|  |
| --- |
| **Problem Statement:**  For this problem you have to perform the following tasks:  **SubTask 1**: Create a model class as ***Product*** with product\_id, name, price and rating as its properties.  **SubTask2**: Create a List, ***L*** of type Product .  **SubTask3**: Populate your collection ***L*** with ***N*** number of products i.e. (an array of Product objects)  **SubTask4**:Define separate methods to sort the user defined collection ***L*** based on the product’s ***rating*** value, ***product\_id*** *and*product ***name***. (Hint: Using Comparable / Comparator)  **SubTask5**: Display the sorted collection using an iterator.  **Note:**  1. Do not remove the predefined code else your code may not execute as expected.  2. You’ve to solve the problem using **Linked List only**. Skipping the concept may lead to disqualification. |
| **Input & Output Format:**  **Input Format**  The first line of input contains a single integer **N** denoting the number of objects.  The subsequent **N** lines contain an integer denoting product*\_id.*  The subsequent **N** lines contain a string denoting *name.*  The subsequent **N** lines contain a double denoting *price.*  The subsequent **N** lines contain a integer denoting rating in scale of 1-5*.*  The last line contains a number **x** denoting the choice of sorting.  1.SortById  2.SortByName  3.SortByRating  **Output Format**  Displays the evaluated result of the invoked methods |
| **Sample Input :**  2  101  laptop  70000.00  4  102  mobile  50000.00  4  2  **Sample Output:**  [Product [product\_id=101, nameString=laptop, price=70000.0, rating=4]  , Product [product\_id=102, nameString=mobile, price=50000.0, rating=4]  ] |
| **Test Case 1 :**  3  1  bag  1000.00  3  2  shoes  500.00  5  3  dress  4000.00  2  3  [Product [product\_id=3, nameString=dress, price=4000.0, rating=2]  , Product [product\_id=1, nameString=bag, price=1000.0, rating=3]  , Product [product\_id=2, nameString=shoes, price=500.0, rating=5]  ]  **Test Case 2 :**  2  111  Car  1000000.00  4  222  bike  200000.00  2  3  [Product [product\_id=222, nameString=bike, price=200000.0, rating=2]  , Product [product\_id=111, nameString=Car, price=1000000.0, rating=4]  ]  **Test case 3:**  4  002  Book  500.00  5  001  USB  2000.00  4  203  mousepad  100.00  3  001  Clock  400.00  5  1  [Product [product\_id=1, nameString=USB, price=2000.0, rating=4]  , Product [product\_id=1, nameString=Clock, price=400.0, rating=5]  , Product [product\_id=2, nameString=Book, price=500.0, rating=5]  , Product [product\_id=203, nameString=mousepad, price=100.0, rating=3]  ]  **Test case 4:**  1  101  SportsItems  40000.00  4  1  **[Product [product\_id=101, nameString=SportsItems, price=40000.0, rating=4]**  **]**  **Test case 5:**  1  2  345  678.90  5  5  **Invalid choice** |