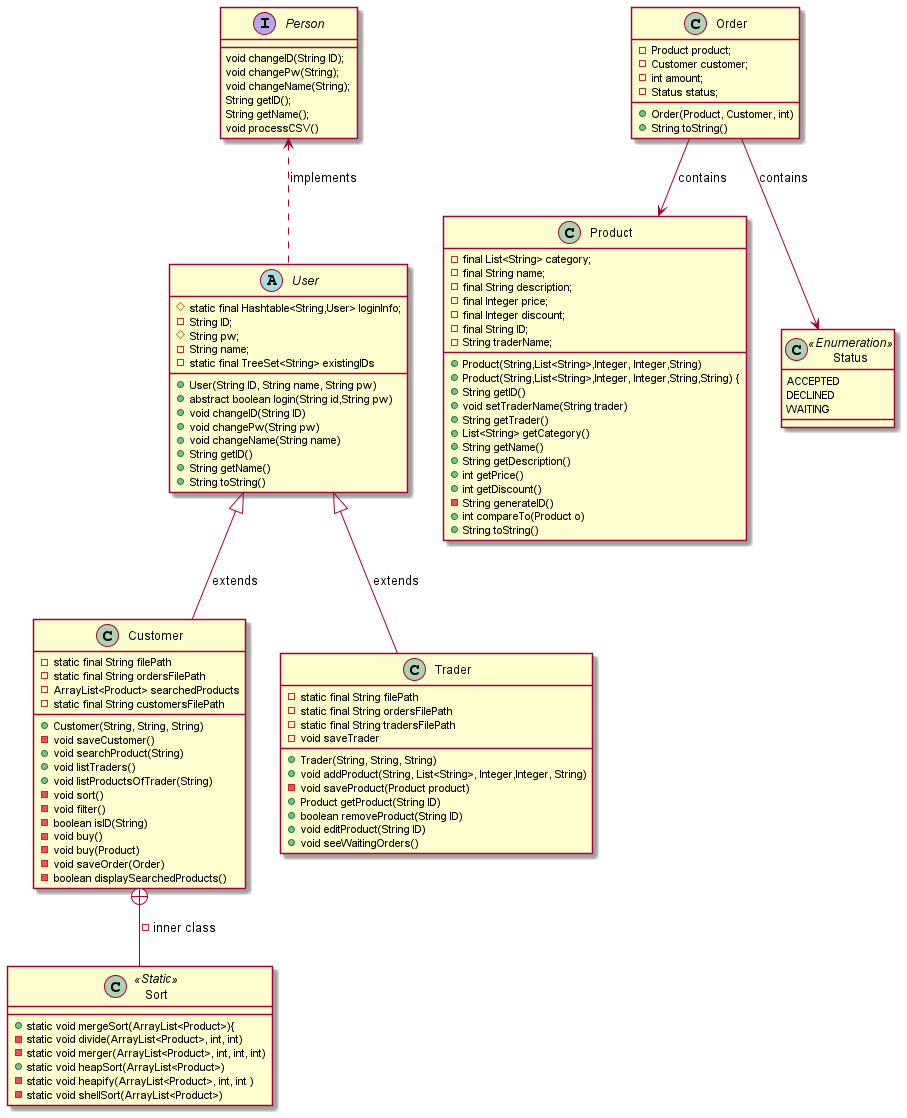
**GTU Department of Computer Engineering**

**CSE 222/505 - Spring 2021**

**Homework #4 Report**

**Gökbey Gazi KESKİN**

**1901042631**

1. **Class Diagram**
2. **Problem Solution Approach**

First, I created a static method to process csv files in Person interface. Then I created 4 different data files. One for products, one for orders, one for traders and one for customers. I created a private inner class in Customer to sort products. The amount of data is too big to keep in memory at once, so only the needed data is kept in memory. For example, when Customer searches “Book” only products which contains the word book are saved into memory. And when a customer makes a new search, searchedProducts arraylist is emptied and refilled. This process of not keeping everything at once slows things a bit, but this is the only way to hold this amount of data.

**Where did I used which Data Structure and why?**

1)Hashtable



I used hashtable to keep ID and the related User. Because searching a User with ID is O(1) this way.

2)Tree



I used a tree to keep existing id’s because everytime a new user is created, program check if this id is already existing or not. And search operation on TreeSet (self-balancing bst) is O(logn).

3)Queue



I used queue for listing all the traders. I added all traders while reading to queue and then polled them one by one.

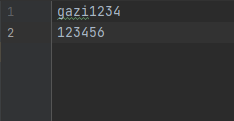
4)ArrayList

I used lots of array list.



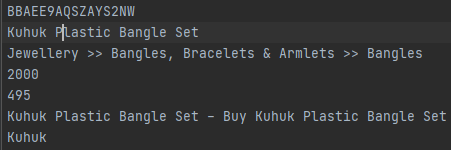
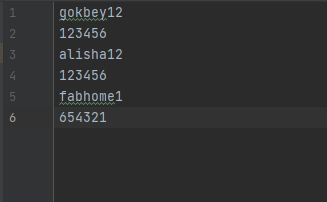
One example is searched products. There is no shifting so arraylist is ideal for this condition. Because everytime customer searches, products are added one by one to the end, and on next search searchProduct points to a new ArrayList. And garbage collector collects the old arraylist.

Note: I saw I should save the users in a separate file after I finished the assignment. So I only kept the users which are created by the user of my program, I didn’t kept the users from csv file. They already exist in products.txt file.



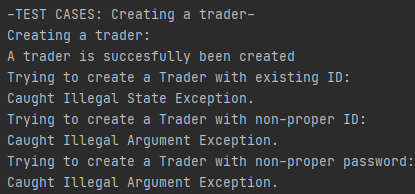
customers.txt

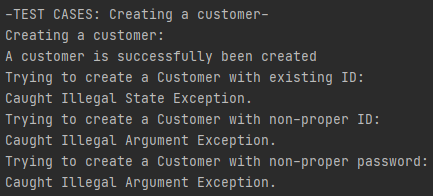
traders.txt

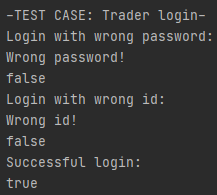
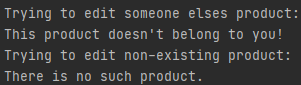


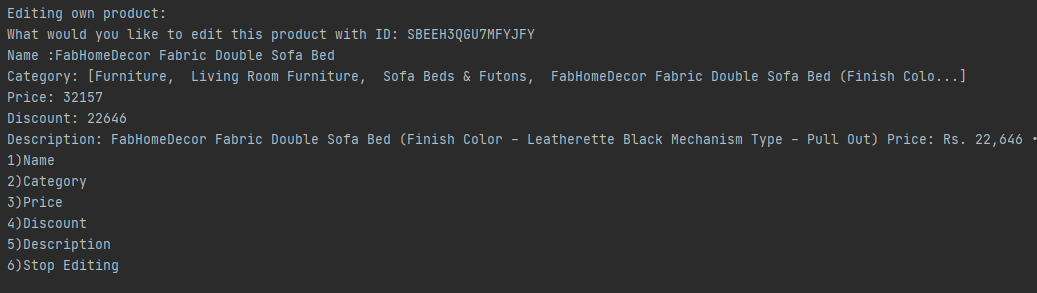
An entry from products.txt

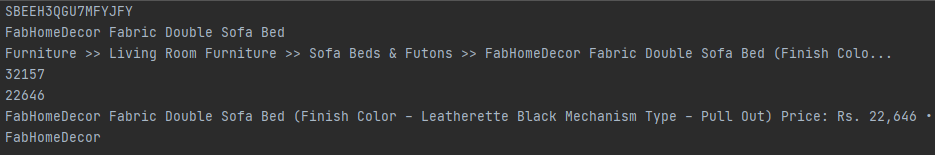
1. **Test Cases & Results**

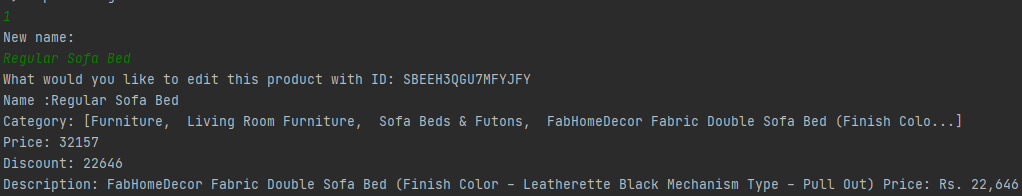
****

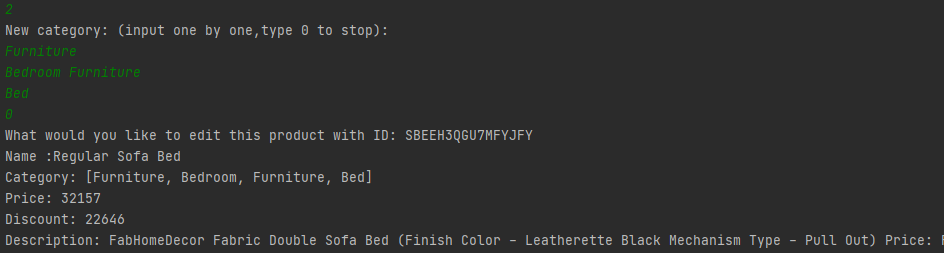
****

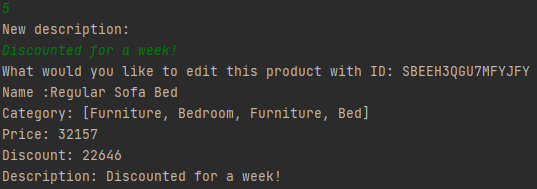
****

**Before:**

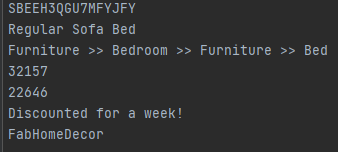
****

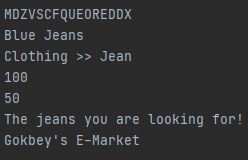
****

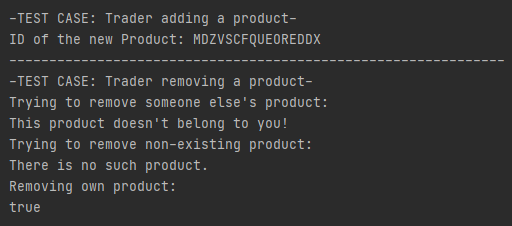
****

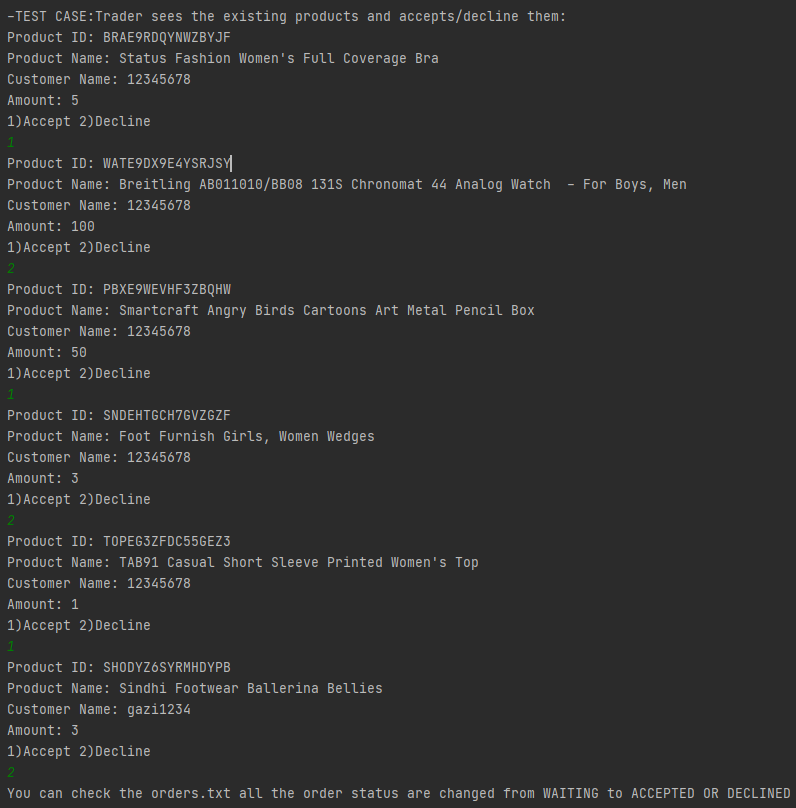
****

**After:**

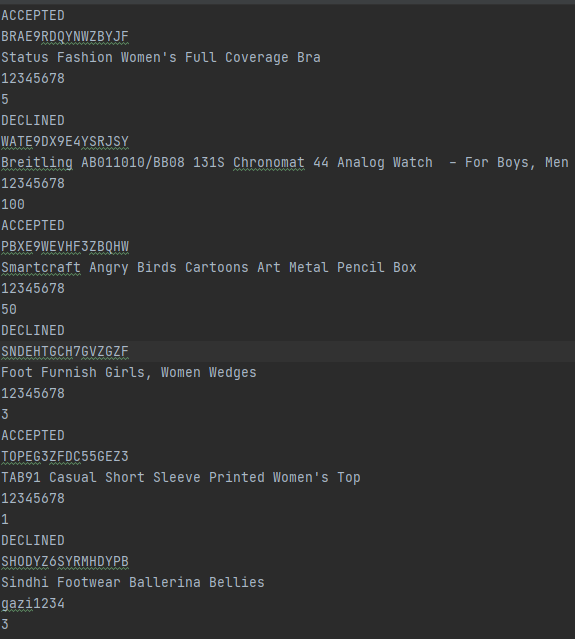
****

****

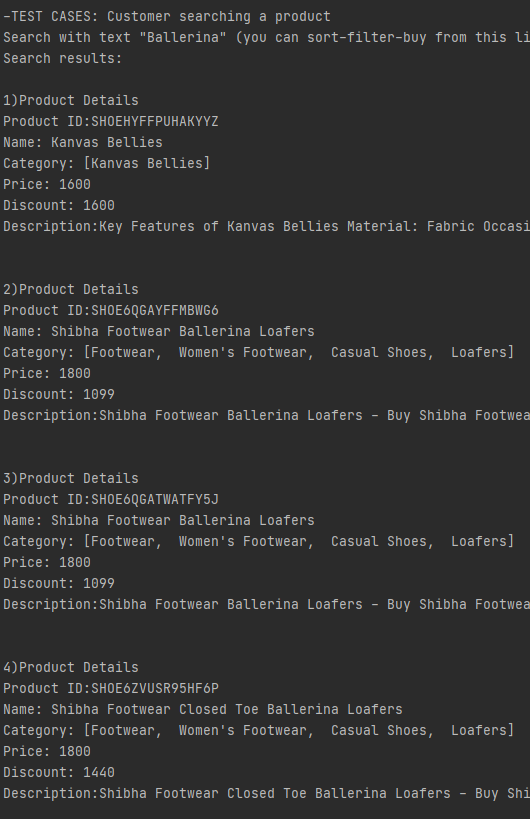
****

****

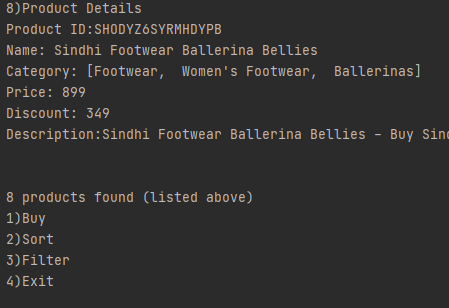
**Before:**

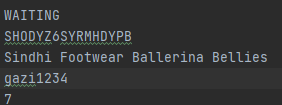
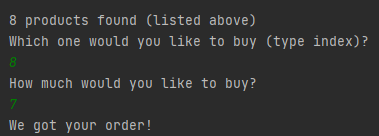
****

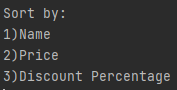
**After:**

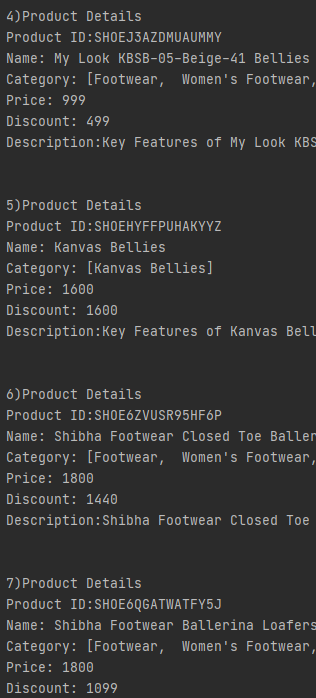
****

**…**

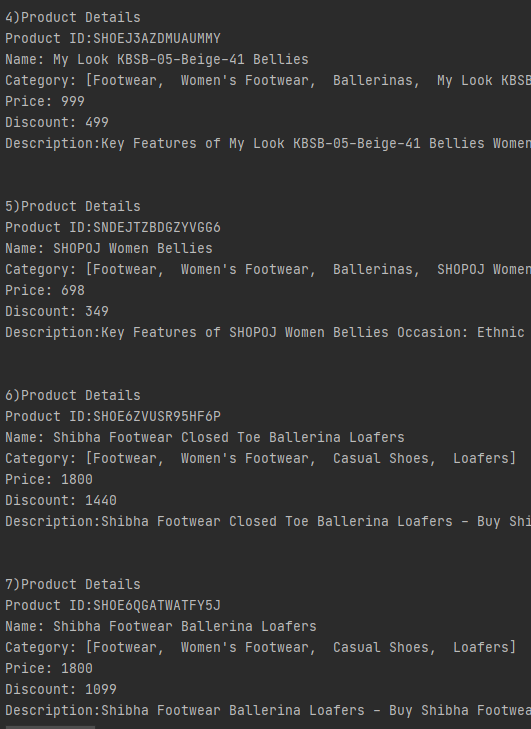
****

**Buy)**

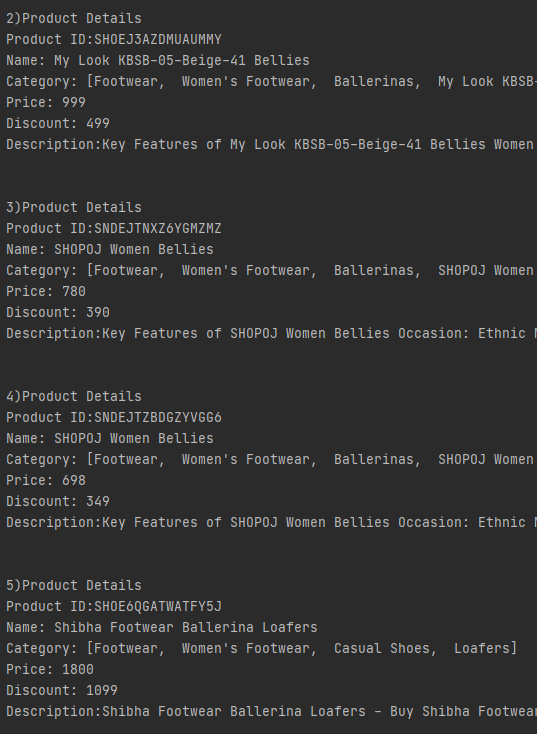
**Sort)**

**Sort By Price)**

**Sort By Discount Percentage)**

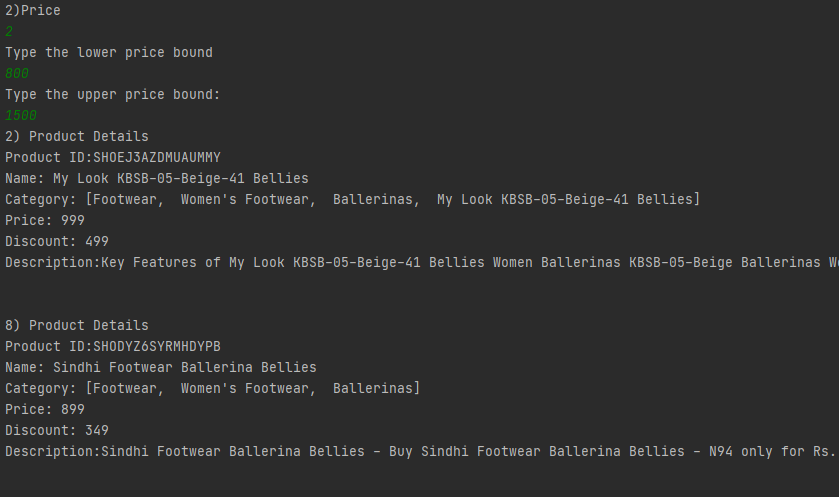
****

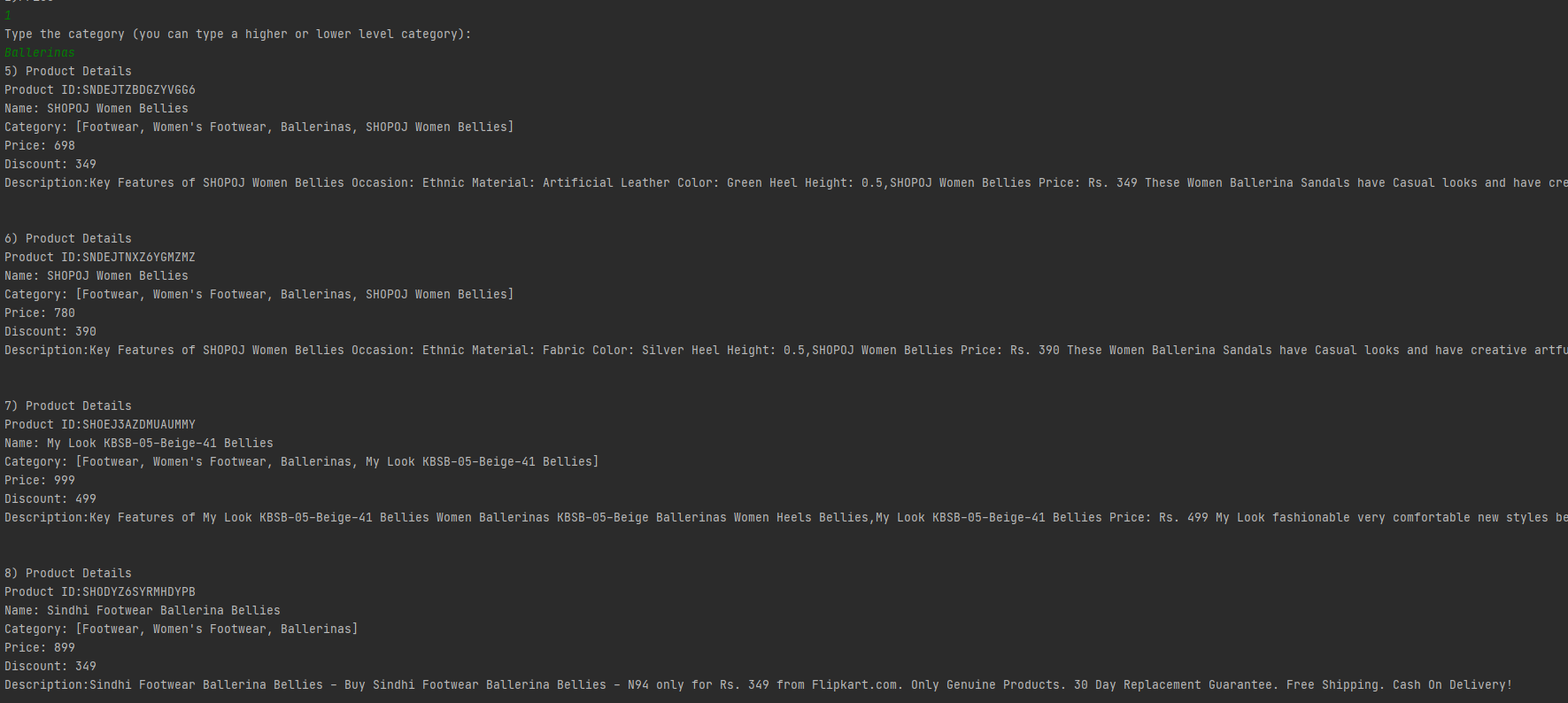
**Sort By Name)**

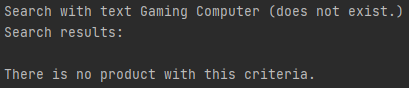
****

**Filter)**

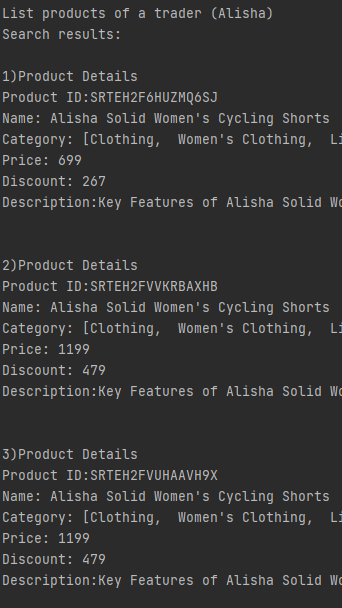
**Filter By Price)**

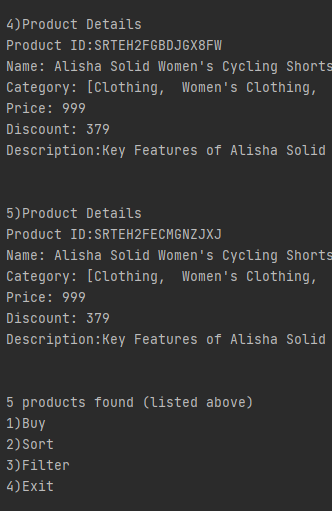
****

**Filter By Category)**

****

**….**

****

****