ASSIGNMENT ON STREAM

Q1) Create the following classes:

class Fruit { String name; int calories; int price; String color; }

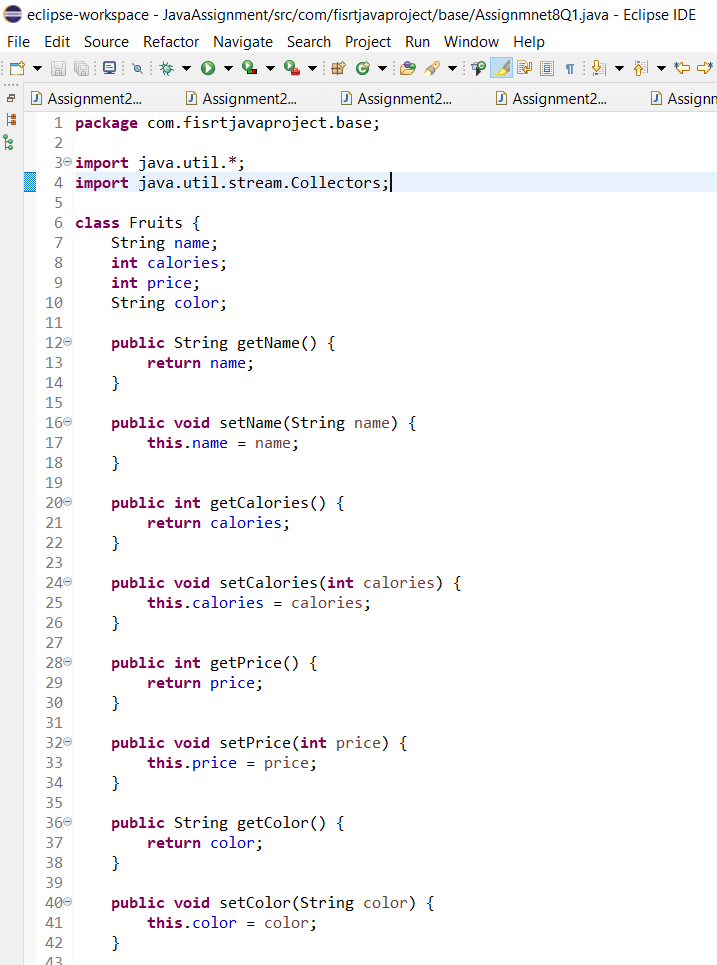
Display the following:

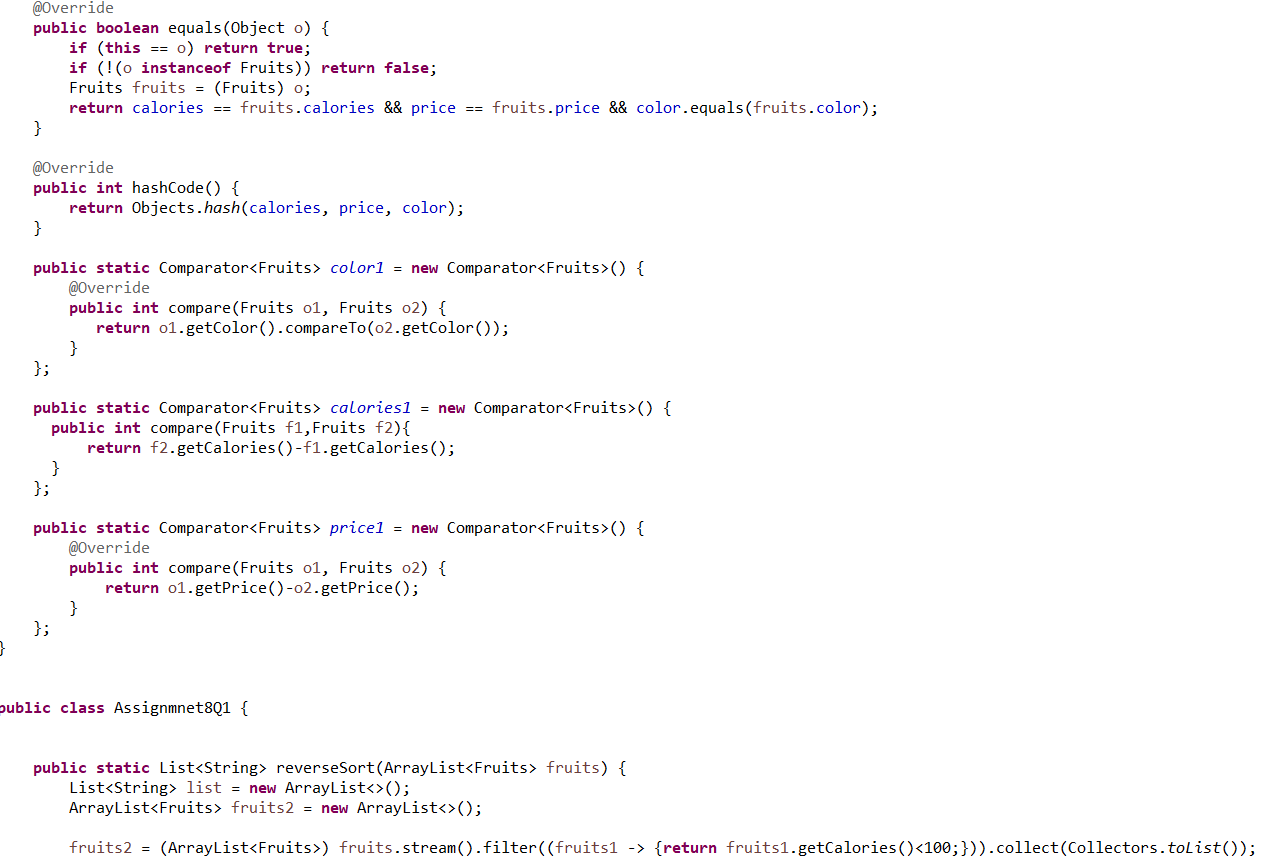
1. Display the fruit names of low calories fruits i.e. calories < 100 sorted in descending order of calories.

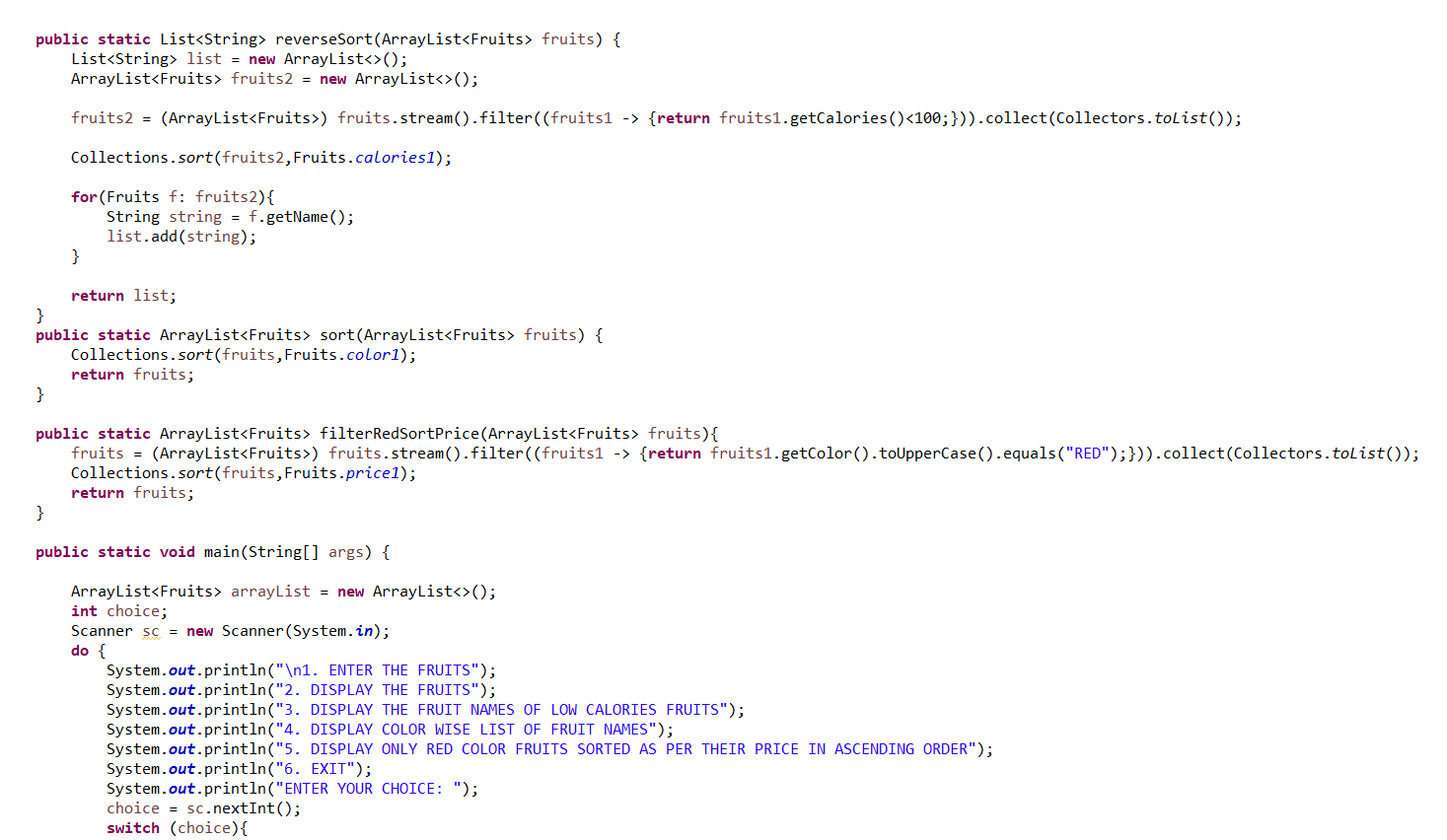
2. Display color wise list of fruit names.

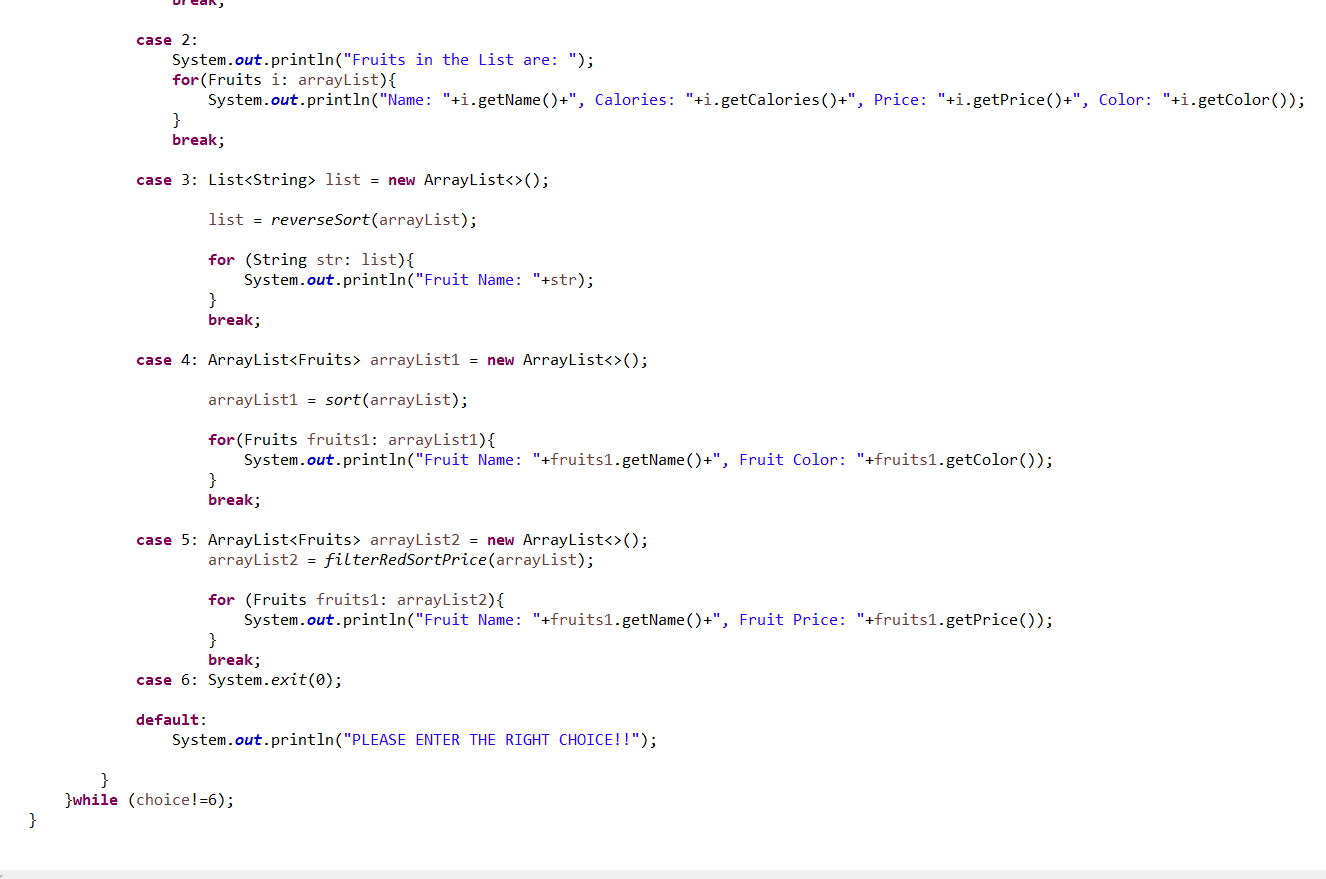
3. Display only RED color fruits sorted as per their price in ascending order.

**Code:**

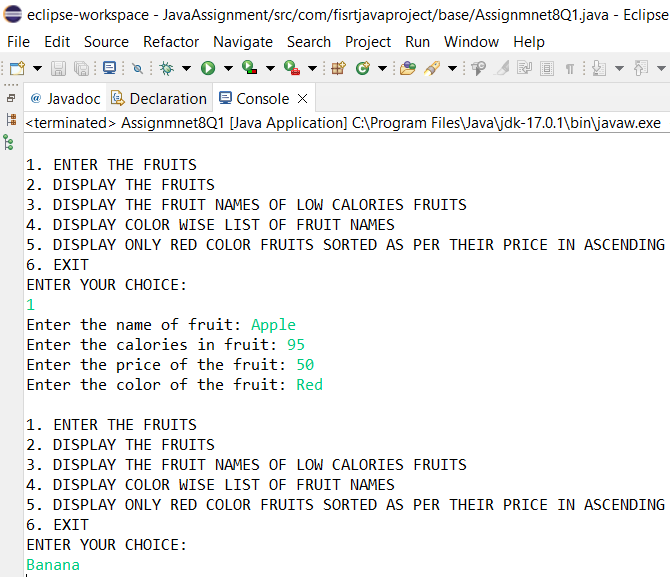








**Output:**



Q2) Setup:

Create the following classes:

class News { int newsId; String postedByUser; String commentByUser; String comment; }

Find Out:

1. Find out the newsId which has received maximum comments.

2. Find out how many times the word 'budget' arrived in user comments all news.

3. Find out which user has posted maximum comments.

4. Display commentByUser wise number of comments.

**Code:**

package com.fisrtjavaproject.base;

import java.util.\*;

import java.util.stream.Collectors;

class News {

int newsId;

String postedByUser;

String commentByUser;

String comment;

public News(int newsId, String postedByUser, String commentByUser, String comment) {

this.newsId = newsId;

this.postedByUser = postedByUser;

this.commentByUser = commentByUser;

this.comment = comment;

}

public News() {

}

public int getNewsId() {

return newsId;

}

public void setNewsId(int newsId) {

this.newsId = newsId;

}

public String getPostedByUser() {

return postedByUser;

}

public void setPostedByUser(String postedByUser) {

this.postedByUser = postedByUser;

}

public String getCommentByUser() {

return commentByUser;

}

public void setCommentByUser(String commentByUser) {

this.commentByUser = commentByUser;

}

public String getComment() {

return comment;

}

public void setComment(String comment) {

this.comment = comment;

}

public boolean equals(Object o) {

if (this == o) return true;

if (!(o instanceof News)) return false;

News news = (News) o;

return newsId == news.newsId && commentByUser.equals(news.commentByUser) && comment.equals(news.comment);

}

public int hashCode() {

return Objects.hash(newsId, commentByUser, comment);

}

}

public class Assignment8Q2 {

public static int maxComments(List<News> news) {

Map<Integer,Integer> map = new HashMap<>();

for (News news1: news){

int id = news1.getNewsId();

if(map.containsKey(id)){

map.put(id,map.get(id)+1);

} else {

map.put(id,1);

}

}

List<Map.Entry<Integer, Integer>> list = new ArrayList<>(map.entrySet());

list.sort(Map.Entry.comparingByValue());

return list.get(list.size()-1).getKey();

}

public static int budgetCount (List < News > news) {

List<String> list = new ArrayList<>();

for(News i: news){

String comment = i.getComment();

list.add(Arrays.toString(comment.split("budget")));

}

return list.size();

}

public static String maxCommentsByUser (List < News > news) {

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

for(News news1: news){

String username = news1.getCommentByUser();

if(map.containsKey(username)){

map.put(username,map.get(username)+1);

} else {

map.put(username,1);

}

}

List<Map.Entry<String, Integer>> list = new ArrayList<>(map.entrySet());

list.sort(Map.Entry.comparingByValue());

return list.get(list.size()-1).getKey();

}

public static Map<String, Integer> sortMaxCommentsByUser (List < News > news) {

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

for(News news1: news){

String username = news1.getCommentByUser();

if(map.containsKey(username)){

map.put(username,map.get(username)+1);

} else {

map.put(username,1);

}

}

HashMap<String, Integer> temp

= map.entrySet()

.stream()

.sorted((i1, i2)

-> i2.getValue().compareTo(

i1.getValue()))

.collect(Collectors.toMap(

Map.Entry::getKey,

Map.Entry::getValue,

(e1, e2) -> e1, LinkedHashMap::new));

return temp;

}

public static void main(String[] args) {

List<News> list = new ArrayList<>();

int choice;

Scanner sc = new Scanner(System.in);

News news1 = new News(1,

"Ayush Agrawal",

"Atharva Upadhye",

"I want to see the budget");

News news2 = new News(1,

"Ayush Agrawal",

"Pratik Nandurkar",

"budget is my fundamental right");

News news3 = new News(2,

"Ritik Dixit",

"Shraddha Gupta",

"budget 2022");

list.add(news1);

list.add(news2);

list.add(news3);

do{

System.out.println("1. ENTER THE NEWS DETAILS");

System.out.println("2. DISPLAY THE NEWS DETAILS");

System.out.println("3. FIND OUT THE newsId WHICH HAS RECEIVED MAXIMUM COMMENTS.");

System.out.println("4. FIND OUT HOW MANY TIMES THE WORD 'BUDGET' ARRIVED IN USER COMMENTS ALL NEWS.");

System.out.println("5. FIND OUT WHICH USER HAS POSTED MAXIMUM COMMENTS.");

System.out.println("6. DISPLAY COMMENT BY USER WISE NUMBER OF COMMENTS.");

System.out.println("7. EXIT");

System.out.println("ENTER YOUR CHOICE:");

choice = sc.nextInt();

switch (choice){

case 1:

News news = new News();

try {

System.out.print("Enter the newsID: ");

int newId = sc.nextInt();

news.setNewsId(newId);

sc.nextLine();

System.out.print("Enter the name who posted the news: ");

String name = sc.nextLine();

news.setPostedByUser(name);

System.out.print("Enter the comment: ");

String comment = sc.nextLine();

news.setComment(comment);

System.out.print("Enter the name who posted the comment: ");

String username = sc.nextLine();

news.setCommentByUser(username);

list.add(news);

}catch (Exception e){

System.out.println(e);

System.out.println("Enter the correct input please!!");

}

break;

case 2: for(News new1: list){

System.out.println("News ID: "+new1.getNewsId());

System.out.println("News Posted By: "+new1.getPostedByUser());

System.out.println("Comment by User: "+new1.getCommentByUser());

System.out.println("Comment Posted: "+new1.getComment());

System.out.println();

}

break;

case 3: int id = maxComments(list);

System.out.println("NEWS ID WHICH HAS RECEIVED MAXIMUM COMMENTS: "+id);

System.out.println();

break;

case 4: int countBudget = budgetCount(list);

System.out.println("HOW MANY TIMES THE WORD 'BUDGET' ARRIVED IN USER COMMENTS ALL NEWS: "+countBudget);

System.out.println();

break;

case 5: String name = maxCommentsByUser(list);

System.out.println("USER HAS POSTED MAXIMUM COMMENTS: "+name);

System.out.println();

break;

case 6: Map<String,Integer> maxCommentByUser = sortMaxCommentsByUser(list);

for (Map.Entry<String,Integer> mp: maxCommentByUser.entrySet()){

System.out.println("Name of the User: "+mp.getKey());

System.out.println("Numbers of Comments: "+mp.getValue());

System.out.println();

}

System.out.println();

break;

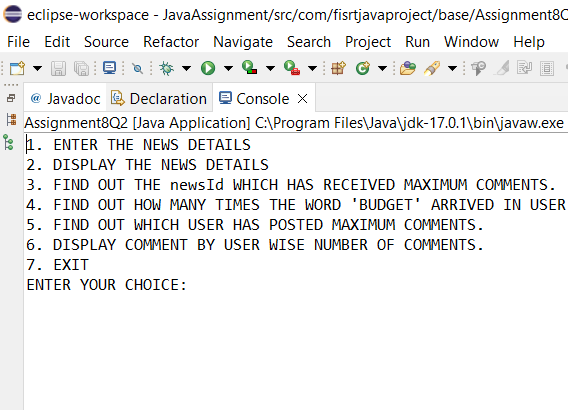
}

}while (choice!=7);

}

}

**OUTPUT :**



Q3) Setup:

Create the following classes:

class Trader { String name; String city; }

Find Out:

1. What are all the unique cities where the traders work?

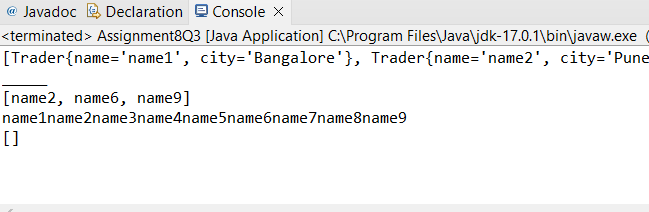
2. Find all traders from Pune and sort them by name.

3. Return a string of all traders’ names sorted alphabetically.

4. Are any traders based in Indore?

**Code:**

**Output:**



Q4) Setup:

Create the following classes:

class Trader { String name; String city; }

class Transaction { Trader trader; int year; int value; }

1. Find all transactions in the year 2011 and sort them by value (small to high).

2. Print all transactions’ values from the traders living in Delhi.

3. What’s the highest value of all the transactions?

4. Find the transaction with the smallest value.

**CODE :**

**OUTPUT :**