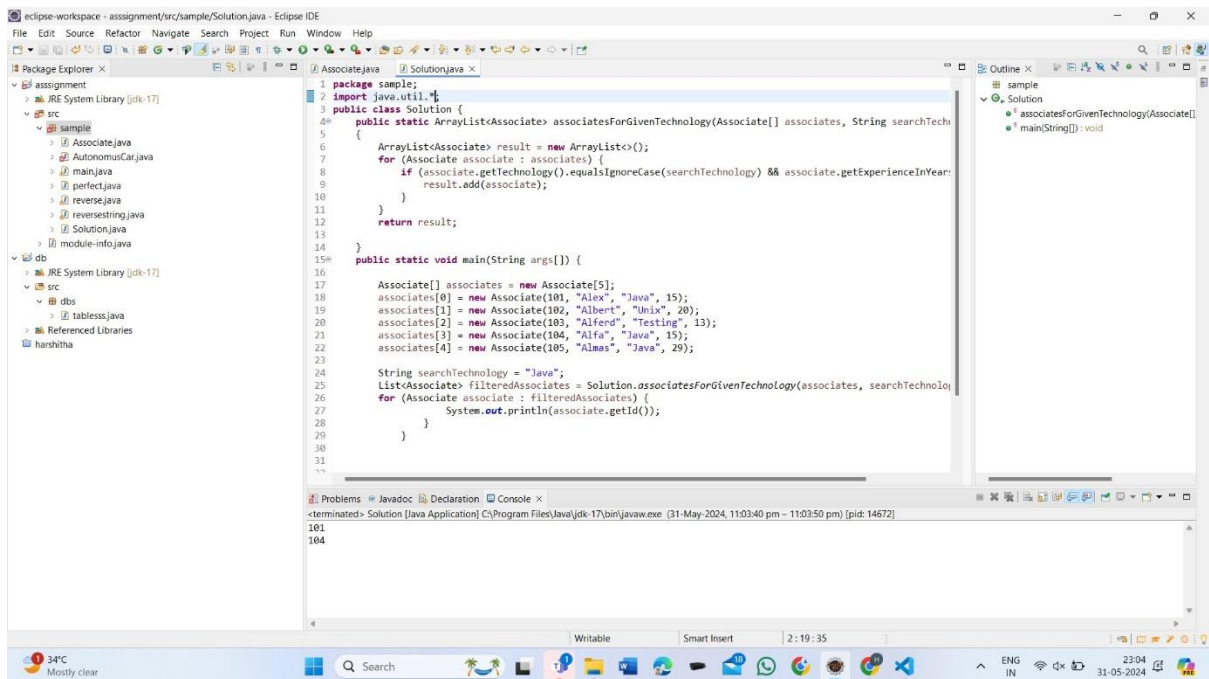


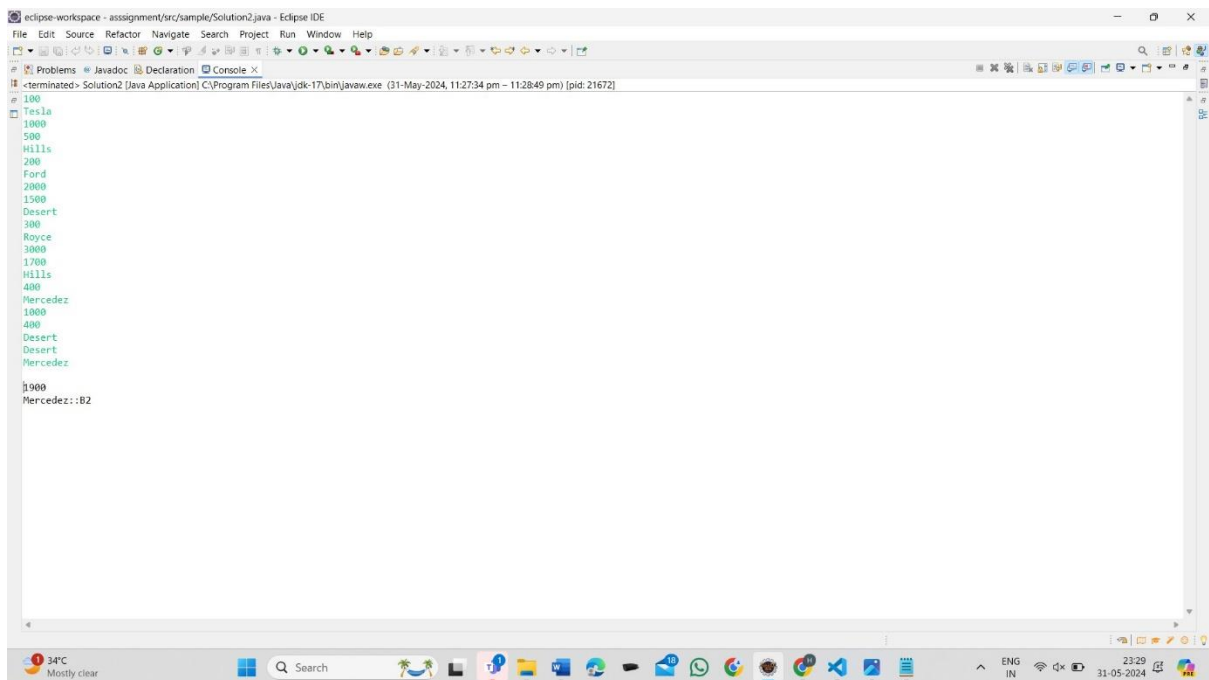
Question: 1

Associates

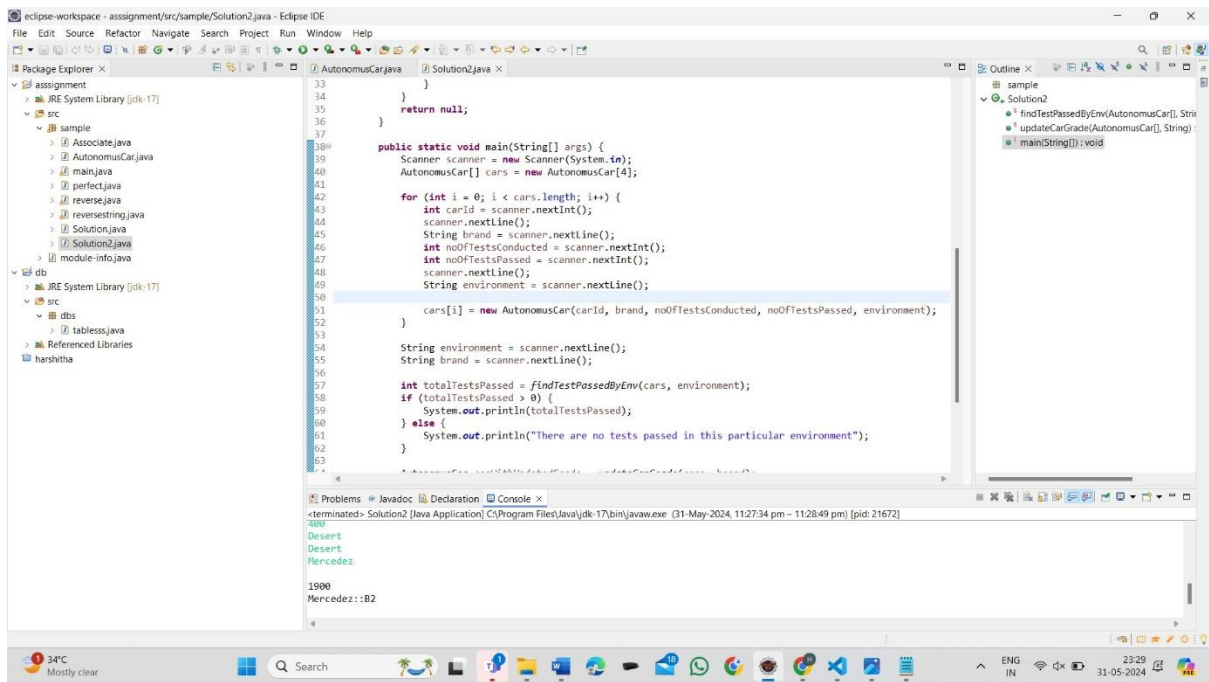


```
1 package sample;
2 import java.util.*;
3 public class Solution {
4     public static ArrayList<Associate> associatesForGivenTechnology(Associate[] associates, String searchTechnology) {
5         ArrayList<Associate> result = new ArrayList<>();
6         for (Associate associate : associates) {
7             if (associate.getTechnology().equalsIgnoreCase(searchTechnology) && associate.getExperienceInYear() > 10) {
8                 result.add(associate);
9             }
10        }
11        return result;
12    }
13
14    public static void main(String args[]) {
15
16        Associate[] associates = new Associate[5];
17        associates[0] = new Associate(101, "Alex", "Java", 15);
18        associates[1] = new Associate(102, "Albert", "Unix", 20);
19        associates[2] = new Associate(103, "Alfred", "Testing", 13);
20        associates[3] = new Associate(104, "Alfa", "Java", 15);
21        associates[4] = new Associate(105, "Almas", "Java", 29);
22
23        String searchTechnology = "Java";
24        List<Associate> filteredAssociates = Solution.associatesForGivenTechnology(associates, searchTechnology);
25        for (Associate associate : filteredAssociates) {
26            System.out.println(associate.getId());
27        }
28    }
29
30 }
31
32 }
```

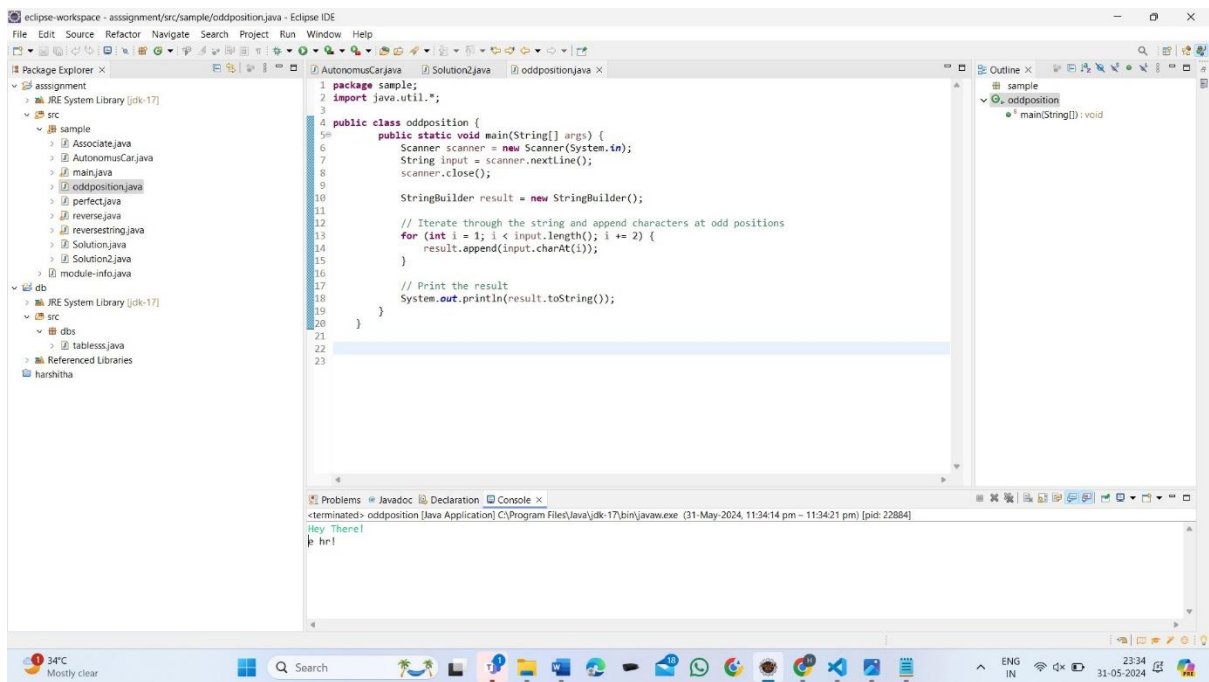
2. Autonomus car



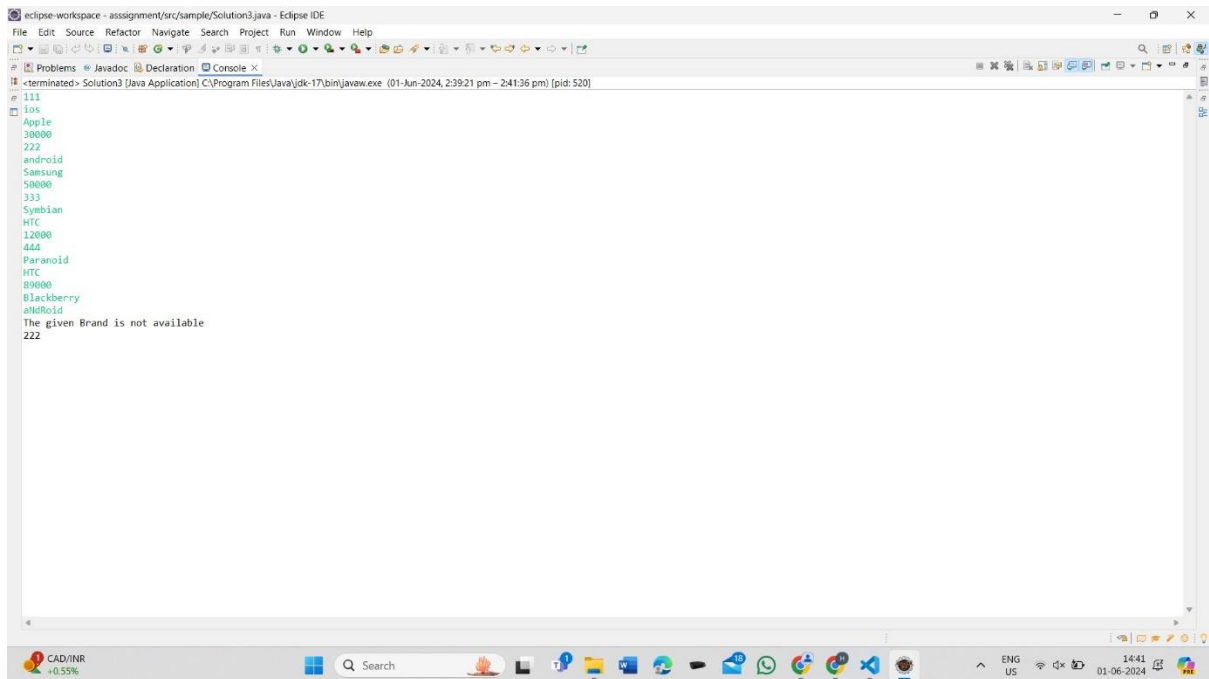
```
1 package sample;
2 import java.util.*;
3 public class Solution2 {
4     public static ArrayList<Car> getCar(Car[] cars, String searchCar) {
5         ArrayList<Car> result = new ArrayList<>();
6         for (Car car : cars) {
7             if (car.getCar().equalsIgnoreCase(searchCar) && car.getPrice() > 1000) {
8                 result.add(car);
9             }
10        }
11        return result;
12    }
13
14    public static void main(String args[]) {
15
16        Car[] cars = new Car[10];
17        cars[0] = new Car(1000, "Tesla", 1000);
18        cars[1] = new Car(500, "Hills", 200);
19        cars[2] = new Car(200, "Ford", 2000);
20        cars[3] = new Car(1500, "Desert", 300);
21        cars[4] = new Car(300, "Royce", 3000);
22        cars[5] = new Car(1700, "Hills", 400);
23        cars[6] = new Car(400, "Mercedes", 1000);
24        cars[7] = new Car(400, "Desert", 1000);
25        cars[8] = new Car(1000, "Mercedes", 1000);
26        cars[9] = new Car(1000, "Mercedes", 1000);
27
28        String searchCar = "Mercedes";
29        List<Car> filteredCars = Solution2.getCar(cars, searchCar);
30        for (Car car : filteredCars) {
31            System.out.println(car.getId());
32        }
33    }
34
35 }
```



3 oddCharacters in a String

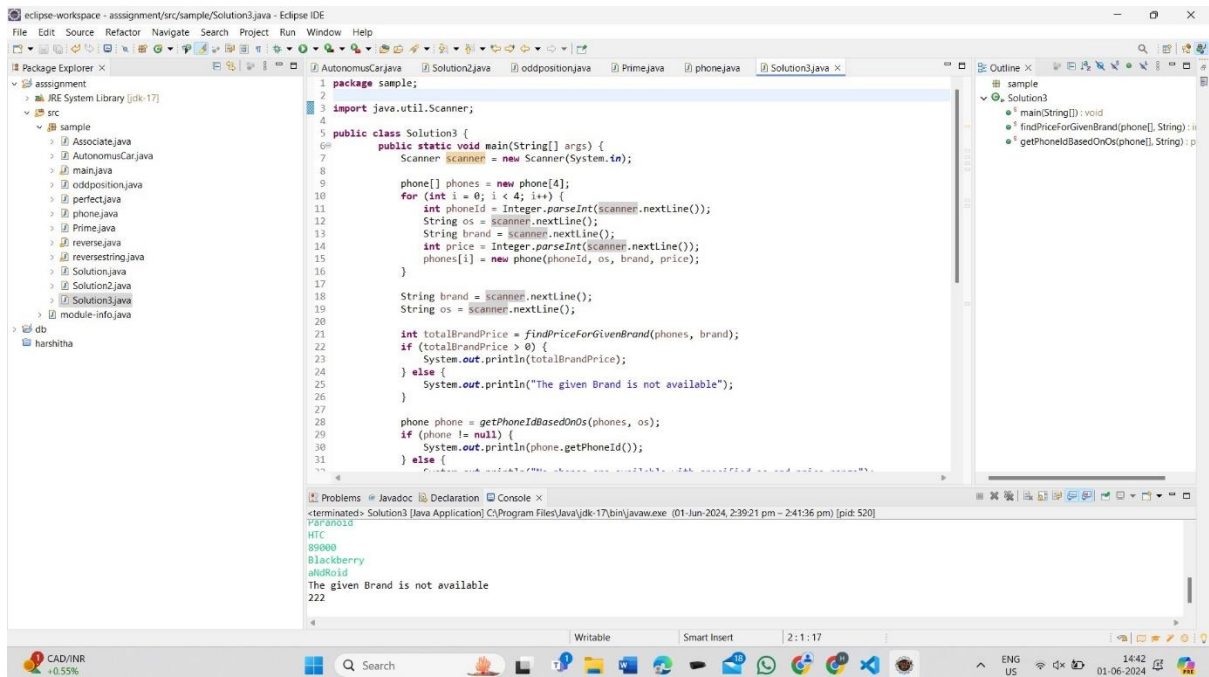


5. device management



The screenshot shows the Eclipse IDE with the console window open. The console displays the output of a Java application. The output is as follows:

```
111
101
Apple
30000
222
android
Samsung
50000
333
Symbian
HTC
12000
444
Paranoid
HTC
89000
Blackberry
aNdroid
The given Brand is not available
222
```



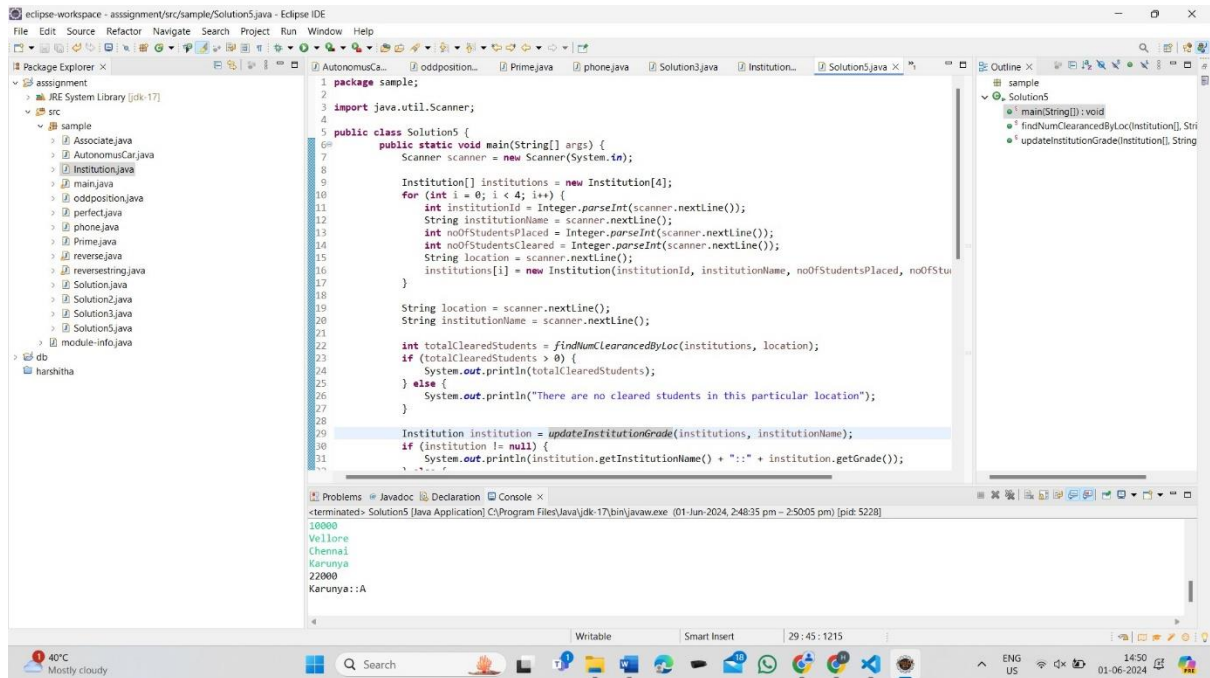
The screenshot shows the Eclipse IDE with the source code of `Solution3.java` open. The code is as follows:

```
1 package sample;
2
3 import java.util.Scanner;
4
5 public class Solution3 {
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner(System.in);
8
9         phone[] phones = new phone[4];
10        for (int i = 0; i < 4; i++) {
11            int phoneId = Integer.parseInt(scanner.nextLine());
12            String os = scanner.nextLine();
13            String brand = scanner.nextLine();
14            int price = Integer.parseInt(scanner.nextLine());
15            phones[i] = new phone(phoneId, os, brand, price);
16        }
17
18        String brand = scanner.nextLine();
19        String os = scanner.nextLine();
20
21        int totalBrandPrice = findPriceForGivenBrand(phones, brand);
22        if (totalBrandPrice > 0) {
23            System.out.println(totalBrandPrice);
24        } else {
25            System.out.println("The given Brand is not available");
26        }
27
28        phone phone = getPhoneIdBasedOnOs(phones, os);
29        if (phone != null) {
30            System.out.println(phone.getPhoneId());
31        } else {
32            System.out.println("The given OS is not available");
33        }
34    }
35}
```

The console window shows the output of the application, which is the same as the one in the first screenshot:

```
Paranoid
HTC
89000
Blackberry
aNdroid
The given Brand is not available
222
```

6.institution

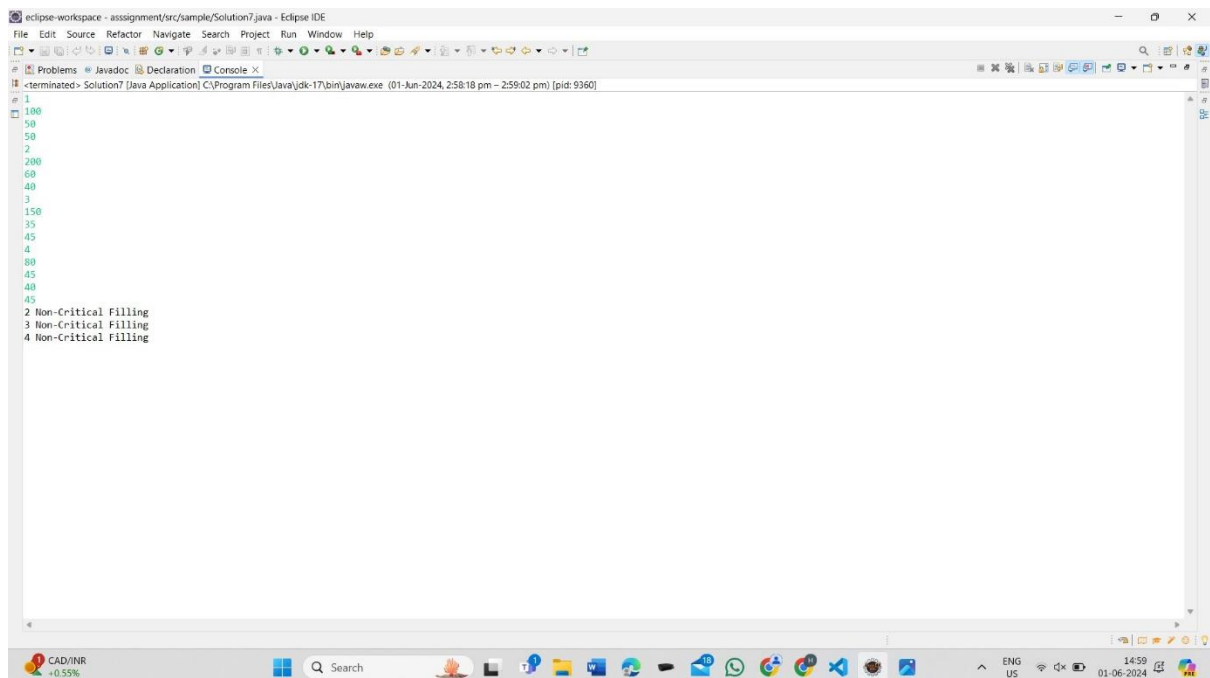


```
1 package sample;
2
3 import java.util.Scanner;
4
5 public class Solution5 {
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner(System.in);
8
9         Institution[] institutions = new Institution[4];
10        for (int i = 0; i < 4; i++) {
11            int institutionId = Integer.parseInt(scanner.nextLine());
12            String institutionName = scanner.nextLine();
13            int noOfStudentsPlaced = Integer.parseInt(scanner.nextLine());
14            int noOfStudentsCleared = Integer.parseInt(scanner.nextLine());
15            String location = scanner.nextLine();
16            institutions[i] = new Institution(institutionId, institutionName, noOfStudentsPlaced, noOfStudentsCleared, location);
17        }
18
19        String location = scanner.nextLine();
20        String institutionName = scanner.nextLine();
21
22        int totalClearedStudents = findNumClearedByLoc(institutions, location);
23        if (totalClearedStudents > 0) {
24            System.out.println(totalClearedStudents);
25        } else {
26            System.out.println("There are no cleared students in this particular location");
27        }
28
29        Institution institution = updateInstitutionGrade(institutions, institutionName);
30        if (institution != null) {
31            System.out.println(institution.getInstitutionName() + " :: " + institution.getGrade());
32        }
33    }
34}
```

Console Output:

```
<terminated> Solution5 [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (01-Jun-2024, 2:48:35 pm - 2:50:05 pm) [pid: 5228]
10000
Vellore
Chennai
Karunya
22000
Karunya::A
```

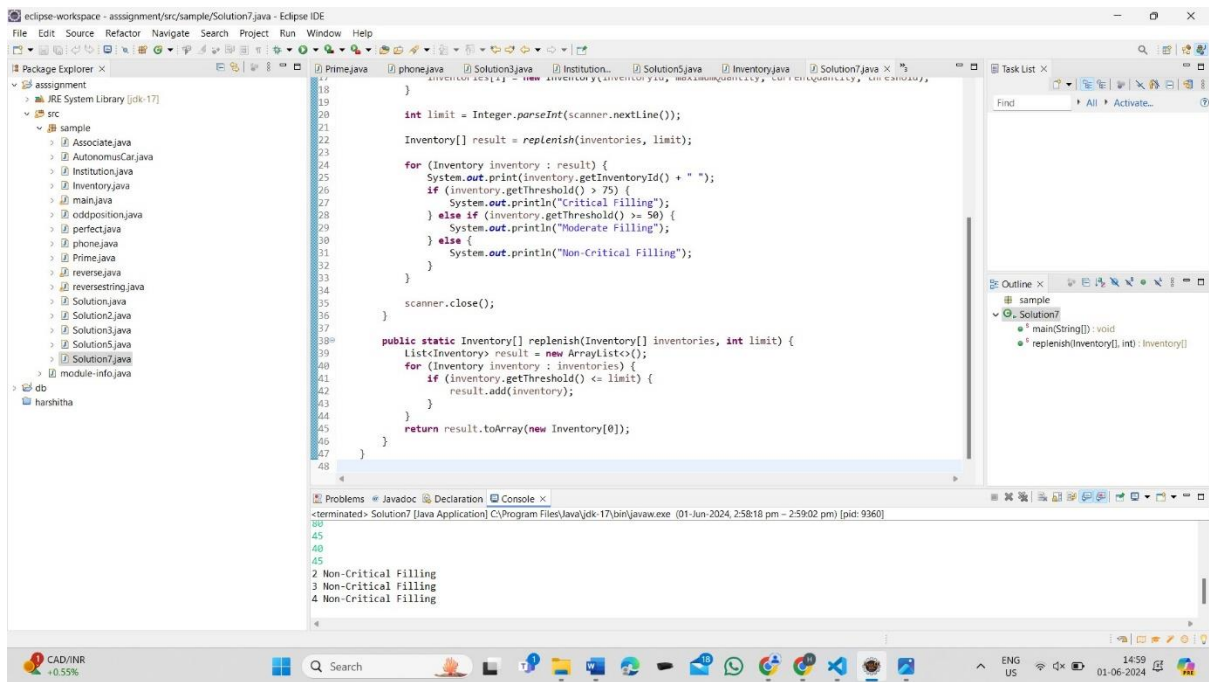
7.inventory replenish



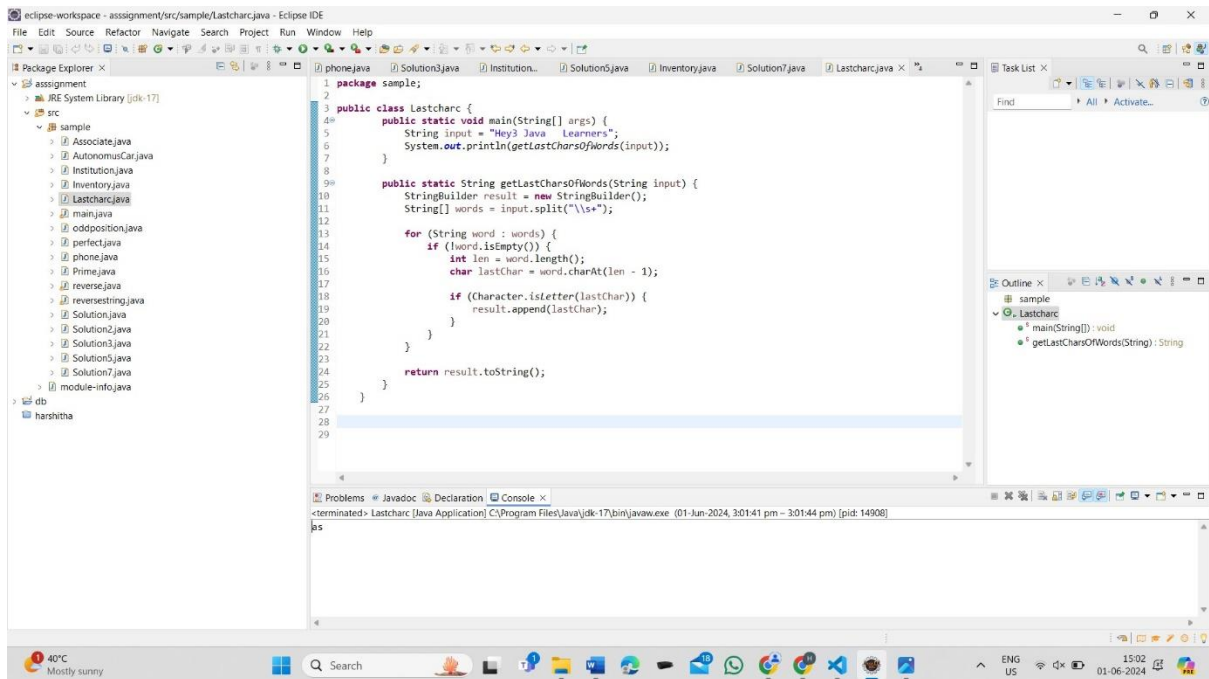
```
1 package sample;
2
3 import java.util.Scanner;
4
5 public class Solution7 {
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner(System.in);
8
9         Inventory[] inventory = new Inventory[4];
10        for (int i = 0; i < 4; i++) {
11            int productId = Integer.parseInt(scanner.nextLine());
12            String productName = scanner.nextLine();
13            int quantity = Integer.parseInt(scanner.nextLine());
14            double price = Double.parseDouble(scanner.nextLine());
15            inventory[i] = new Inventory(productId, productName, quantity, price);
16        }
17
18        String productId = scanner.nextLine();
19        String productName = scanner.nextLine();
20
21        Inventory inventoryItem = replenishInventory(inventory, productId, productName);
22        if (inventoryItem != null) {
23            System.out.println(inventoryItem.getProductName() + " :: " + inventoryItem.getQuantity());
24        }
25    }
26}
```

Console Output:

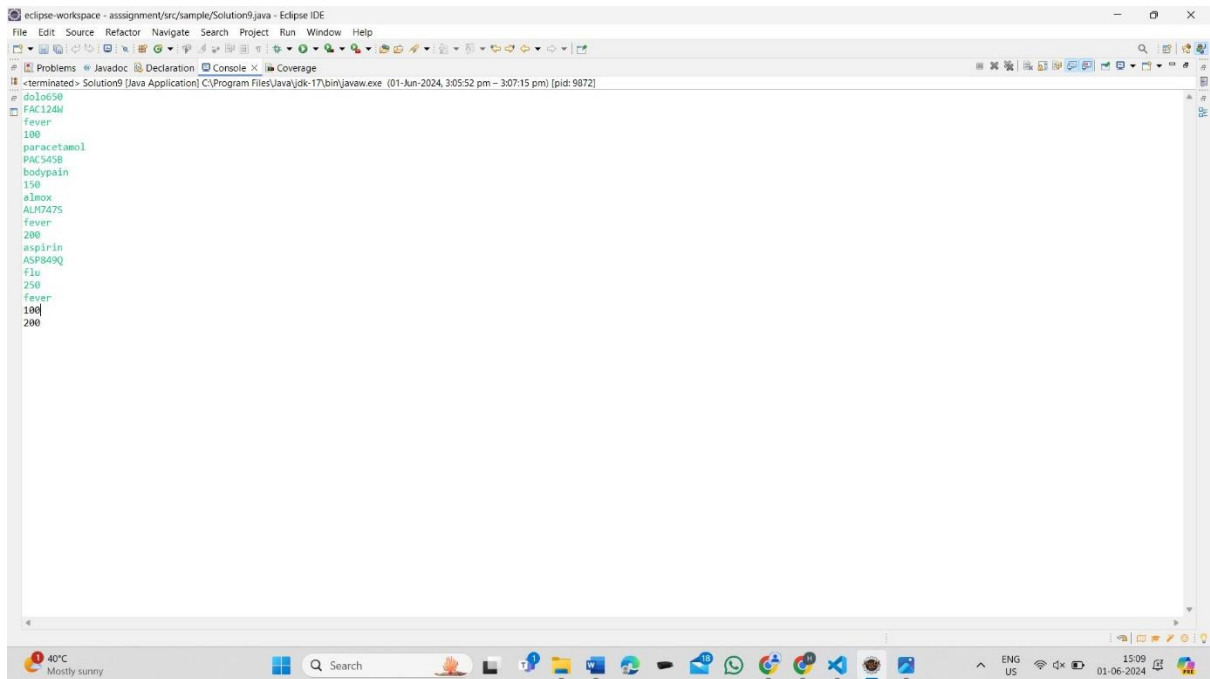
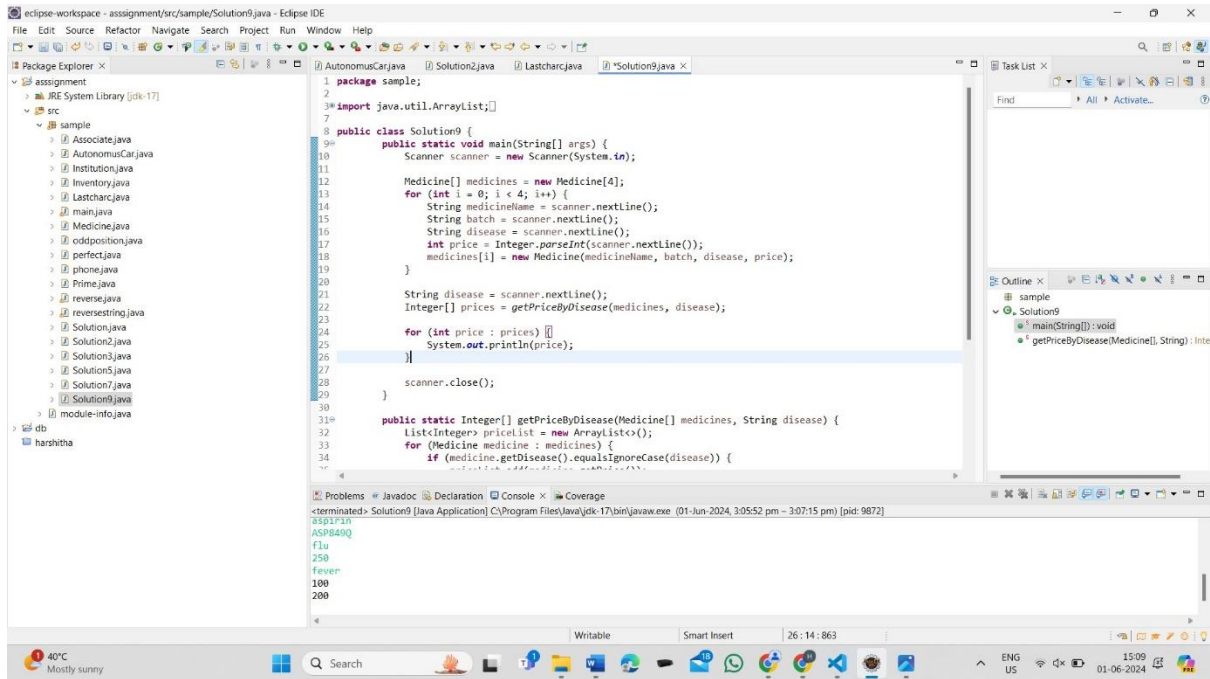
```
<terminated> Solution7 [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (01-Jun-2024, 2:58:18 pm - 2:59:02 pm) [pid: 9360]
100
50
50
2
200
69
40
3
150
35
45
4
80
45
40
45
2 Non-Critical Filling
3 Non-Critical Filling
4 Non-Critical Filling
```



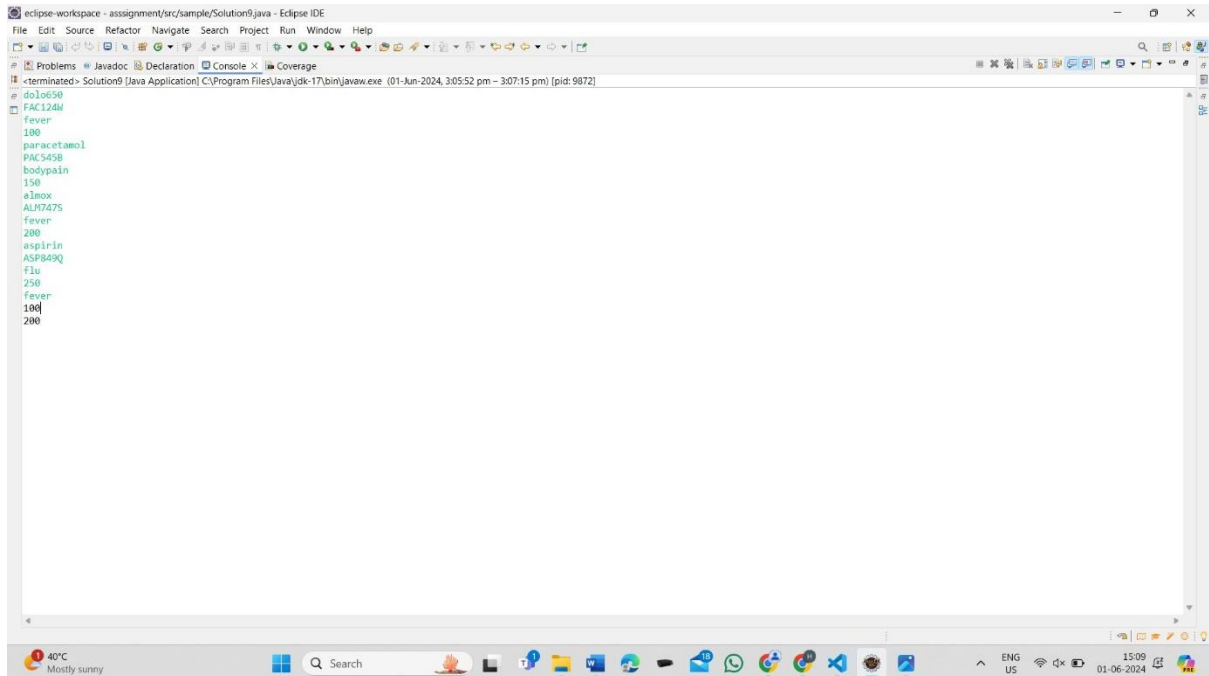
8.last character in every word



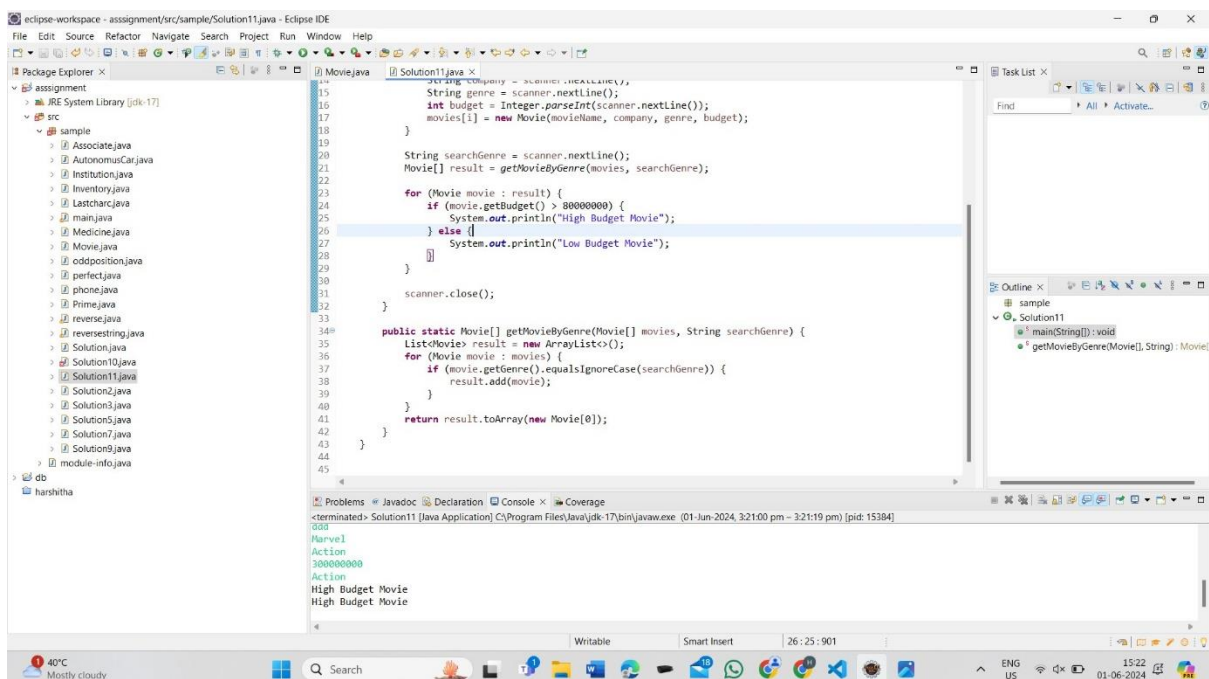
9.medicine get by price



11 movie get by genre

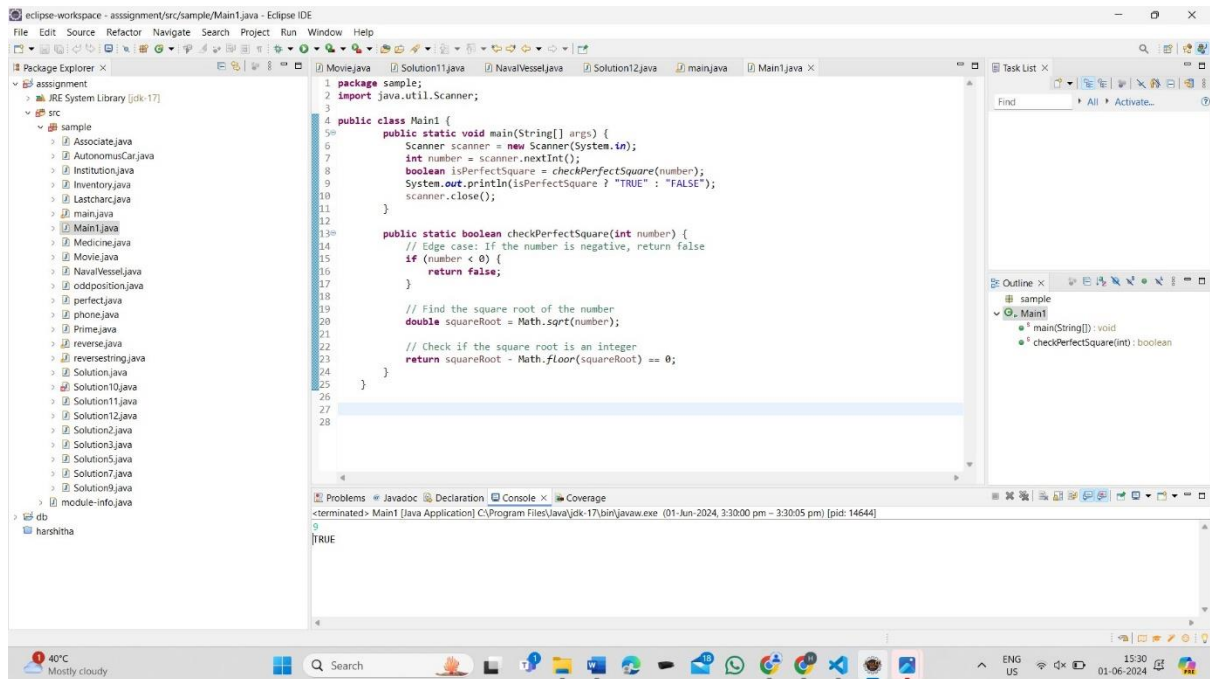


```
eclipse-workspace - assignment/src/sample/Solution9.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
-terminated> Solution9 [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (01-Jun-2024, 3:05:52 pm - 3:07:15 pm) [pid: 9872]
do1o650
FAC124M
Fever
100
paracetamol
PAC545B
bodypain
150
alaox
ALH247S
Fever
200
aspirin
ASP849Q
flu
250
Fever
100
200
```

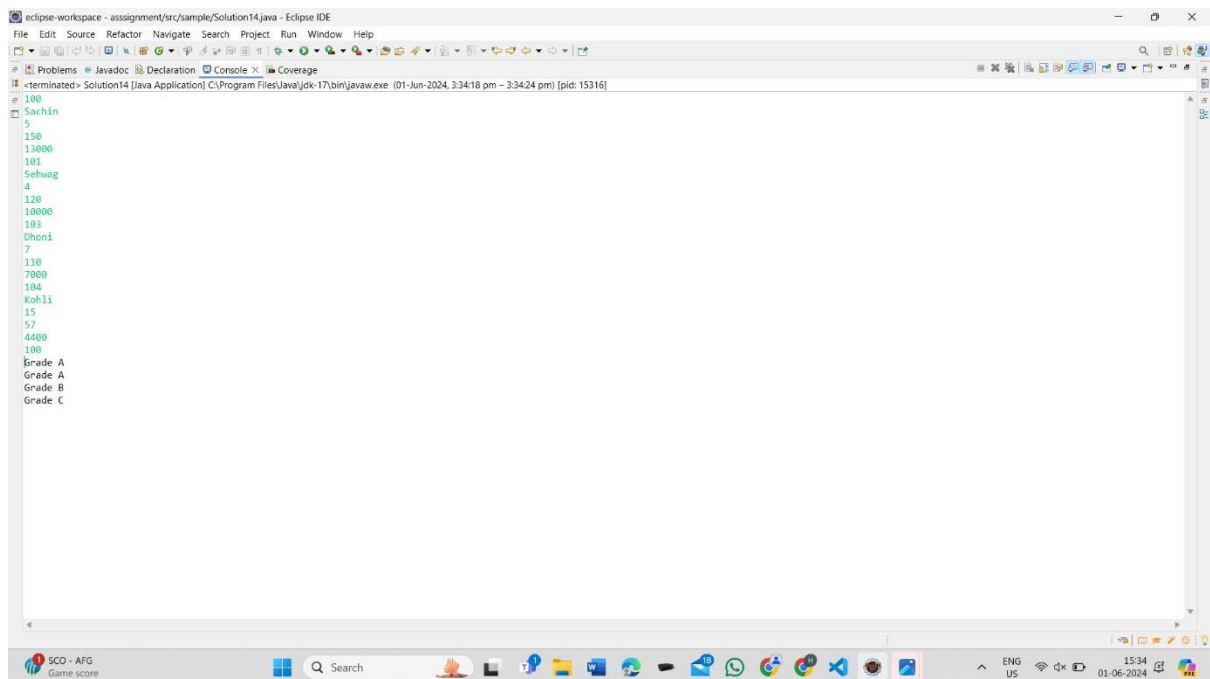


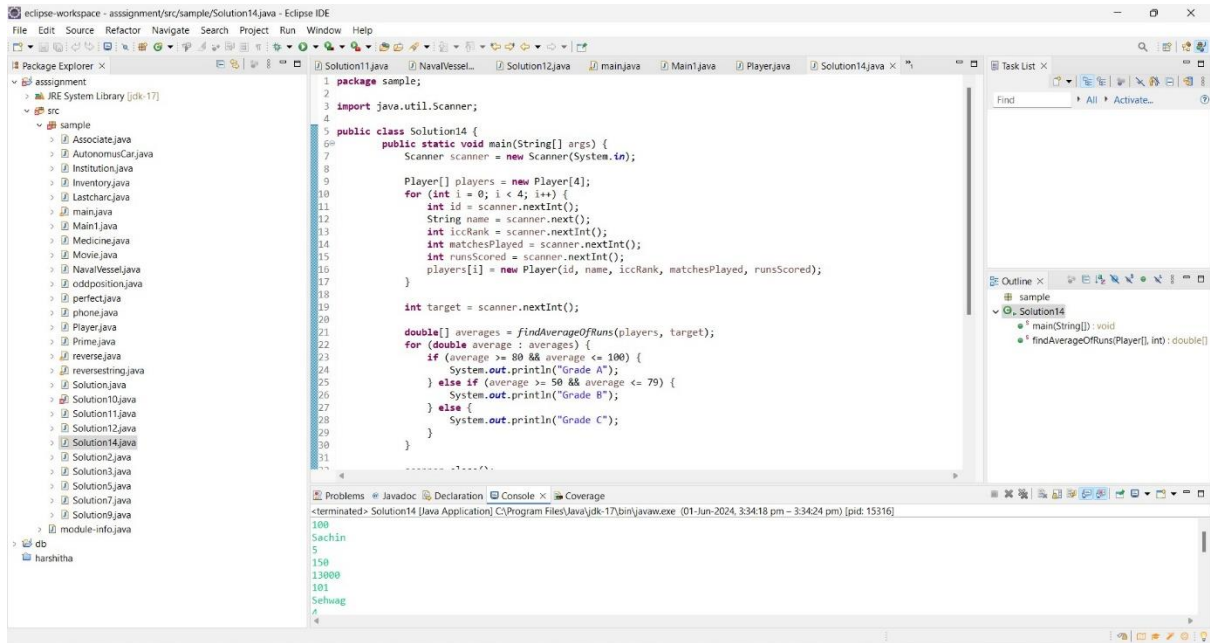
```
eclipse-workspace - assignment/src/sample/Solution11.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
assignment
  JRE System Library [jdk-17]
  src
    sample
      Associate.java
      AutonomousCar.java
      Institution.java
      Inventory.java
      Lastchar.java
      main.java
      Medicine.java
      Movie.java
      oddposition.java
      perfect.java
      phone.java
      Prime.java
      reverse.java
      reversestring.java
      Solution.java
      Solution10.java
      Solution11.java
      Solution2.java
      Solution3.java
      Solution5.java
      Solution7.java
      Solution9.java
      module-info.java
  db
    harshitha
Movie.java
Solution11.java
14 String company = scanner.nextLine();
15 String genre = scanner.nextLine();
16 int budget = Integer.parseInt(scanner.nextLine());
17 movies[i] = new Movie(movieName, company, genre, budget);
18 }
19
20 String searchGenre = scanner.nextLine();
21 Movie[] result = getMovieByGenre(movies, searchGenre);
22
23 for (Movie movie : result) {
24     if (movie.getBudget() > 80000000) {
25         System.out.println("High Budget Movie");
26     } else {
27         System.out.println("Low Budget Movie");
28     }
29 }
30 scanner.close();
31
32
33
34 public static Movie[] getMovieByGenre(Movie[] movies, String searchGenre) {
35     List<Movie> result = new ArrayList<>();
36     for (Movie movie : movies) {
37         if (movie.getGenre().equalsIgnoreCase(searchGenre)) {
38             result.add(movie);
39         }
40     }
41     return result.toArray(new Movie[0]);
42 }
43
44
45
Problems Javadoc Declaration Console Coverage
-terminated> Solution11 [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (01-Jun-2024, 3:21:00 pm - 3:21:19 pm) [pid: 15384]
004
Harve]
Action
300000000
High Budget Movie
High Budget Movie
```

13. perfect number

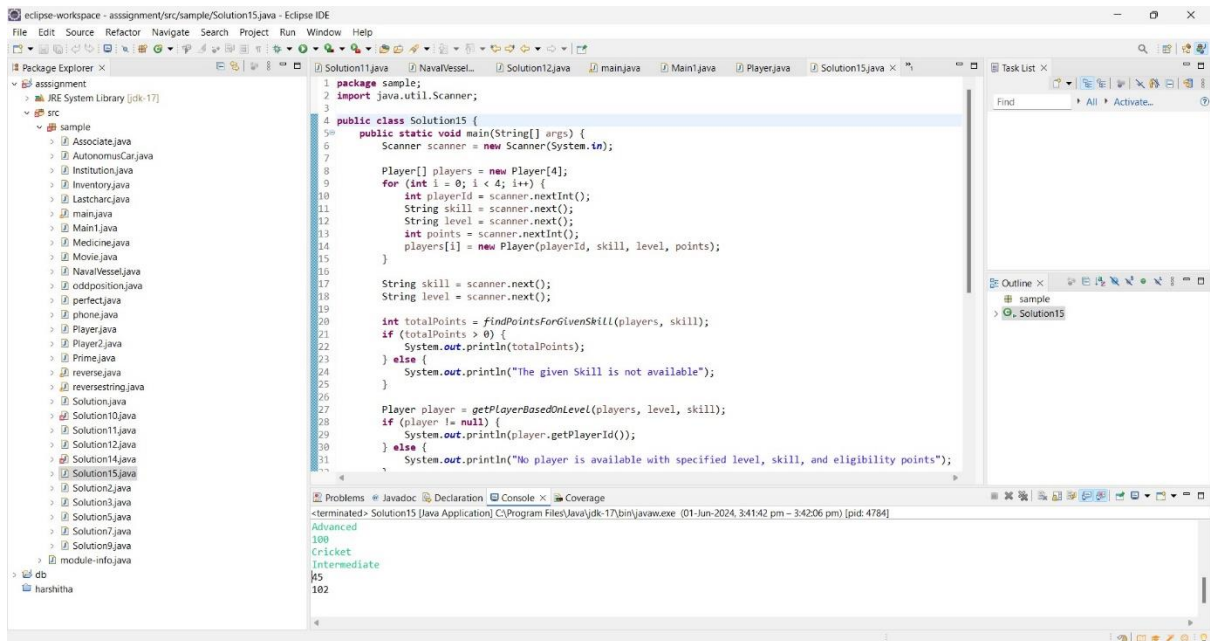


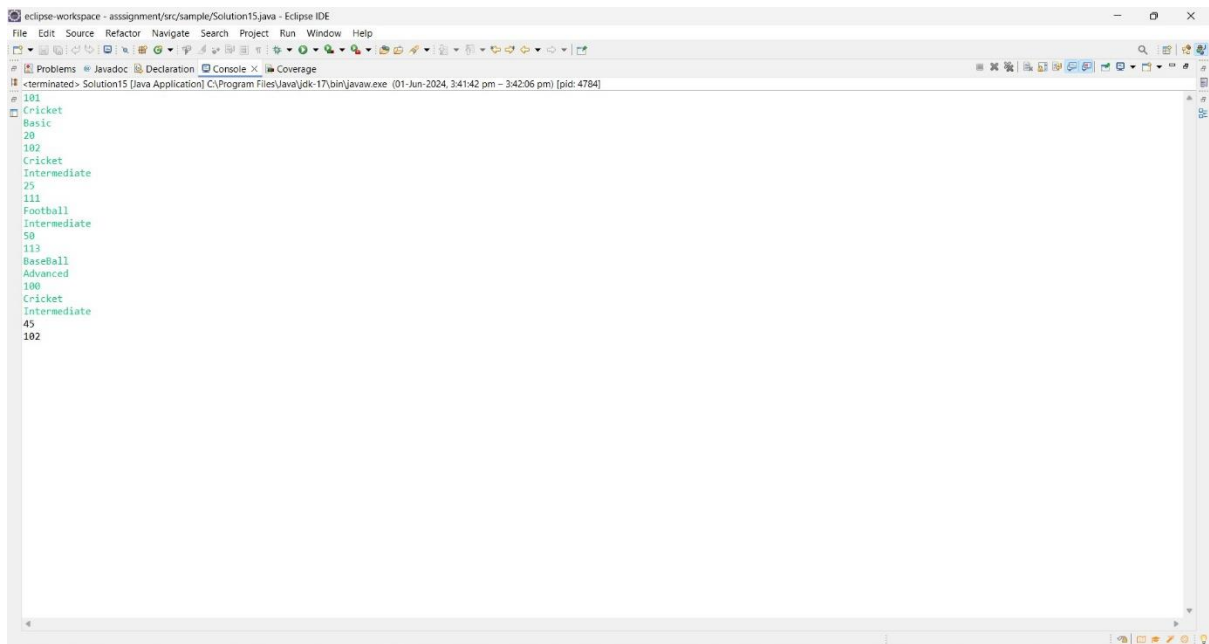
14. player average runs



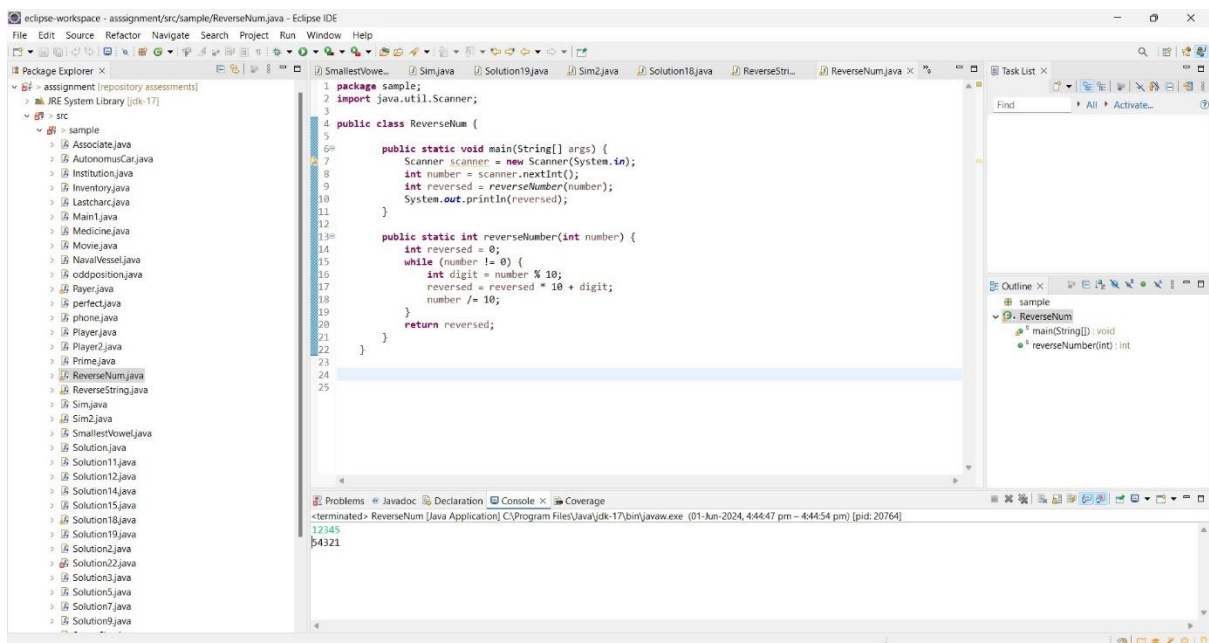


15 playerr





16. Reverse of Number



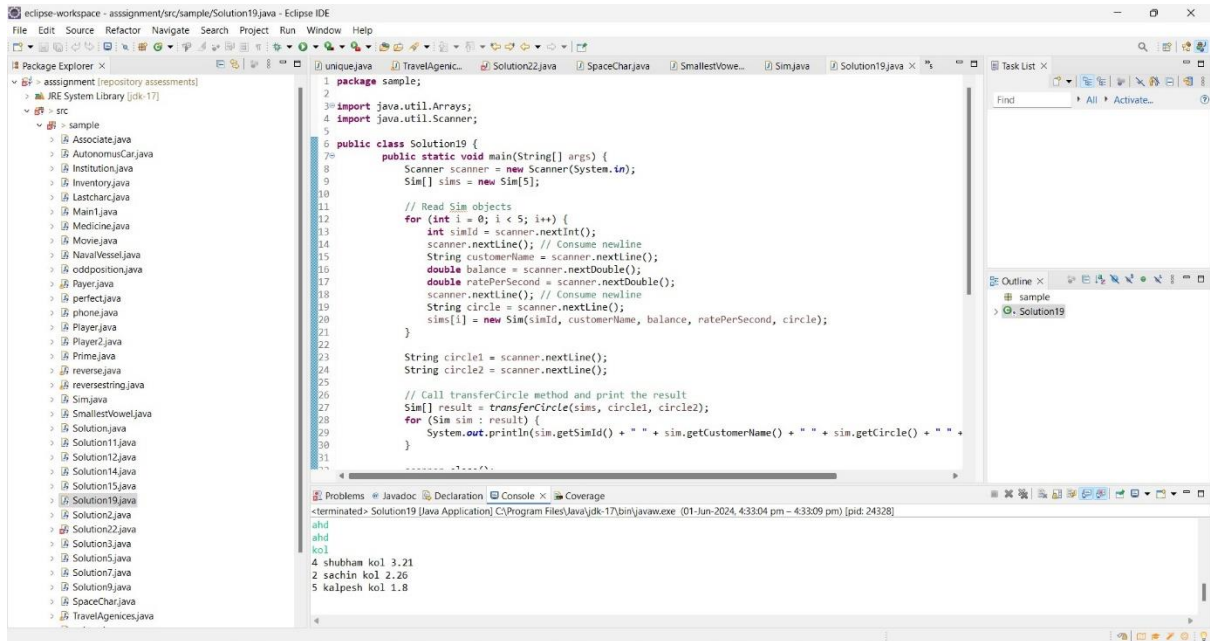
17 reverse string

```
1 package sample;
2 import java.util.Scanner;
3 class ReverseString {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         String input = scanner.nextLine();
7         String reversed = reverseString(input);
8         System.out.println(reversed);
9     }
10
11     public static String reverseString(String input) {
12         StringBuilder reversed = new StringBuilder();
13         for (int i = input.length() - 1; i >= 0; i--) {
14             char ch = input.charAt(i);
15             if (Character.isLetter(ch)) {
16                 reversed.append(Character.toLowerCase(ch));
17             }
18         }
19         return reversed.toString();
20     }
21 }
22
23
24
```

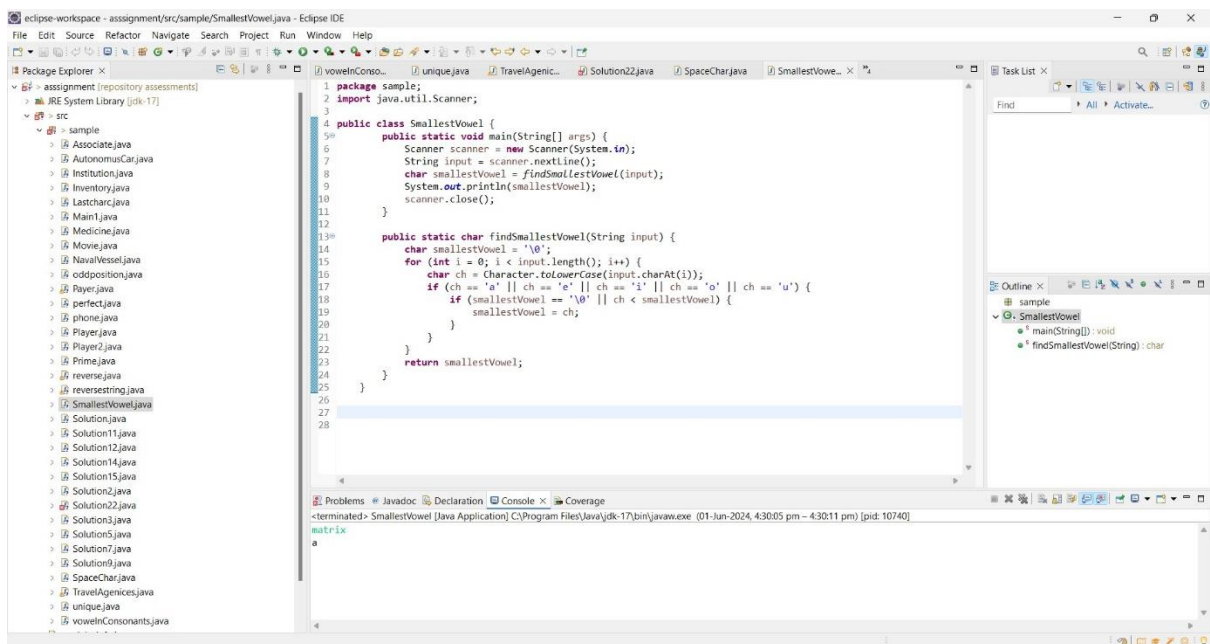
18 sim match n sort

```
2
3 import java.util.ArrayList;
4 import java.util.Collections;
5 import java.util.Comparator;
6 import java.util.List;
7 import java.util.Scanner;
8
9 public class Solution18 {
10     public static Sim2[] matchAndSort(Sim2[] sims, String search_circle, double search_rate) {
11         List<Sim2> resultList = new ArrayList<>();
12
13         for (Sim2 sim : sims) {
14             if (sim.getCircle().equalsIgnoreCase(search_circle) && sim.getRatePerSecond() < search_rate) {
15                 resultList.add(sim);
16             }
17         }
18
19         Collections.sort(resultList, Comparator.comparing(Sim2::getBalance).reversed());
20
21         return resultList.toArray(new Sim2[0]);
22     }
23
24     public static void main(String[] args) {
25         Scanner scanner = new Scanner(System.in);
26
27         Sim2[] sims = new Sim2[4];
28
29         for (int i = 0; i < 4; i++) {
30             int id = scanner.nextInt();
31             scanner.nextLine();
32             String company = scanner.nextLine();
33             ...
34         }
35     }
36 }
37
```

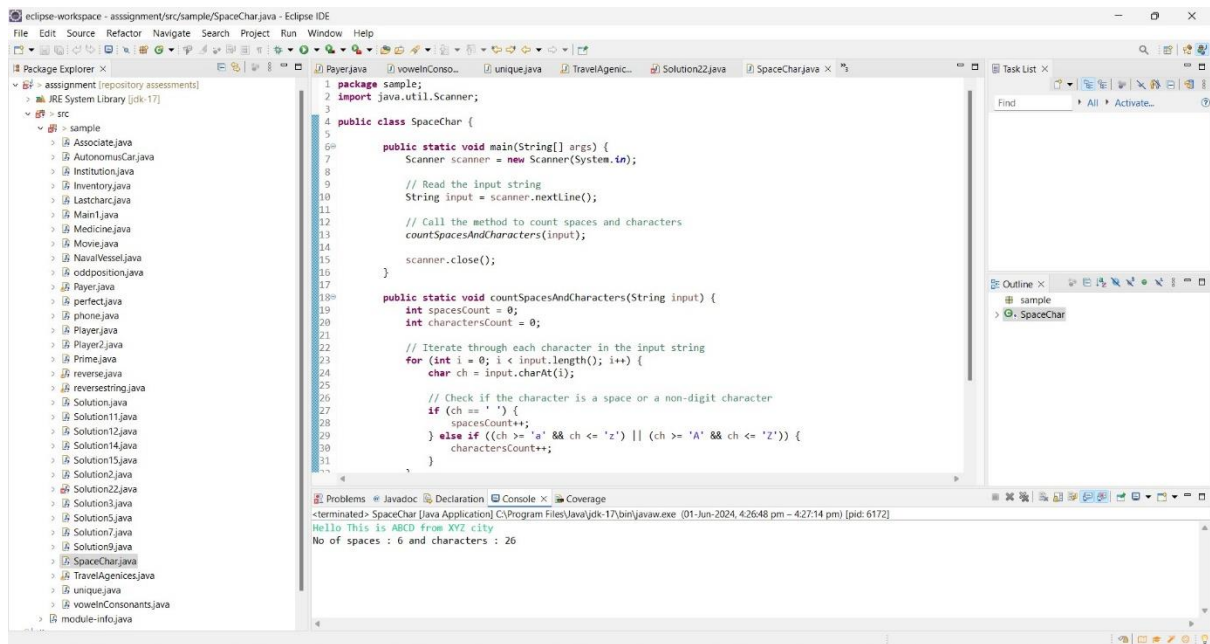
19.sim transfer circle



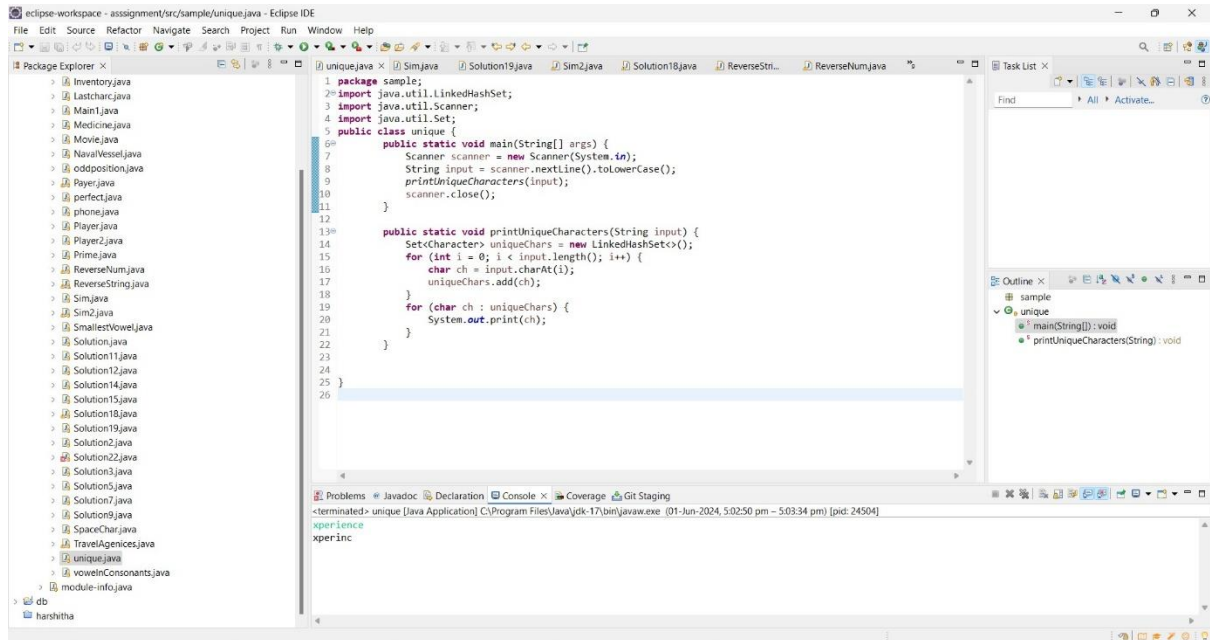
20. smallest vowel



21. spacechar count



23 . unique



24 counting vowel n consonats

The screenshot shows the Eclipse IDE interface. The Package Explorer on the left lists a project named 'assignment' with a source folder 'src' containing a 'sample' package. The 'sample' package contains several Java files, including 'vowelConsonants.java'. The main editor displays the code for 'vowelConsonants.java'.

```
1 package sample;
2 import java.util.Scanner;
3
4 public class vowelConsonants {
5     public static void main(String[] args) {
6         Scanner scanner = new Scanner(System.in);
7
8         System.out.println("Enter the string:");
9         String input = scanner.nextLine();
10
11         countVowelsAndConsonants(input);
12         scanner.close();
13     }
14
15     public static void countVowelsAndConsonants(String input) {
16         input = input.toLowerCase();
17
18         int vowelsCount = 0;
19         int consonantsCount = 0;
20
21         for (int i = 0; i < input.length(); i++) {
22             char ch = input.charAt(i);
23
24             if (Character.isAlphabetic(ch)) {
```

The Outline view on the right shows the class structure with methods 'main(String[]) : void' and 'countVowelsAndConsonants(String) : void'.

The Console view at the bottom shows the program's execution output:

```
<terminated> vowelConsonants [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (01-Jun-2024, 4:15:56 pm - 4:16:14 pm) [pid: 17832]
Enter the string:
Hello world
Vowels count - 3
Consonants count - 7
```