = bf.readLine();
       String expiryDate = bf.readLine();
       sObj=new Stock(identity,name,quantity,price,manufacturingDate,expiryDate);
       s1.add(sObj);
       System.out.println("DO U WANT TO ADD MORE???");
       ch=bf.readLine();
    }while(!ch.equals("NO"));
    for(Stock each:s1){
       System.out.println(each);

    }



                System.out.println("1.IDENTITY COMPARATOR ASCENDING\n2.IDENTITY COMPARATOR DESCENDING\n3.NAME COMPARATOR ASCENDING\n4.NAME COMPARATOR DESCENDING\n5.QUANTITY COMPARATOR ASCENDING6.QUANTITY COMPARATOR DESCENDING\n7.PRICE COMPARATOR ASCENDING\n8.PRICE COMPARATOR DESCENDING\n9.MANUFACTURING DATE COMPARATOR ASCENDING\n10.MANUFATURING DATE COMPARATOR DESCENDING\n11.EXPIRY DATE COMPARATOR ASCENDING\n12.EXPIRY DATE COMPARATOR DESCENDING");

                int choice = Integer.parseInt(bf.readLine());

                switch(choice) {

                case 1:



                    s1.sort(new IdentityComparator());

                    for(Stock S:s1)

              System.out.println(S);

                    break;





                case 2:

                    s1.sort(new NameComparator());

                    for(Stock S:s1)

                    System.out.println(S);

                    break;



                case 3:

                    s1.sort(new QuantityComparator());
```

```
        for(Stock S:s1)
        System.out.println(S);
        break;

case 4:
        s1.sort(new PriceComparator());
        for(Stock S:s1)
        System.out.println(S);
        break;

case 5:
        s1.sort(new ManufacturingDateComparator());
        for(Stock S:s1)
        System.out.println(S);
        break;

case 6:
        s1.sort(new ExpiryDateComparator());
        for(Stock S:s1)
        System.out.println(S);
        break;

case 7:
        s1.sort(new IdentityComparatorDescending());
        for(Stock S:s1)
```

```
System.out.println(S);
            break;

    case 8:
            s1.sort(new NameComparatorDescending());
            for(Stock S:s1)
            System.out.println(S);
            break;

    case 9:
            s1.sort(new QuantityComparatorDescending());
            for(Stock S:s1)
            System.out.println(S);
            break;

    case 10:
            s1.sort(new PriceComparatorDescending());
            for(Stock S:s1)
            System.out.println(S);
            break;

    case 11:
            s1.sort(new ManufacturingDateComparatorDescending());
            for(Stock S:s1)
            System.out.println(S);
            break;

        case 12:
            s1.sort(new ExpiryDateComparatorDescending());
            for(Stock S:s1)
            System.out.println(S);
            break;

        default:System.out.println("INVALID SELECTION");


        }
    }

}
```

CLASSES USED FOR SORTING IN ASCENDING & DESCENDING

1.

```
package stk;

import java.util.Comparator;

public class ExpiryDateComparator implements Comparator <Stock> {

    public int compare(Stock o1, Stock o2) {
                return o1.getExpiryDate().compareTo(o2.getExpiryDate());

        }

}
```

2.

```
package stk;
```

```
import java.util.Comparator;

public class ExpiryDateComparatorDescending implements Comparator <Stock> {

       public int compare(Stock o1, Stock o2) {
                          return o2.getExpiryDate().compareTo(o1.getExpiryDate());

             }


}
```

3.

```
package stk;

import java.util.Comparator;

public class IdentityComparator implements Comparator <Stock> {

       public int compare(Stock s1, Stock s2) {

       return s1.getIdentity().compareTo(s2.getIdentity());

       }


             }
```

4.

```
package stk;

import java.util.Comparator;

public class IdentityComparatorDescending implements Comparator <Stock> {

       public int compare(Stock s1, Stock s2) {

       return s2.getIdentity().compareTo(s1.getIdentity());

       }


             }
```

5.

```
package stk;

import java.util.Comparator;

public class ManufacturingDateComparator implements Comparator <Stock> {
       public int compare(Stock o1, Stock o2) {
                          return 
o1.getManufacturingDate().compareTo(o2.getManufacturingDate());

             }

}
```

6.

```
package stk;

import java.util.Comparator;

public class ManufacturingDateComparatorDescending implements Comparator <Stock> {
       public int compare(Stock o1, Stock o2) {
                          return 
o1.getManufacturingDate().compareTo(o2.getManufacturingDate());

             }

}
```

7.

```
package stk;

import java.util.Comparator;

public class NameComparator implements Comparator <Stock>{
public int compare(Stock s1, Stock s2) {
                   return s1.getName().compareTo(s2.getName());

       }



}
```

8.

```
package stk;

import java.util.Comparator;

public class NameComparatorDescending implements Comparator <Stock>{
public int compare(Stock s1, Stock s2) {
            return s2.getName().compareTo(s1.getName());

    }



}
```

9.

```
package stk;

import java.util.Comparator;

public class PriceComparator implements Comparator <Stock>{

    public int compare(Stock o1, Stock o2) {
        // TODO Auto-generated method stub
        if(o1.getPrice() > o2.getPrice()) {
            return 1;
        }
        else if(o1.getPrice() < o2.getPrice  ()) {
            return -1;
        } else

        return 0;

}
}
```

10.

```
package stk;

import java.util.Comparator;

public class PriceComparatorDescending implements Comparator <Stock>{

    public int compare(Stock o1, Stock o2) {
        // TODO Auto-generated method stub
        if(o1.getPrice() < o2.getPrice()) {
            return 1;
        }
        else if(o1.getPrice() > o2.getPrice  ()) {
            return -1;
        } else

        return 0;

}
}
```

11.

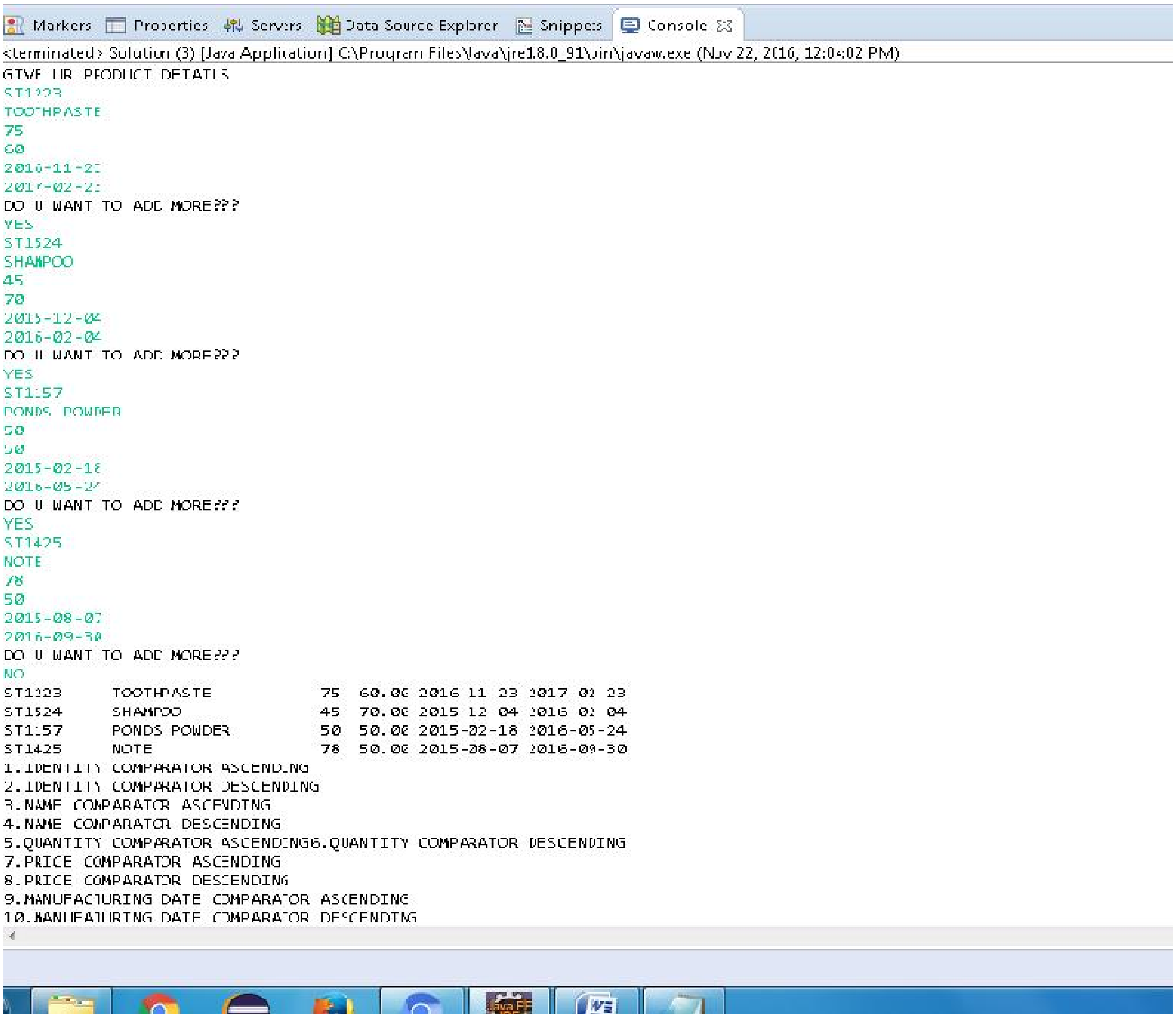
```
package stk;

import java.util.Comparator;

public class QuantityComparator implements Comparator <Stock>{
    public int compare(Stock o1, Stock o2) {
        // TODO Auto-generated method stub
        if(o1.getQuantity() > o2.getQuantity()) {
            return 1;
        }
        else if(o1.getQuantity() < o2.getQuantity()) {
            return -1;
        } else


        return 0;
    }
}
```

12.

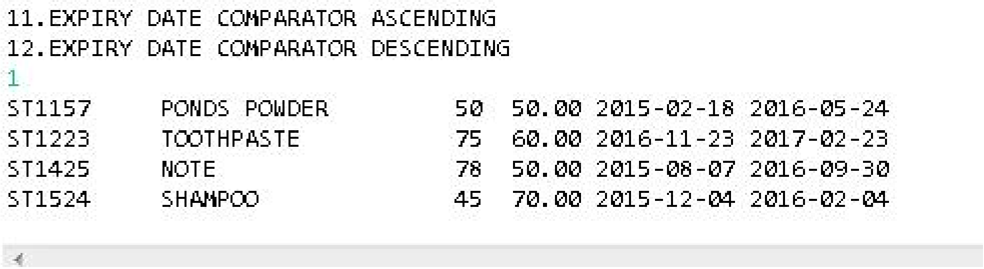
```
package stk;

import java.util.Comparator;

public class QuantityComparatorDescending implements Comparator <Stock>{
    public int compare(Stock o1, Stock o2) {
        // TODO Auto-generated method stub
        if(o1.getQuantity() < o2.getQuantity()) {
            return 1;
        }
        else if(o1.getQuantity() > o2.getQuantity()) {
            return -1;
        } else


        return 0;
    }
}
```

OUTPUTS

CASE 1

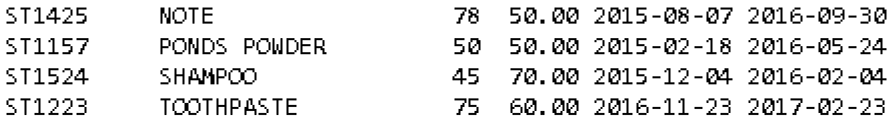


```
<terminated> Solution (3) [Java Application] C:\Program Files\Java\jre1.8.0_91\bin\javaw.exe (Nov 22, 2016, 12:04:02 PM)
GIVE UR PRODUCT DETAILS
ST1223
TOOTHPASTE
75
60
2016-11-23
2017-02-23
DO U WANT TO ADD MORE???
YES
ST1524
SHAMPOO
45
70
2015-12-04
2016-02-04
DO U WANT TO ADD MORE???
YES
ST1157
PONDS POWDER
50
50
2015-02-18
2016-05-24
DO U WANT TO ADD MORE???
YES
ST1425
NOTE
78
50
2015-08-07
2016-09-30
DO U WANT TO ADD MORE???
NO
ST1223      TOOTHPASTE           75  60.00 2016-11-23 2017-02-23
ST1524      SHAMPOO              45  70.00 2015-12-04 2016-02-04
ST1157      PONDS POWDER         50  50.00 2015-02-18 2016-05-24
ST1425      NOTE                 78  50.00 2015-08-07 2016-09-30
1.IDENTITY COMPARATOR ASCENDING
2.IDENTITY COMPARATOR DESCENDING
3.NAME COMPARATOR ASCENDING
4.NAME COMPARATOR DESCENDING
5.QUANTITY COMPARATOR ASCENDING6.QUANTITY COMPARATOR DESCENDING
7.PRICE COMPARATOR ASCENDING
8.PRICE COMPARATOR DESCENDING
9.MANUFACTURING DATE COMPARATOR ASCENDING
10.MANUFATURING DATE COMPARATOR DESCENDING
```



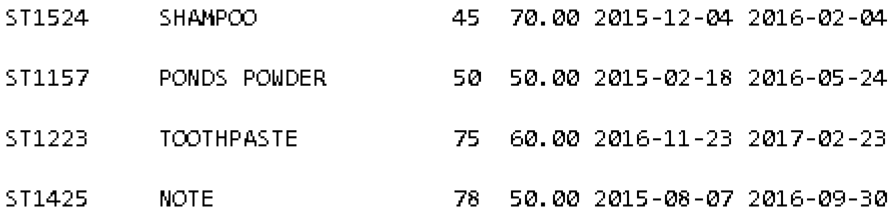
```
11.EXPIRY DATE COMPARATOR ASCENDING
12.EXPIRY DATE COMPARATOR DESCENDING
1
ST1157      PONDS POWDER         50  50.00 2015-02-18 2016-05-24
ST1223      TOOTHPASTE           75  60.00 2016-11-23 2017-02-23
ST1425      NOTE                 78  50.00 2015-08-07 2016-09-30
ST1524      SHAMPOO              45  70.00 2015-12-04 2016-02-04
```

CASE 2



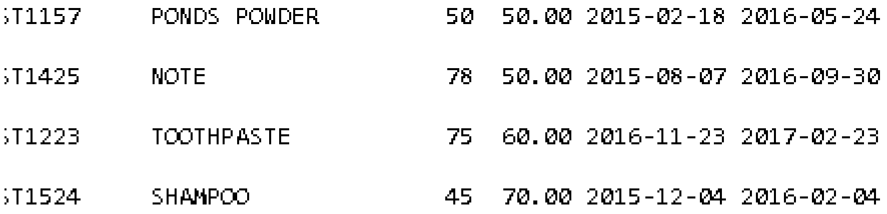
```
ST1425     NOTE                  78  50.00 2015-08-07 2016-09-30
ST1157     PONDS POWDER          50  50.00 2015-02-18 2016-05-24
ST1524     SHAMPOO               45  70.00 2015-12-04 2016-02-04
ST1223     TOOTHPASTE            75  60.00 2016-11-23 2017-02-23
```

CASE 3



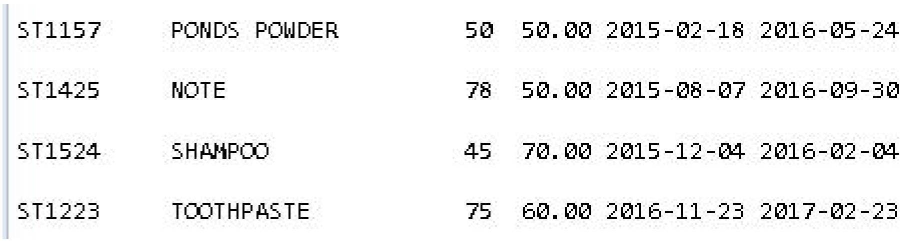
```
ST1524     SHAMPOO               45  70.00 2015-12-04 2016-02-04

ST1157     PONDS POWDER          50  50.00 2015-02-18 2016-05-24

ST1223     TOOTHPASTE            75  60.00 2016-11-23 2017-02-23

ST1425     NOTE                  78  50.00 2015-08-07 2016-09-30
```

CASE 4



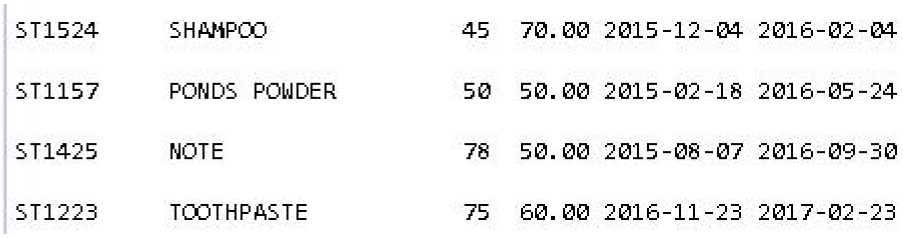
```
ST1157     PONDS POWDER          50  50.00 2015-02-18 2016-05-24

ST1425     NOTE                  78  50.00 2015-08-07 2016-09-30

ST1223     TOOTHPASTE            75  60.00 2016-11-23 2017-02-23

ST1524     SHAMPOO               45  70.00 2015-12-04 2016-02-04
```

CASE 5



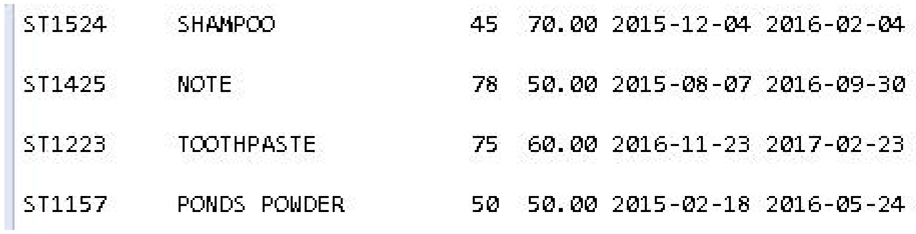
```
ST1157     PONDS POWDER          50  50.00 2015-02-18 2016-05-24

ST1425     NOTE                  78  50.00 2015-08-07 2016-09-30

ST1524     SHAMPOO               45  70.00 2015-12-04 2016-02-04

ST1223     TOOTHPASTE            75  60.00 2016-11-23 2017-02-23
```

CASE 6



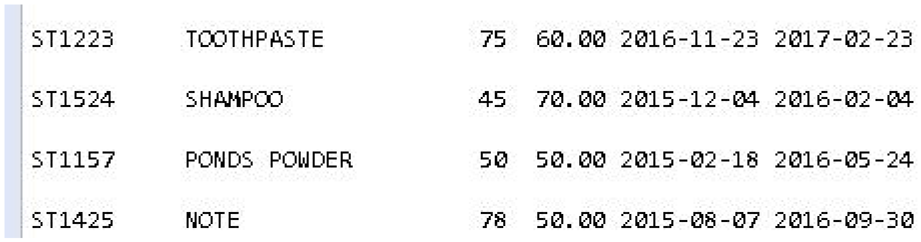
```
ST1524     SHAMPOO               45  70.00 2015-12-04 2016-02-04

ST1157     PONDS POWDER          50  50.00 2015-02-18 2016-05-24

ST1425     NOTE                  78  50.00 2015-08-07 2016-09-30

ST1223     TOOTHPASTE            75  60.00 2016-11-23 2017-02-23
```

CASE 7



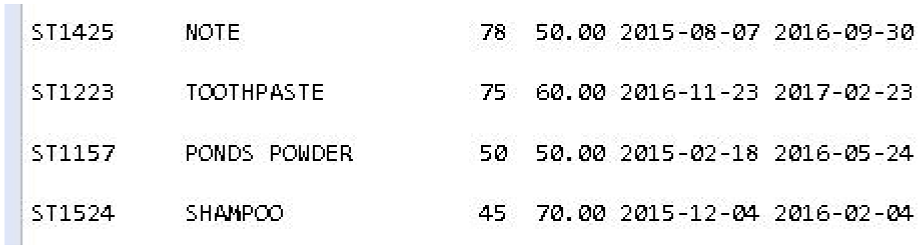
```
ST1524     SHAMPOO              45  70.00 2015-12-04 2016-02-04

ST1425     NOTE                 78  50.00 2015-08-07 2016-09-30

ST1223     TOOTHPASTE           75  60.00 2016-11-23 2017-02-23

ST1157     PONDS POWDER         50  50.00 2015-02-18 2016-05-24
```

CASE 8



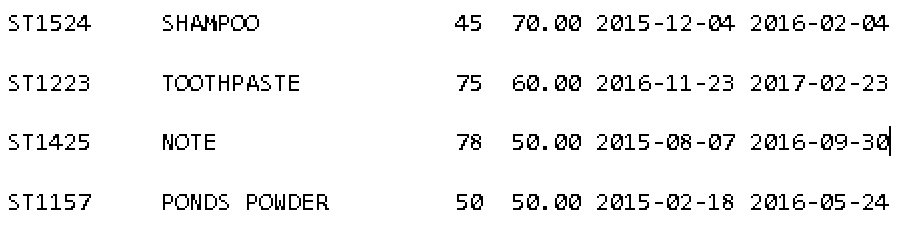
```
ST1223     TOOTHPASTE           75  60.00 2016-11-23 2017-02-23

ST1524     SHAMPOO              45  70.00 2015-12-04 2016-02-04

ST1157     PONDS POWDER         50  50.00 2015-02-18 2016-05-24

ST1425     NOTE                 78  50.00 2015-08-07 2016-09-30
```

CASE 9



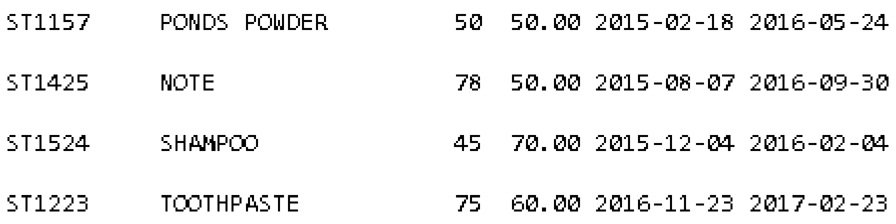
```
ST1425     NOTE                 78  50.00 2015-08-07 2016-09-30

ST1223     TOOTHPASTE           75  60.00 2016-11-23 2017-02-23

ST1157     PONDS POWDER         50  50.00 2015-02-18 2016-05-24

ST1524     SHAMPOO              45  70.00 2015-12-04 2016-02-04
```

CASE 10



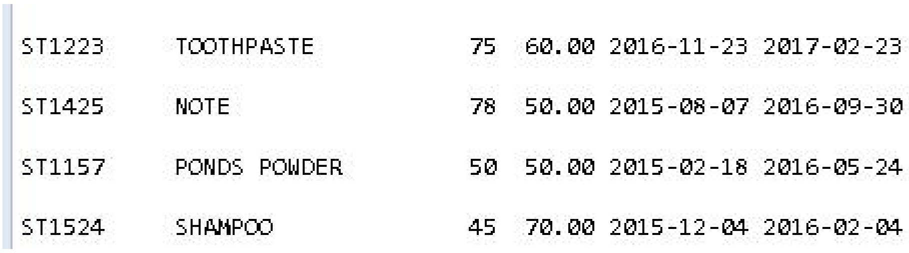
```
ST1524     SHAMPOO              45  70.00 2015-12-04 2016-02-04

ST1223     TOOTHPASTE           75  60.00 2016-11-23 2017-02-23

ST1425     NOTE                 78  50.00 2015-08-07 2016-09-30

ST1157     PONDS POWDER         50  50.00 2015-02-18 2016-05-24
```

CASE 11



```
ST1157     PONDS POWDER         50  50.00 2015-02-18 2016-05-24

ST1425     NOTE                 78  50.00 2015-08-07 2016-09-30

ST1524     SHAMPOO              45  70.00 2015-12-04 2016-02-04

ST1223     TOOTHPASTE           75  60.00 2016-11-23 2017-02-23
```

CASE 12



```
ST1223     TOOTHPASTE          75  60.00 2016-11-23 2017-02-23

ST1425     NOTE                78  50.00 2015-08-07 2016-09-30

ST1157     PONDS POWDER        50  50.00 2015-02-18 2016-05-24

ST1524     SHAMPOO             45  70.00 2015-12-04 2016-02-04
```