

RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR, THANDALAM – 602 105



**RAJALAKSHMI
ENGINEERING COLLEGE**
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

CS23333-Object Oriented Programming Using Java

Laboratory Record Notebook

NAME : JABARAJ E

REG NO: 230701521

**BRANCH : B.E COMPUTER SCIENCE AND
ENGINEERING**

YEAR : II

SECTION : A

SEMESTER : III

Lab-01-Java Architecture, Language Basics:

Question 1
Correct
Marked out of 5.00
 Flag question

Write a program to find whether the given input number is Odd.
If the given number is odd, the program should return 2 else It should return 1.
Note: The number passed to the program can either be negative, positive or zero. Zero should NOT be treated as Odd.

For example:

Input	Result
123	2
456	1

Answer: (penalty regime: 0 %)

```
1 import java.util.*;
2 public class odd{
3     public static void main(String [] args)
4     {
5         Scanner input = new Scanner(System.in);
6         int n = input.nextInt();
7
8         if (n%2 !=0)
9         {
10            System.out.println("2");
11        }
12        else{
13            System.out.println("1");
14        }
15    }
16
17 }
```

	Input	Expected	Got	
✓	123	2	2	✓
✓	456	1	1	✓

Passed all tests! ✓

Question 2
Correct
Marked out of 5.00
 Flag question

Write a program that returns the last digit of the given number. Last digit is being referred to the least significant digit i.e. the digit in the ones (units) place in the given number.
The last digit should be returned as a positive number.

For example,

if the given number is 197, the last digit is 7

if the given number is -197, the last digit is 7

For example:

Input	Result
197	7
-197	7

Answer: (penalty regime: 0 %)

```
1 import java.util.*;
2 class Least{
3     public static void main(String args[])
4     {
5         Scanner s = new Scanner(System.in);
6         int a= s.nextInt();
7         int abs = Math.abs(a);
8         System.out.print(abs % 10);
9     }
10 }
```

	Input	Expected	Got	
✗	197	7	***Run error*** Error: Could not find or load main class prog Caused by: java.lang.ClassNotFoundException: prog	✗

Testing was aborted due to error.
Your code must pass all tests to earn any marks. Try again.

Show differences

Question 3
Correct
Marked out of
5.00
[Flag question](#)

Rohit wants to add the last digits of two given numbers.
For example,
If the given numbers are 267 and 154, the output should be 11.
Below is the explanation:
Last digit of the 267 is 7
Last digit of the 154 is 4
Sum of 7 and 4 = 11
Write a program to help Rohit achieve this for any given two numbers.
Note: The sign of the input numbers should be ignored.
i.e.
if the input numbers are 267 and 154, the sum of last two digits should be 11
if the input numbers are 267 and -154, the sum of last two digits should be 11
if the input numbers are -267 and 154, the sum of last two digits should be 11
if the input numbers are -267 and -154, the sum of last two digits should be 11

For example:

Input	Result
267	11
154	
267	11
-154	
-267	11
154	
-267	11
-154	

```
1 import java.util.*;
2 import java.math.*;
3
4 public class Main{
5     public static void main(String args[]){
6         int a=0,b=0;
7         Scanner s = new Scanner(System.in);
8         a = s.nextInt();
9         b = s.nextInt();
10        a = Math.abs( a % 10);
11        b = Math.abs(b % 10);
12        System.out.println(a + b);
13
14    }
15 }
16 }
```

	Input	Expected	Got	
✓	267 154	11	11	✓
✓	267 -154	11	11	✓
✓	-267 154	11	11	✓
✓	-267 -154	11	11	✓

Passed all tests! ✓

Lab-02-Flow Control Statements

Question 1

Incorrect

Marked out of
5.00

[Flag question](#)

You and your friend are movie fans and want to predict if the movie is going to be a hit!

The movie's success formula depends on 2 parameters:

the acting power of the actor (range 0 to 10)

the critic's rating of the movie (range 0 to 10)

The movie is a hit if the acting power is excellent (more than 8) or the rating is excellent (more than 8). This holds true except if either the acting power is poor (less than 2) or rating is poor (less than 2), then the movie is a flop. Otherwise the movie is average.

Write a program that takes 2 integers:

the first integer is the acting power

second integer is the critic's rating.

You have to print Yes if the movie is a hit, Maybe if the movie is average and No if the movie is flop.

Example input:

9 5

Output:

Yes

Example input:

1 9

Output:

No

Example input:

6 4

Output:

Maybe

```
3 public class MoviePredictor {  
4  
5     public static void main(String[] args) {  
6         Scanner scanner = new Scanner(System.in);  
7  
8         double actingPower = scanner.nextDouble();  
9         double criticRating = scanner.nextDouble();  
10  
11         // Prediction logic  
12         String prediction = predictMovieHit(actingPower, criticRating);  
13  
14         scanner.close();  
15     }  
16  
17     public static String predictMovieHit(double actingPower, double criticRating) {  
18         if (actingPower > 8 && criticRating < 6) {  
19             return "Yes";  
20         } else if (actingPower < 2 || criticRating < 2) {  
21             return "No";  
22         } else {  
23             return "Maybe";  
24         }  
25     }  
26 }  
27  
28 }  
29 }
```

	Input	Expected	
X	9 5	Yes	X
X	1 9	No	X

Question 2

Correct

Marked out of
5.00[Flag question](#)

You have recently seen a motivational sports movie and want to start exercising regularly. Your coach tells you that it is important to get up early in the morning to exercise. She sets up a schedule for you:

On weekdays (Monday - Friday), you have to get up at 5:00. On weekends (Saturday & Sunday), you can wake up at 6:00. However, if you are on vacation, then you can get up at 7:00 on weekdays and 9:00 on weekends.

Write a program to print the time you should get up.

Input Format

Input containing an integer and a boolean value.

The integer tells you the day it is (1-Sunday, 2-Monday, 3-Tuesday, 4-Wednesday, 5-Thursday, 6-Friday, 7-Saturday). The boolean is true if you are on vacation and false if you're not on vacation.

You have to print the time you should get up.

Example Input:

1 false

Output:

6:00

Example Input:

5 false

Output:

5:00

Example Input:

1 true

Output:

9:00

8 JABARAJ E 2023-CSE-A J

```

1 import java.util.Scanner;
2 public class WakeUpTime{
3     public static void main(String[] args){
4         Scanner sc=new Scanner(System.in);
5         int day=sc.nextInt();
6         boolean isVacation=sc.nextBoolean();
7         String wakeUpTime = "";
8         if(day==1|| day==7){
9             if(isVacation){
10                 wakeUpTime="9:00";
11             }
12         else{
13             wakeUpTime ="6:00";
14         }
15     }
16     else{
17         if(isVacation){
18             wakeUpTime="7:00";
19         }
20         else{
21             wakeUpTime="5:00";
22         }
23     }
24     System.out.println(wakeUpTime);
25     sc.close();
26 }
27 }
```

	Input	Expected	Got	
✓	1 false	6:00	6:00	✓
✓	5 false	5:00	5:00	✓

Question 3

Correct

Marked out of
5.00[Flag question](#)

Write a Java program to input a number from user and print it into words using for loop. How to display number in words using loop in Java programming.

Logic to print number in words in Java programming.

Example**Input**

1234

Output

One Two Three Four

Input:

16

Output:

one six

For example:

Test	Input	Result
1	45	Four Five
2	13	One Three
3	87	Eight Seven

```
1 import java.util.Scanner;
2
3 public class ToWords {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         int number = scanner.nextInt();
7         String numberStr = String.valueOf(number);
8         String[] words = {"Zero", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine"};
9
10        for (int i = 0; i < numberStr.length(); i++) {
11            char digitno = numberStr.charAt(i);
12            int digit = Character.getNumericValue(digitno);
13            System.out.print(words[digit] + " ");
14        }
15        scanner.close();
16    }
17 }
```

	Test	Input	Expected	Got	
✓	1	45	Four Five	Four Five	✓
✓	2	13	One Three	One Three	✓
✓	3	87	Eight Seven	Eight Seven	✓

Passed all tests! ✓

Lab-03-Arrays :

Question 1
Correct
Marked out of
5.00
[Flag question](#)

Given an array of numbers, you are expected to return the sum of the longest sequence of POSITIVE numbers in the array.
If there are NO positive numbers in the array, you are expected to return -1.
In this question's scope, the number 0 should be considered as positive.
Note: If there are more than one group of elements in the array having the longest sequence of POSITIVE numbers, you are expected to return the total sum of all those POSITIVE numbers (see example 3 below).
input1 represents the number of elements in the array.
input2 represents the array of integers.
Example 1:
input1 = 16
input2 = {-12, -16, 12, 18, 18, 14, -4, -12, -13, 32, 34, -5, 66, 78, 78, -79}
Expected output = 62
Explanation:
The input array contains four sequences of POSITIVE numbers, i.e. "12, 18, 18, 14", "12", "32, 34", and "66, 78, 78". The first sequence "12, 18, 18, 14" is the longest of the four as it contains 4 elements. Therefore, the expected output = sum of the longest sequence of POSITIVE numbers = $12 + 18 + 18 + 14 = 62$.
Example 2:
input1 = 11
input2 = {-22, -24, 16, -1, -17, -19, -37, -25, -19, -93, -61}
Expected output = -1
Explanation:
There are NO positive numbers in the input array. Therefore, the expected output for such cases = -1.

```
1 * import java.util.Scanner;
2 *
3 * public class LongestPositiveSequenceSum {
4 *     public static void main(String[] args) {
5 *         Scanner scanner = new Scanner(System.in);
6 *
7 *         // Read first input (number of elements in the array)
8 *         int input1 = scanner.nextInt();
9 *         int[] input2 = new int[input1];
10 *
11 *         // Read the array elements
12 *         for (int i = 0; i < input1; i++) {
13 *             input2[i] = scanner.nextInt();
14 *         }
15 *
16 *         // Calculate and print the result
17 *         int result = sumOfLongestPositiveSequence(input1, input2);
18 *         System.out.println(result);
19 *     }
20 *
21 *     public static int sumOfLongestPositiveSequence(int n, int[] arr) {
22 *         int maxLength = 0;
23 *         int currentLength = 0;
24 *         int currentSum = 0;
25 *         int maxSum = 0;
26 *
27 *         for (int num : arr) {
28 *             if (num >= 0) { // Consider 0 as positive
29 *                 currentLength++;
30 *                 currentSum += num;
31 *             } else {
32 *                 if (currentLength > maxLength) {
33 *                     maxLength = currentLength;
34 *                     maxSum = currentSum;
35 *                 } else if (currentLength == maxLength) {
36 *                     maxSum += currentSum;
37 *                 }
38 *                 currentLength = 0;
39 *                 currentSum = 0;
40 *             }
41 *         }
42 *
43 *         // Final check in case the array ends with a positive sequence
44 *         if (currentLength > maxLength) {
45 *             maxSum = currentSum;
46 *         } else if (currentLength == maxLength) {
47 *             maxSum += currentSum;
48 *         }
49 *
50 *         return maxLength == 0 ? -1 : maxSum;
51 *     }
52 * }
53 *
```

	Input	Expected	Got	
✓	16 -12 -16 12 18 18 14 -4 -12 -13 32 34 -5 66 78 78 -79	62	62	✓
✓	11 -22 -24 -16 -1 -17 -19 -37 -25 -19 -93 -61	-1	-1	✓
✓	16 -58 32 26 92 -10 -4 12 0 12 -2 4 32 -9 -7 78 -79	174	174	✓

Passed all tests! ✓

Question 2
Correct
Marked out of 5.00
[Flag question](#)

Given an integer array as input, perform the following operations on the array, in the below specified sequence.

1. Find the maximum number in the array.
2. Subtract the maximum number from each element of the array.
3. Multiply the maximum number (found in step 1) to each element of the resultant array.

After the operations are done, return the resultant array.

Example 1:

input1 = 4 (represents the number of elements in the input1 array)

input2 = {1, 5, 6, 9}

Expected Output = {-72, -36, 27, 0}

Explanation:

Step 1: The maximum number in the given array is 9.

Step 2: Subtracting the maximum number 9 from each element of the array:

{(1 - 9), (5 - 9), (6 - 9), (9 - 9)} = {-8, -4, -3, 0}

Step 3: Multiplying the maximum number 9 to each of the resultant array:

{(-8 x 9), (-4 x 9), (3 x 9), (0 x 9)} = {-72, -36, -27, 0}

So, the expected output is the resultant array {-72, -36, -27, 0}.

Example 2:

input1 = 5 (represents the number of elements in the input1 array)

input2 = {10, 87, 63, 42, 2}

Expected Output = {-6699, 0, -2088, -3915, -7395}

Explanation:

Step 1: The maximum number in the given array is 87.

Step 2: Subtracting the maximum number 87 from each element of the array:

{(10 - 87), (87 - 87), (63 - 87), (42 - 87), (2 - 87)} = {-77, 0, -24, -45, -85}

Step 3: Multiplying the maximum number 87 to each of the resultant array:

{(-77 x 87), (0 x 87), (-24 x 87), (-45 x 87), (-85 x 87)} = {-6699, 0, -2088, -3915, -7395}

So, the expected output is the resultant array {-6699, 0, -2088, -3915, -7395}.

```

1 * import java.util.Scanner;
2
3 public class ArrayOperations {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6
7         // Read the number of elements in the array
8         int input1 = scanner.nextInt();
9         int[] input2 = new int[input1];
10
11        // Read the array elements
12        for (int i = 0; i < input1; i++) {
13            input2[i] = scanner.nextInt();
14        }
15
16        // Perform operations and get the resultant array
17        int[] result = performOperations(input2);
18
19        // Print the resultant array
20        for (int num : result) {
21            System.out.print(num + " ");
22        }
23        System.out.println();
24    }
25
26    public static int[] performOperations(int[] arr) {
27        // Step 1: Find the maximum number in the array
28        int maxNum = Integer.MIN_VALUE;
29        for (int num : arr) {
30            if (num > maxNum) {
31                maxNum = num;
32            }
33        }
34
35        // Step 2: Subtract the maximum number from each element
36        int[] subtractedArray = new int[arr.length];
37        for (int i = 0; i < arr.length; i++) {
38            subtractedArray[i] = arr[i] - maxNum;
39        }
40
41        // Step 3: Multiply the maximum number to each element of the resultant array
42        int[] resultArray = new int[subtractedArray.length];
43        for (int i = 0; i < subtractedArray.length; i++) {
44            resultArray[i] = subtractedArray[i] * maxNum;
45        }
46
47        return resultArray;
48    }
49 }
```

	Input	Expected	Got	
✓	4 1 5 6 9	-72 -36 -27 0	-72 -36 -27 0	✓
✓	5 10 87 63 42 2	-6699 0 -2088 -3915 -7395	-6699 0 -2088 -3915 -7395	✓
✓	2 -9 9	-162 0	-162 0	✓

Passed all tests! ✓

Question 3
Correct
Marked out of 5.00
[Flag question](#)

You are provided with a set of numbers (array of numbers).
You have to generate the sum of specific numbers based on its position in the array set provided to you.

This is explained below:

Example 1:

Let us assume the encoded set of numbers given to you is:

input1:5 and input2: {1, 51, 436, 7860, 41236}

Step 1:

Starting from the 0th index of the array pick up digits as per below:

0th index - pick up the units value of the number (in this case is 1).

1st index - pick up the tens value of the number (in this case it is 5).

2nd index - pick up the hundreds value of the number (in this case it is 4).

3rd index - pick up the thousands value of the number (in this case it is 7).

4th index - pick up the ten thousands value of the number (in this case it is 4).

(Continue this for all the elements of the input array).

The array generated from Step 1 will then be – {1, 5, 4, 7, 4}.

Step 2:

Square each number present in the array generated in Step 1.

{1, 25, 16, 49, 16}

Step 3:

Calculate the sum of all elements of the array generated in Step 2 to get the final result. The result will be = 107.

Note:

- 1) While picking up a number in Step1, if you observe that the number is smaller than the required position then use 0.
- 2) In the given function, input1[] is the array of numbers and input2 represents the number of elements in input1.

```

1 * import java.util.Scanner;
2 *
3 * public class SumOfSpecificNumbers {
4 *     public static void main(String[] args) {
5 *         Scanner scanner = new Scanner(System.in);
6 *
7 *         // Read the number of elements in the array
8 *         int input1 = scanner.nextInt();
9 *         int[] input2 = new int[input1];
10 *
11 *         // Read the array elements
12 *         for (int i = 0; i < input1; i++) {
13 *             input2[i] = scanner.nextInt();
14 *         }
15 *
16 *         // Calculate the result
17 *         int result = calculateSum(input2);
18 *         System.out.println(result);
19 *     }
20 *
21 *     public static int calculateSum(int[] arr) {
22 *         int[] digitArray = new int[arr.length];
23 *
24 *         // Step 1: Extract digits based on their positions
25 *         for (int i = 0; i < arr.length; i++) {
26 *             int number = arr[i];
27 *             digitArray[i] = getDigitAtPosition(number, i);
28 *         }
29 *
30 *         // Step 2: Square each number in the digitArray
31 *         int sumOfSquares = 0;
32 *         for (int digit : digitArray) {
33 *             sumOfSquares += digit * digit;
34 *         }
35 *
36 *         return sumOfSquares;
37 *     }
38 *
39 *     private static int getDigitAtPosition(int number, int position) {
40 *         // Check if the position is valid for the number
41 *         if (position >= String.valueOf(number).length()) {
42 *             return 0; // If the position is greater than the number of digits, return 0
43 *         }
44 *
45 *         // Extract the digit at the specified position
46 *         for (int j = 0; j < position; j++) {
47 *             number /= 10;
48 *         }
49 *         return number % 10; // Return the last digit (the desired position)
50 *     }
51 * }
```

	Input	Expected	Got	
✓	5 1 51 436 7860 41236	107	107	✓
✓	5 1 5 423 310 61540	53	53	✓

Passed all tests! ✓

Lab-04-Classes and Objects

Question 1
Correct!
Marked out of 5.00
 Flag question

Create a class Student with two private attributes, name and roll number. Create three objects by invoking different constructors available in the class Student.

Student()

Student(String name)

Student(String name, int rollno)

Input:

No input

Output:

No-arg constructor is invoked

1 arg constructor is invoked

2 arg constructor is invoked

Name =null , Roll no = 0

Name =Rajalakshmi , Roll no = 0

Name =Lakshmi , Roll no = 101

```
1 * class Student {  
2     private String name;  
3     private int rollno;  
4  
5     // No-argument constructor  
6     public Student() {  
7         System.out.println("No-arg constructor is invoked");  
8         this.name = null;  
9         this.rollno = 0;  
10    }  
11  
12    // One-argument constructor  
13    public Student(String name) {  
14        System.out.println("1 arg constructor is invoked");  
15        this.name = name;  
16        this.rollno = 0;  
17    }  
18  
19    // Two-argument constructor  
20    public Student(String name, int rollno) {  
21        System.out.println("2 arg constructor is invoked");  
22        this.name = name;  
23        this.rollno = rollno;  
24    }  
25  
26    // Method to display student details  
27    public void display() {  
28        System.out.println("Name = " + name + " , Roll no = " + rollno);  
29    }  
30}  
31  
32    public class TestStudent {  
33        public static void main(String[] args) {  
34  
35            Student student1 = new Student();  
36            Student student2 = new Student("Rajalakshmi");  
37            Student student3 = new Student("Lakshmi", 101);  
38  
39            student1.display();  
40            student2.display();  
41            student3.display();  
42        }  
43    }  
44}  
45}
```

	Test	Expected	Got	
<input checked="" type="checkbox"/>	1	No-arg constructor is invoked 1 arg constructor is invoked 2 arg constructor is invoked Name =null , Roll no = 0 Name =Rajalakshmi , Roll no = 0 Name =Lakshmi , Roll no = 101	No-arg constructor is invoked 1 arg constructor is invoked 2 arg constructor is invoked Name =null , Roll no = 0 Name =Rajalakshmi , Roll no = 0 Name =Lakshmi , Roll no = 101	<input checked="" type="checkbox"/>

Passed all tests! ✓

Question 2

0%
Marked out of
5.00
Flag question

Create a class Mobile with the attributes listed below.

```
private String manufacturer;
private String operatingSystem;
public String color;
private int cost;

Define a Parameterized constructor to initialize the above instance variables.

Define getter and setter methods for the attributes above.

For example: setter method for manufacturer is

void setManufacturer(String manufacturer)

this.manufacturer = manufacturer;
}

String getManufacturer()

return manufacturer;

Display the object details by overriding the toString() method.

See examples:
```

```
1+ public class Mobile {
2   private String manufacturer;
3   private String operatingSystem;
4   public String color;
5   private int cost;
6
7+   public Mobile(String manufacturer, String operatingSystem, String color, int cost) {
8     this.manufacturer = manufacturer;
9     this.operatingSystem = operatingSystem;
10    this.color = color;
11    this.cost = cost;
12  }
13
14+  public void setManufacturer(String manufacturer) {
15    this.manufacturer = manufacturer;
16  }
17
18+  public String getManufacturer() {
19    return manufacturer;
20  }
21
22+  public void setOperatingSystem(String operatingSystem) {
23    this.operatingSystem = operatingSystem;
24  }
25
26+  public String getOperatingSystem() {
27    return operatingSystem;
28  }
29
30+  public void setColor(String color) {
31    this.color = color;
32  }
33
34+  public String getColor() {
35    return color;
36  }
37
38+  public void setCost(int cost) {
39    this.cost = cost;
40  }
41
42+  public int getCost() {
43    return cost;
44  }
45
46  @Override
47+  public String toString() {
48    return "manufacturer = " + manufacturer +
49      "\noperating_system = " + operatingSystem +
50      "\ncolor = " + color +
51      "\ncost = " + cost;
52  }
53
54+  public static void main(String[] args) {
55    Mobile mobile = new Mobile("Redmi", "Andriod", "Blue", 34000);
56    System.out.println(mobile);
57  }
58 }
```

	Test	Expected	Got	
✓	1	manufacturer = Redmi operating_system = Andriod color = Blue cost = 34000	manufacturer = Redmi operating_system = Andriod color = Blue cost = 34000	✓

Passed all tests! ✓

Question 3
Correct
Marked out of
5.00
Flag question

Create a class called "Circle" with a radius attribute. You can access and modify this attribute using getter and setter methods. Calculate the area and circumference of the circle.
Area of Circle = πr^2
Circumference = $2\pi r$
Input:
2
Output:
Area = 12.57
Circumference = 12.57

```
1 import java.util.*;  
2 public class Circle {  
3     public static void main(String args[]){  
4         Scanner a =new Scanner (System.in);  
5         int b=a.nextInt();  
6         double area =Math.PI* (b*b);  
7         double cir=Math.PI*b*2;  
8         System.out.printf("Area = %.2f%n", area);  
9         System.out.printf("Circumference = %.2f%n", cir);  
10    }  
11 }
```

	Test	Input	Expected	Got	
✓	1	4	Area = 50.27 Circumference = 25.13	Area = 50.27 Circumference = 25.13	✓
✓	2	6	Area = 113.10 Circumference = 37.70	Area = 113.10 Circumference = 37.70	✓
✓	3	2	Area = 12.57 Circumference = 12.57	Area = 12.57 Circumference = 12.57	✓

Passed all tests! ✓

Lab-05-Inheritance

Question 1
Correct
Marked out of
5.00
Flag question

Create a class called College with attribute String name, constructor to initialize the name attribute, a method called Admitted(), Create a subclass called CSE that extends Student class, with department attribute, Course() method to sub class. Print the details of the Student.

College:

```
String collegeName;
public College() {}
public admitted() {}
```

Student:

```
String studentName;
String department;
public Student(String collegeName, String studentName, String depart) {}
public toString()
```

Expected Output:

```
A student admitted in REC
CollegeName : ITC
StudentName : Venkatesh
Department : CSE
```

```
1 * class Student extends College{
2
3     String studentName;
4     String department;
5
6     public Student(String collegeName, String studentName, String department) {
7         super(collegeName);
8         this.studentName=studentName;
9         this.department=department;
10    }
11 }
12
13     public String toString(){
14         return "CollegeName : "+collegeName+"\n"+ "StudentName : "+studentName+"\n"+ "Department : "+department;
15     }
16 }
17
18
19
20 class College
21 {
22     public String collegeName;
23
24     public College(String collegeName) {
25         this.collegeName=collegeName;
26     }
27
28     public void admitted() {
29         System.out.println("A student admitted in "+collegeName);
30     }
31 }
32
33 public class Main {
34     public static void main (String[] args) {
35         Student s1 = new Student("REC", "Venkatesh", "CSE");
36         s1.admitted();
37     }
38 }
```

	Expected	Got	
✓	A student admitted in REC CollegeName : REC StudentName : Venkatesh Department : CSE	A student admitted in REC CollegeName : REC StudentName : Venkatesh Department : CSE	✓

Passed all tests! ✓

Question 2
Correct
Marked out of 5.00
 Flag question

Create a class Mobile with constructor and a method basicMobile().
Create a subclass CameraMobile which extends Mobile class , with constructor and a method newFeature().
Create a subclass AndroidMobile which extends CameraMobile,with constructor and a method androidMobile().
display the details of the Android Mobile class by creating the instance. .

```
class Mobile{  
}  
class CameraMobile extends Mobile {  
}  
class AndroidMobile extends CameraMobile {  
}  
expected output:  
Basic Mobile is Manufactured  
Camera Mobile is Manufactured  
Android Mobile is Manufactured  
Camera Mobile with 5MG px  
Touch Screen Mobile is Manufactured
```

```
2 * class Mobile {  
3 *     // Constructor  
4 *     public Mobile() {  
5 *         System.out.println("Basic Mobile is Manufactured");  
6 *     }  
7 *  
8 *     // Method for basic mobile features  
9 *     public void basicMobile() {  
10 *         System.out.println("Basic mobile features are available.");  
11 *     }  
12 * }  
13 *  
14 * // Subclass CameraMobile that extends Mobile  
15 * class CameraMobile extends Mobile {  
16 *     // Constructor  
17 *     public CameraMobile() {  
18 *         super(); // Call the parent constructor  
19 *         System.out.println("Camera Mobile is Manufactured");  
20 *     }  
21 *  
22 *     // Method for camera features  
23 *     public void newFeature() {  
24 *         System.out.println("Camera Mobile with 5MG px");  
25 *     }  
26 * }  
27 *  
28 * // Subclass AndroidMobile that extends CameraMobile  
29 * class AndroidMobile extends CameraMobile {  
30 *     // Constructor  
31 *     public AndroidMobile() {  
32 *         super(); // Call the parent constructor  
33 *         System.out.println("Android Mobile is Manufactured");  
34 *     }  
35 *  
36 *     // Method for Android mobile features  
37 *     public void androidMobile() {  
38 *         System.out.println("Touch Screen Mobile is Manufactured");  
39 *     }  
40 * }  
41 *  
42 * // Test class to display details of Android Mobile  
43 * public class TestMobile {  
44 *     public static void main(String[] args) {  
45 *         // Create an instance of AndroidMobile  
46 *         AndroidMobile androidMobile = new AndroidMobile();  
47 *  
48 *         // Call the newFeature and androidMobile methods  
49 *         androidMobile.newFeature();  
50 *         androidMobile.androidMobile();  
51 *     }  
52 * }
```

	Expected	Got	
✓	Basic Mobile is Manufactured Camera Mobile is Manufactured Android Mobile is Manufactured Camera Mobile with 5MG px Touch Screen Mobile is Manufactured	Basic Mobile is Manufactured Camera Mobile is Manufactured Android Mobile is Manufactured Camera Mobile with 5MG px Touch Screen Mobile is Manufactured	✓

Passed all tests! ✓

Question 3
Correct
Marked out of 5.00
[Flag question](#)

Create a class known as "BankAccount" with methods called deposit() and withdraw().
Create a subclass called SavingsAccount that overrides the withdraw() method to prevent withdrawals if the account balance falls below one hundred.

For example:

Result

```
Create a Bank Account object (A/c No. BA1234) with initial balance of $500:  
Deposit $1000 into account BA1234:  
New balance after depositing $1000: $1500.0  
Withdraw $600 from account BA1234:  
New balance after withdrawing $600: $900.0  
Create a SavingsAccount object (A/c No. SA1000) with initial balance of $300:  
Try to withdraw $250 from SA1000!  
Minimum balance of $100 required!  
Balance after trying to withdraw $250: $300.0
```

```
1+ class BankAccount {  
2+     private String accountNumber;  
3+     private double balance;  
4+     public BankAccount(String accountNumber, double balance){  
5+         this.accountNumber=accountNumber;  
6+         this.balance=balance;  
7+     }  
8+     public void deposit(double amount) {  
9+         balance+=amount;  
10+    }  
11+  
12+    public void withdraw(double amount) {  
13+        if (balance >= amount) {  
14+            balance -= amount;  
15+        } else {  
16+            System.out.println("Insufficient balance");  
17+        }  
18+    }  
19+    public double getBalance() {  
20+        return balance;  
21+    }  
22+    public String getAccountNumber(){  
23+        return accountNumber;  
24+    }  
25+}  
26+ class SavingsAccount extends BankAccount {  
27+     public SavingsAccount(String accountNumber, double balance) {  
28+         super(accountNumber,balance);  
29+     }  
30+     @Override  
31+     public void withdraw(double amount) {  
32+         if (getBalance() - amount < 100) {  
33+             System.out.println("Minimum balance of $100 required!");  
34+         } else {  
35+             super.withdraw(amount);  
36+         }  
37+     }  
38+ }  
39+  
40+ public class Main {  
41+  
42+     public static void main(String[] args) {  
43+         System.out.println("Create a Bank Account object (A/c No. BA1234) with initial balance of $500:");  
44+         BankAccount BA1234 = new BankAccount("BA1234", 500);  
45+         BA1234.deposit(1000);  
46+         System.out.println("Deposit $1000 into account BA1234:");  
47+         System.out.println("New balance after depositing $1000: $" + BA1234.getBalance());  
48+         System.out.println("Withdraw $600 from account BA1234:");  
49+         BA1234.withdraw(600);  
50+         System.out.println("New balance after withdrawing $600: $" + BA1234.getBalance());  
51+  
52+         System.out.println("Create a SavingsAccount object (A/c No. SA1000) with initial balance of $300:");  
53+         SavingsAccount SA1000 = new SavingsAccount("SA1000", 300);  
54+  
55+         System.out.println("Try to withdraw $250 from SA1000!");  
56+         SA1000.withdraw(250);  
57+         System.out.println("Balance after trying to withdraw $250: $" + SA1000.getBalance());  
58+     }  
59+ }
```

Expected	Got	
<p>✓ Create a Bank Account object (A/c No. BA1234) with initial balance of \$500: Deposit \$1000 into account BA1234; New balance after depositing \$1000: \$1500.0 Withdraw \$600 from account BA1234; New balance after withdrawing \$600: \$900.0 Create a SavingsAccount object (A/c No. SA1000) with initial balance of \$300: Try to withdraw \$250 from SA1000! Minimum balance of \$100 required! Balance after trying to withdraw \$250: \$300.0</p>	<p>Create a Bank Account object (A/c No. BA1234) with initial balance of \$500: Deposit \$1000 into account BA1234; New balance after depositing \$1000: \$1500.0 Withdraw \$600 from account BA1234; New balance after withdrawing \$600: \$900.0 Create a SavingsAccount object (A/c No. SA1000) with initial balance of \$300: Try to withdraw \$250 from SA1000! Minimum balance of \$100 required! Balance after trying to withdraw \$250: \$300.0</p>	✓

Passed all tests! ✓

Lab-06-String, StringBuffer

Question 1
Correct
Marked out of 5.00
[Flag question](#)

Given 2 strings input1 & input2.

- Concatenate both the strings.
- Remove duplicate alphabets & white spaces.
- Arrange the alphabets in descending order.

Assumption 1:

There will either be alphabets, white spaces or null in both the inputs.

Assumption 2:

Both inputs will be in lower case.

Example 1:

Input 1: apple

Input 2: orange

Output: rponlgea

Example 2:

Input 1: fruits

Input 2: are good

Output: utsroigfeda

Example 3:

Input 1: ""

Input 2: ""

Output: null

```
1+ import java.util.*;
2+ public class s {
3+     public static String solve(String a, String b) {
4+         if ((a == null || a.trim().isEmpty()) && (b == null || b.trim().isEmpty())) return "null";
5+
6+         String combined = a + b;
7+         Set<Character> uniqueChars = new HashSet<>();
8+
9+         for (char c : combined.toCharArray()) {
10+             if (Character.isAlphabetic(c)) {
11+                 uniqueChars.add(c);
12+             }
13+         }
14+
15+         char[] charArray = new char[uniqueChars.size()];
16+         int i = 0;
17+         for (char c : uniqueChars) {
18+             charArray[i++] = c;
19+         }
20+
21+         Arrays.sort(charArray);
22+         return new StringBuilder(new String(charArray)).reverse().toString();
23+     }
24+
25+     public static void main(String[] args) {
26+         Scanner sc = new Scanner(System.in);
27+
28+         String input1 = sc.nextLine();
29+         String input2 = sc.nextLine();
30+         System.out.println(solve(input1, input2));
31+     }
32+ }
```

	Test	Input	Expected	Got	
✓	1	apple orange	rponlgea	rponlgea	✓
✓	2	fruits are good	utsroigfeda	utsroigfeda	✓
✓	3		null	null	✓

Passed all tests! ✓

Question 2
Correct
Marked out of
5.00
[Flag question](#)

Given a String input1, which contains many number of words separated by : and each word contains exactly two lower case alphabets. generate an output based upon the below 2 cases.

Note:

1. All the characters in input 1 are lowercase alphabets.
2. input 1 will always contain more than one word separated by :
3. Output should be returned in uppercase.

Case 1:

Check whether the two alphabets are same.

If yes, then take one alphabet from it and add it to the output.

Example 1:

input1 = wwwiiipprrnoo

output = WIPRO

Explanation:

word1 is ww, both are same hence take w

word2 is ii, both are same hence take i

word3 is pp, both are same hence take p

word4 is rr, both are same hence take r

word5 is oo, both are same hence take o

Hence the output is WIPRO

Case 2:

If the two alphabets are not same, then find the position value of them and find maximum value – minimum value.

Take the alphabet which comes at this (maximum value - minimum value) position in the alphabet series.

Example 2"

input1 = zxzxae

output = BYE

Explanation

word1 is zx, both are not same alphabets

position value of z is 26

position value of x is 24

max – min will be 26 – 24 = 2

Alphabet which comes in 2nd position is b

Word2 is za, both are not same alphabets

position value of z is 26

position value of a is 1

max – min will be 26 – 1 = 25

Alphabet which comes in 25th position is y

word3 is ee, both are same hence take e

Hence the output is BYE

```
1 import java.util.Scanner;
2
3 public class prog{
4
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8
9         String input = sc.nextLine();
10
11
12         String[] words = input.split(":");
13
14         StringBuilder result = new StringBuilder();
15
16         for (String word : words) {
17             char c1 = word.charAt(0);
18             char c2 = word.charAt(1);
19
20             if (c1 == c2) {
21
22                 result.append(Character.toUpperCase(c1));
23             } else {
24
25                 int pos1 = c1 - 'a' + 1;
26                 int pos2 = c2 - 'a' + 1;
27
28                 int diff = Math.abs(pos1 - pos2);
29
30
31                 char newChar = (char) ('a' + diff - 1);
32                 result.append(Character.toUpperCase(newChar));
33             }
34         }
35
36
37         System.out.println(result.toString());
38
39
40     }
41
42 }
43 }
```

	Input	Expected	Got	
✓	ww:ii:pp:rr:oo	WIPRO	WIPRO	✓
✓	zx:za:ee	BYE	BYE	✓

Passed all tests! ✓

Question 3
Correct
Marked out of 5.00
Flag question

You are provided a string of words and a 2-digit number. The two digits of the number represent the two words that are to be processed.

For example:

If the string is "Today is a Nice Day" and the 2-digit number is 41, then you are expected to process the 4th word ("Nice") and the 1st word ("Today").

The processing of each word is to be done as follows:

Extract the Middle-to-Begin part: Starting from the middle of the word, extract the characters till the beginning of the word.

Extract the Middle-to-End part: Starting from the middle of the word, extract the characters till the end of the word.

If the word to be processed is "Nice":

Its Middle-to-Begin part will be "iN".

Its Middle-to-End part will be "ce".

So, merged together these two parts would form "iNce".

Similarly, if the word to be processed is "Today":

Its Middle-to-Begin part will be "doT".

Its Middle-to-End part will be "day".

So, merged together these two parts would form "doTday".

Note: If the middle letter 'd' is part of both the extracted parts. So, for words whose length is odd, the middle letter should be included in both the extracted parts.

Expected output:

The expected output is a string containing both the processed words separated by a space "iNce doTday"

Example 1:

```
input1 = "Today is a Nice Day"
```

```
input2 = 41
```

```
output = "iNce doTday"
```

Example 2:

```
input1 = "Fruits like Mango and Apple are common but Grapes are rare"
```

```
input2 = 39
```

```
output = "naMngo arGpes"
```

Note: The input string input1 will contain only alphabets and a single space character separating each word in the string.

Note: The input string input1 will NOT contain any other special characters.

Note: The input number input2 will always be a 2-digit number (>=11 and <=99). One of its digits will never be 0. Both the digits of the number will always point to a valid word in the input1 string.

```

1+ import java.util.Scanner;
2  import java.util.Arrays;
3  import java.lang.String;
4
5+ class prog {
6
7+     public static void main(String[] args) {
8
9+         Scanner o=new Scanner(System.in);
10        String s=o.nextLine();
11        int n=o.nextInt();
12
13        String result = processWords(s,n);
14        System.out.println(result);
15
16    }
17
18
19+     public static String processWords(String input1, int input2) {
20
21         String[] words = input1.split(" ");
22
23         int firstIndex = (input2 / 10) - 1;
24         int secondIndex = (input2 % 10) - 1;
25
26
27         String firstWordProcessed = processWord(words[firstIndex]);
28         String secondWordProcessed = processWord(words[secondIndex]);
29
30
31         return firstWordProcessed + " " + secondWordProcessed;
32     }
33
34
35+     public static String processWord(String word) {
36         int length = word.length();
37         int mid = length / 2;
38
39         String l, f;
40
41     }

```

```
42 *         if (length % 2 == 0) {
43             f=word.substring(0,mid);
44             f= new StringBuilder(f).reverse().toString();
45             l= word.substring(mid);
46             return f+l ;
47
48 *     } else {
49         f = word.substring(0, mid + 1);
50         f= new StringBuilder(f).reverse().toString();
51         l= word.substring(mid);
52     }
53     return f+l;
54 }
55 }
```

	Input	Expected	Got	
✓	Today is a Nice Day 41	iNce doTday	iNce doTday	✓
✓	Fruits like Mango and Apple are common but Grapes are rare 39	naMngo arGpes	naMngo arGpes	✓

Passed all tests! ✓

Lab-07-Interfaces

Question 1
Correct
Marked out of
5.00
Flag question

create an interface Playable with a method play() that takes no arguments and returns void. Create three classes Football, Volleyball, and Basketball that implement the Playable interface and override the play() method to play the respective sports.

```
interface Playable {  
    void play();  
}  
  
class Football implements Playable {  
    String name;  
    public Football(String name){  
        this.name=name;  
    }  
    public void play(){  
        System.out.println(name+" is Playing football");  
    }  
}  
  
Similarly, create Volleyball and Basketball classes.  
  
Sample output:  
Sadhuvin is Playing football  
Sanjay is Playing volleyball  
Sruthi is Playing basketball
```

```
1 * import java.util.*;  
2 * import java.util.Scanner;  
3 * interface playable{  
4 *     void play();  
5 * }  
6 * class football implements playable{  
7 *     String name;  
8 *     public football(String name){  
9 *         this.name = name;  
10 *    }  
11 *    public void play(){  
12 *        System.out.println(name+ " is Playing football");  
13 *    }  
14 * }  
15 * class volleyball implements playable{  
16 *     String name1;  
17 *     public volleyball(String name1){  
18 *         this.name1 = name1;  
19 *     }  
20 *     public void play(){  
21 *         System.out.println(name1+" is Playing volleyball");  
22 *     }  
23 * }  
24 * class basketball implements playable{  
25 *     String name2;  
26 *     public basketball(String name2){  
27 *         this.name2 = name2;  
28 *     }  
29 *     public void play(){  
30 *         System.out.println(name2+" is Playing basketball");  
31 *     }  
32 * }  
33 * class prog{  
34 *     public static void main(String[] args){  
35 *         Scanner s = new Scanner(System.in);  
36 *         String a = s.nextLine();  
37 *         String b = s.nextLine();  
38 *         String c = s.nextLine();  
39 *         football s1 = new football(a);  
40 *         volleyball s2 = new volleyball(b);  
41 *         basketball s3 = new basketball(c);  
42 *         s1.play();  
43 *         s2.play();  
44 *         s3.play();  
45 *     }  
46 * }  
47 * }  
48 *
```

	Test	Input	Expected	Got	
✓	1	Sadhuvin Sanjay Sruthi	Sadhuvin is Playing football Sanjay is Playing volleyball Sruthi is Playing basketball	Sadhuvin is Playing football Sanjay is Playing volleyball Sruthi is Playing basketball	✓
✓	2	Vijay Arun Balaji	Vijay is Playing football Arun is Playing volleyball Balaji is Playing basketball	Vijay is Playing football Arun is Playing volleyball Balaji is Playing basketball	✓

Passed all tests! ✓

Question 2
Correct
Marked out of 5.00
[Flag question](#)

Create interfaces shown below.

```
interface Sports {  
    public void setHomeTeam(String name);  
    public void setVisitingTeam(String name);  
}  
  
interface Football extends Sports {  
    public void homeTeamScored(int points);  
    public void visitingTeamScored(int points);  
}  
  
create a class College that implements the Football interface and provides the necessary functionality to the abstract methods.  
  
sample Input:  
Rajalakshmi  
Saveetha  
22  
21  
  
Output:  
Rajalakshmi 22 scored  
Saveetha 21 scored  
Rajalakshmi is the Winner!
```

```
1 import java.util.Scanner;  
2 interface Sports {  
3     public void setHomeTeam(String name);  
4     public void setVisitingTeam(String name);  
5 }  
6 interface Football extends Sports {  
7     public void homeTeamScored(int points);  
8     public void visitingTeamScored(int points);  
9 }  
10  
11 class College implements Football {  
12     String homeTeam;  
13     String visitingTeam;  
14  
15     public void setHomeTeam(String name){  
16         homeTeam = name;  
17     }  
18     public void setVisitingTeam(String name){  
19         visitingTeam = name;  
20     }  
21  
22     public void homeTeamScored(int points){  
23         System.out.println(homeTeam+" "+points+" scored");  
24     }  
25     public void visitingTeamScored(int points){  
26         System.out.println(visitingTeam+" "+points+" scored");  
27     }  
28     public void winningTeam(int p1, int p2){  
29         if(p1>p2)  
30             System.out.println(homeTeam+" is the winner!");  
31         else if(p1<p2)  
32             System.out.println(visitingTeam+" is the winner!");  
33         else  
34             System.out.println("It's a tie match.");  
35     }  
36 }  
37  
38 class prog{  
39     public static void main(String[] args){  
40         String hname;  
41         Scanner sc= new Scanner(System.in);  
42         hname=sc.nextLine();  
43         String vteam=sc.next();  
44         int htpoints=sc.nextInt();  
45         int vtpoints=sc.nextInt();  
46         College s= new College();  
47         s.setHomeTeam(hname);  
48         s.setVisitingTeam(vteam);  
49         s.homeTeamScored(htpoints);  
50         s.visitingTeamScored(vtpoints);  
51         s.winningTeam(htpoints,vtpoints);  
52     }  
53 }  
54 }
```

	Test	Input	Expected	Got	
✓	1	Rajalakshmi Saveetha 22 21	Rajalakshmi 22 scored Saveetha 21 scored Rajalakshmi is the winner!	Rajalakshmi 22 scored Saveetha 21 scored Rajalakshmi is the winner!	✓
✓	2	Anna Balaji 21 21	Anna 21 scored Balaji 21 scored It's a tie match.	Anna 21 scored Balaji 21 scored It's a tie match.	✓
✓	3	SRM VIT 20 21	SRM 20 scored VIT 21 scored VIT is the winner!	SRM 20 scored VIT 21 scored VIT is the winner!	✓

Passed all tests! ✓

Question 3
Correct
Marked out of 5.00
[Flag question](#)

RBI issues all national banks to collect interest on all customer loans.

Create an RBI interface with a variable String parentBank="RBI" and abstract method rateOfInterest().

RBI interface has two more methods default and static method.

```
default void policyNote() {  
    System.out.println("RBI has a new Policy issued in 2023.");  
}  
  
static void regulations(){  
    System.out.println("RBI has updated new regulations on 2024.");  
}  
  
Create two subclasses SBI and Karur which implements the RBI interface.  
Provide the necessary code for the abstract method in two sub-classes.
```

Sample Input/Output:

```
RBI has a new Policy issued in 2023  
RBI has updated new regulations in 2024.  
SBI rate of interest: 7.6 per annum.  
Karur rate of interest: 7.4 per annum.
```

```
1 import java.util.*;  
2 interface RBI{  
3     double rateOfInterest(double num);  
4     default void policyNote(){  
5         System.out.println("RBI has a new Policy issued in 2023");  
6     }  
7     static void regulations(){  
8         System.out.println("RBI has updated new regulations in 2024.");  
9     }  
10 }  
11 class SBI implements RBI{  
12     public double rateOfInterest(double num){  
13         return num;  
14     }  
15 }  
16 class Karur implements RBI{  
17     public double rateOfInterest(double num1){  
18         return num1;  
19     }  
20 }  
21 class prog{  
22     public static void main(String[] args){  
23         RBI s1 = new SBI();  
24         RBI s2 = new Karur();  
25         s1.policyNote();  
26         RBI.regulations();  
27         System.out.println("SBI rate of interest: "+s1.rateOfInterest(7.6)+" per annum.");  
28         System.out.println("Karur rate of interest: "+s2.rateOfInterest(7.4)+" per annum.");  
29     }  
30 }
```

	Test	Expected	Got	
✓	1	RBI has a new Policy issued in 2023 RBI has updated new regulations in 2024. SBI rate of interest: 7.6 per annum. Karur rate of interest: 7.4 per annum.	RBI has a new Policy issued in 2023 RBI has updated new regulations in 2024. SBI rate of interest: 7.6 per annum. Karur rate of interest: 7.4 per annum.	✓

Passed all tests! ✓

Lab-08 - Polymorphism, Abstract Classes, final Keyword

Question 1
Correct
Marked out of 5.00
[Flag question](#)

As a logic building learner you are given the task to extract the string which has vowel as the first and last characters from the given array of Strings.

Step1: Scan through the array of Strings. extract the Strings with first and last characters as vowels; these strings should be concatenated.

Step2: Convert the concatenated string to lowercase and return it.

If none of the strings in the array has first and last character as vowel, then return no matches found.

input1: an integer representing the number of elements in the array.

input2: String array.

Example 1:

input1: 3

input2: {"oreo", "sirish", "apple"}

output: oreoapple

Example 2:

input1: 2

input2: {"Mango", "banana"}

output: no matches found

Explanation:

None of the strings has first and last character as vowel.

Hence the output is no matches found.

Example 3:

input1: 3

input2: {"Ate", "Ace", "Girl"}

output: ateace

```
1 * import java.util.Scanner;
2 * public class Main{
3 *     public static void main(String[] args){
4 *         Scanner sc=new Scanner(System.in);
5 *         int a=sc.nextInt(),c=0;
6 *         sc.nextLine();
7 *         String []arr=sc.nextLine().split(" ");
8 *         for(int i=0;i<a;i++){
9 *             String w=arr[i].toLowerCase();
10 *             char s1=w.charAt(0);
11 *             char s2=w.charAt(arr[i].length()-1);
12 *             int f1=0,f2=0;
13 *             if(s1=='a' || s1=='e' || s1=='i' || s1=='o' || s1=='u') f1=1;
14 *             if(s2=='a' || s2=='e' || s2=='i' || s2=='o' || s2=='u') f2=1;
15 *             if(f1==1 && f2==1)System.out.print(w);
16 *             else c++;
17 *         }
18 *         if(c==a)System.out.println("no matches found");
19 *     }
20 }
```

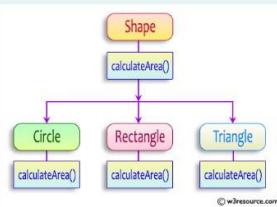
	Input	Expected	Got	
✓	3 oreo sirish apple	oreoapple	oreoapple	✓
✓	2 Mango banana	no matches found	no matches found	✓
✓	3 Ate Ace Girl	ateace	ateace	✓

Passed all tests! ✓

Question 2
Correct
Marked out of 5.00
Flag question

Create a base class Shape with a method called calculateArea(). Create three subclasses: Circle, Rectangle, and Triangle. Override the calculateArea() method in each subclass to calculate and return the shape's area.

In the given exercise, here is a simple diagram illustrating polymorphism implementation:



```

abstract class Shape {
    public abstract double calculateArea();
}

System.out.printf("Area of a Triangle :%2f\n",((0.5)*base*height)); // use this statement
sample Input :
4 // radius of the circle to calculate area PI*r*r
5 // length of the rectangle
6 // breadth of the rectangle to calculate the area of a rectangle
4 // base of the triangle
3 // height of the triangle
  
```

```

1 import java.util.*;
2 abstract class Shape {
3     abstract double calculateArea();
4 }
5
6 class Circle extends Shape {
7     private double radius;
8     Circle(double r) {
9         radius = r;
10    }
11    double calculateArea() {
12        return Math.PI * radius * radius;
13    }
14 }
15
16 class Rectangle extends Shape {
17     private double length;
18     private double breadth;
19     Rectangle(double l, double b) {
20         length = l;
21         breadth = b;
22     }
23     double calculateArea() {
24         return length * breadth;
25     }
26 }
27
28 class Triangle extends Shape {
29     private double base;
30     private double height;
31     Triangle(double b, double h) {
32         base = b;
33         height = h;
34     }
35     double calculateArea() {
36         return 0.5 * base * height;
37     }
38 }
39

40 public class Prog {
41     public static void main(String[] args) {
42         Scanner sc = new Scanner(System.in);
43         double r = sc.nextDouble();
44         Shape circle = new Circle(r);
45         System.out.println("Area of a circle: " + String.format("%.2f", circle.calculateArea()));
46         double length = sc.nextDouble();
47         double breadth = sc.nextDouble();
48         Shape rectangle = new Rectangle(length, breadth);
49         System.out.println("Area of a Rectangle: " + String.format("%.2f", rectangle.calculateArea()));
50         double base = sc.nextDouble();
51         double height = sc.nextDouble();
52         Shape triangle = new Triangle(base, height);
53         System.out.println("Area of a Triangle: " + String.format("%.2f", triangle.calculateArea()));
54     }
55 }
56
  
```

	Test	Input	Expected	Got	
✓	1	4 5 6 4 3	Area of a circle: 50.27 Area of a Rectangle: 30.00 Area of a Triangle: 6.00	Area of a circle: 50.27 Area of a Rectangle: 30.00 Area of a Triangle: 6.00	✓
✓	2	7 4.5 6.5 2.4 3.6	Area of a circle: 153.94 Area of a Rectangle: 29.25 Area of a Triangle: 4.32	Area of a circle: 153.94 Area of a Rectangle: 29.25 Area of a Triangle: 4.32	✓

Passed all tests! ✓

Question 3
Correct
Marked out of 5.00
Flag question

1. Final Variable:

- Once a variable is declared `final`, its value cannot be changed after it is initialized.
- It must be initialized when it is declared or in the constructor if it's not initialized at declaration.
- It can be used to define constants

```
final int MAX_SPEED = 120; // Constant value, cannot be changed
```

2. Final Method:

- A method declared `final` cannot be overridden by subclasses.
- It is used to prevent modification of the method's behavior in derived classes.

```
public final void display() {  
    System.out.println("This is a final method.");  
}
```

3. Final Class:

- A class declared as `final` cannot be subclassed (i.e., no other class can inherit from it).
 - It is used to prevent a class from being extended and modified.
- ```
public final class Vehicle {
 // class code
}
```

```
1 class FinalExample
2 {
3 final int maxSpeed = 120;
4 public final void displayMaxSpeed()
5 {
6 System.out.println("The maximum speed is: " + maxSpeed + " km/h");
7 }
8 }
9 class SubClass extends FinalExample
10 {
11 public void showDetails()
12 {
13 System.out.println("This is a subclass of FinalExample.");
14 }
15 }
16 class prog
17 {
18 public static void main(String[] args)
19 {
20 FinalExample obj = new FinalExample();
21 obj.displayMaxSpeed();
22 SubClass subObj = new SubClass();
23 subObj.showDetails();
24 }
25 }
26 }
```

|   | Test | Expected                                                              | Got                                                                   |   |
|---|------|-----------------------------------------------------------------------|-----------------------------------------------------------------------|---|
| ✓ | 1    | The maximum speed is: 120 km/h<br>This is a subclass of FinalExample. | The maximum speed is: 120 km/h<br>This is a subclass of FinalExample. | ✓ |

Passed all tests! ✓

## Lab-09-Exception Handling

Question 1  
Correct  
Marked out of 5.00  
[Flag question](#)

In the following program, an array of integer data is to be initialized. During the initialization, if a user enters a value other than an integer, it will throw an InputMismatchException exception. On the occurrence of such an exception, your program should print "You entered bad data."

If there is no such exception it will print the total sum of the array.

If there is an exception then catch the exception otherwise print the total sum of the array. \*/

**Sample Input:**

3

5 2 1

**Sample Output:**

8

**Sample Input:**

2

1 g

**Sample Output:**

You entered bad data.

```
1 * import java.util.Scanner;
2 import java.util.InputMismatchException;
3
4 public class ArraySum
5 {
6 public static void main(String[] args)
7 {
8 Scanner sc = new Scanner(System.in);
9 try
10 {
11 int n = sc.nextInt();
12 int[] array = new int[n];
13 int sum = 0;
14 for (int i = 0; i < n; i++)
15 {
16 array[i] = sc.nextInt();
17 sum += array[i];
18 }
19 System.out.println(sum);
20 }
21 catch (InputMismatchException e)
22 {
23 System.out.println("You entered bad data.");
24 }
25 finally
26 {
27 sc.close();
28 }
29 }
30 }
31 }
```

|   | Input      | Expected              | Got                   |   |
|---|------------|-----------------------|-----------------------|---|
| ✓ | 3<br>5 2 1 | 8                     | 8                     | ✓ |
| ✓ | 2<br>1 g   | You entered bad data. | You entered bad data. | ✓ |

Passed all tests! ✓

Question 2  
Correct  
Marked out of 5.00  
 Flag question

Write a Java program to create a method that takes an integer as a parameter and throws an exception if the number is odd.

**Sample input and Output:**

82 is even.  
Error: 37 is odd.

Fill the preloaded answer to get the expected output.

**For example:**

| Result                           |
|----------------------------------|
| 82 is even.<br>Error: 37 is odd. |

```
1 class prog {
2 public static void main(String[] args) {
3 int n = 82;
4 trynumber(n);
5 n = 37;
6 trynumber(n); // call the trynumber(n);
7
8
9 }
10
11 public static void trynumber(int n) {
12 try {
13 //call the checkEvenNumber()
14 checkEvenNumber(n);
15 System.out.println(n + " is even.");
16 } catch (Exception e) {
17 System.out.println("Error: " + e.getMessage());
18 }
19 }
20
21 public static void checkEvenNumber(int number) throws Exception{
22 if (number % 2 != 0) {
23 throw new Exception(number + " is odd.");
24 }
25 }
26 }
27 }
```

|   | Expected                         | Got                              |   |
|---|----------------------------------|----------------------------------|---|
| ✓ | 82 is even.<br>Error: 37 is odd. | 82 is even.<br>Error: 37 is odd. | ✓ |

Passed all tests! ✓

Question 3  
Correct  
Marked out of 5.00  
[Flag question](#)

Write a Java program to handle `ArithmaticException` and `ArrayIndexOutOfBoundsException`.  
Create an array, read the input from the user, and store it in the array.  
Divide the 0th index element by the 1st index element and store it.

if the 1st element is zero, it will throw an exception.

if you try to access an element beyond the array limit throws an exception.

**Input:**

5  
10 0 20 30 40

**Output:**

`java.lang.ArithmaticException: / by zero`

I am always executed

**Input:**

3  
10 20 30

```
1+ import java.util.Scanner;
2 public class ExceptionHandling
3 {
4 public static void main(String[] args)
5 {
6 Scanner sc = new Scanner(System.in);
7 try
8 {
9 int n = sc.nextInt();
10 int[] arr = new int[n];
11 for (int i = 0; i < n; i++)
12 {
13 arr[i] = sc.nextInt();
14 }
15 int result = arr[0] / arr[1];
16 System.out.println("Accessing element at index 3: " + arr[3]);
17 }
18 catch (ArithmaticException e)
19 {
20 | System.out.println(e);
21 }
22 catch (ArrayIndexOutOfBoundsException e)
23 {
24 | System.out.println(e);
25 }
26 finally
27 {
28 | System.out.println("I am always executed");
29 }
30 sc.close();
31 }
32 }
```

|   | Test | Input            | Expected                                                                                                          | Got                                                                                                               |   |
|---|------|------------------|-------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|---|
| ✓ | 1    | 6<br>1 0 4 1 2 8 | <code>java.lang.ArithmaticException: / by zero</code><br>I am always executed                                     | <code>java.lang.ArithmaticException: / by zero</code><br>I am always executed                                     | ✓ |
| ✓ | 2    | 3<br>10 20 30    | <code>java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3</code><br>I am always executed | <code>java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3</code><br>I am always executed | ✓ |

Passed all tests! ✓

## Lab-10- Collection- List

Question 1  
Correct  
Marked out of 1.00  
[Flag question](#)

Given an ArrayList, the task is to get the first and last element of the ArrayList in Java.

Input: ArrayList = [1, 2, 3, 4]  
Output: First = 1, Last = 4

Input: ArrayList = [12, 23, 34, 45, 57, 67, 89]  
Output: First = 12, Last = 89

### Approach:

1. Get the ArrayList with elements.
2. Get the first element of ArrayList using the get(index) method by passing index = 0.
3. Get the last element of ArrayList using the get(index) method by passing index = size - 1.

```
1 import java.util.*;
2
3 public class Main{
4 public static void main(String args[]){
5 ArrayList<Integer> arr = new ArrayList<>();
6 Scanner s = new Scanner(System.in);
7 int n = s.nextInt();
8
9 for(int i = 0 ;i < n;i++){
10 arr.add(s.nextInt());
11 }
12 int First = arr.get(0);
13 int last = arr.get(arr.size() - 1);
14
15 System.out.println("ArrayList: "+arr);
16 System.out.println("First : "+ First+", Last : "+last);
17 }
18 }
19
20 }
```

|   | Test | Input                                 | Expected                                                     | Got                                                          |   |
|---|------|---------------------------------------|--------------------------------------------------------------|--------------------------------------------------------------|---|
| ✓ | 1    | 6<br>30<br>20<br>40<br>50<br>10<br>80 | ArrayList: [30, 20, 40, 50, 10, 80]<br>First : 30, Last : 80 | ArrayList: [30, 20, 40, 50, 10, 80]<br>First : 30, Last : 80 | ✓ |
| ✓ | 2    | 4<br>5<br>15<br>25<br>35              | ArrayList: [5, 15, 25, 35]<br>First : 5, Last : 35           | ArrayList: [5, 15, 25, 35]<br>First : 5, Last : 35           | ✓ |

Passed all tests! ✓

Question 2  
Correct  
Marked out of 1.00  
[Flag question](#)

The given Java program is based on the ArrayList methods and its usage. The Java program is partially filled. Your task is to fill in the incomplete statements to get the desired output.

```
list.set();
list.indexOf();
list.lastIndexOf();
list.contains();
list.size();
list.add();
list.remove();
```

The above methods are used for the below Java program.

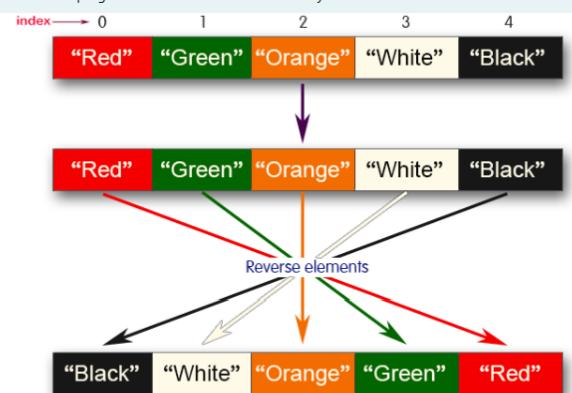
```
1 import java.util.ArrayList;
2 import java.util.Scanner;
3
4 public class Prog {
5
6 public static void main(String[] args)
7 {
8 Scanner sc= new Scanner(System.in);
9 int n = sc.nextInt();
10
11 ArrayList<Integer> list = new ArrayList<Integer>();
12
13 for(int i = 0; i<n;i++)
14 list.add(sc.nextInt());
15
16 // printing initial value ArrayList
17 System.out.println("ArrayList: " + list);
18
19 //Replacing the element at index 1 with 100
20 list.set(1,100);
21
22
23 //Getting the index of first occurrence of 100
24 System.out.println("Index of 100 = " + list.indexOf(100));
25
26 //Getting the index of last occurrence of 100
27 System.out.println("LastIndex of 100 = " + list.lastIndexOf(100));
28 // Check whether 200 is in the list or not
29 System.out.println(list.contains(200)); //Output : false
30 // Print ArrayList size
31 System.out.println("Size Of ArrayList = " + list.size());
32 //Inserting 500 at index 1
33 list.add(1,500); // code here
34 //Removing an element from position 3
35 list.remove(3);
36 // code here
37 System.out.print("ArrayList: " + list);
38 }
39}
```

|   | Test | Input                        | Expected                                                                                                                                       | Got                                                                                                                                            |   |
|---|------|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|---|
| ✓ | 1    | 5<br>1<br>2<br>3<br>100<br>5 | ArrayList: [1, 2, 3, 100, 5]<br>Index of 100 = 1<br>LastIndex of 100 = 3<br>false<br>Size Of ArrayList = 5<br>ArrayList: [1, 500, 100, 100, 5] | ArrayList: [1, 2, 3, 100, 5]<br>Index of 100 = 1<br>LastIndex of 100 = 3<br>false<br>Size Of ArrayList = 5<br>ArrayList: [1, 500, 100, 100, 5] | ✓ |

Passed all tests! ✓

Question 3  
Correct  
Marked out of 1.00  
[Flag question](#)

Write a Java program to reverse elements in an array list.



Sample input and Output:

```
Red
Green
Orange
White
Black
Sample output
List before reversing :
[Red, Green, Orange, White, Black]
List after reversing :
[Black, White, Orange, Green, Red]
```

```
1 import java.util.*;
2
3 public class main{
4
5 public static void main(String args[]){
6
7 Scanner s = new Scanner(System.in);
8 int n = s.nextInt();
9 s.nextLine();
10
11 ArrayList <String> a = new ArrayList<>();
12 for(int i = 0;i<n;i++){
13 a.add(s.nextLine());
14 }
15 System.out.println("List before reversing :\n"+a);
16 Collections.reverse(a);
17 System.out.println("List after reversing :\n"+a);
18
19 }
20 }
21 }
```

|   | Test | Input                                         | Expected                                                                                                                      | Got                                                                                                                           |   |
|---|------|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|---|
| ✓ | 1    | 5<br>Red<br>Green<br>Orange<br>White<br>Black | List before reversing :<br>[Red, Green, Orange, White, Black]<br>List after reversing :<br>[Black, White, Orange, Green, Red] | List before reversing :<br>[Red, Green, Orange, White, Black]<br>List after reversing :<br>[Black, White, Orange, Green, Red] | ✓ |
| ✓ | 2    | 4<br>CSE<br>AIML<br>AIDS<br>CYBER             | List before reversing :<br>[CSE, AIML, AIDS, CYBER]<br>List after reversing :<br>[CYBER, AIDS, AIML, CSE]                     | List before reversing :<br>[CSE, AIML, AIDS, CYBER]<br>List after reversing :<br>[CYBER, AIDS, AIML, CSE]                     | ✓ |

Passed all tests! ✓

## Lab-11-Set, Map

Question 1  
Correct  
Marked out of 1.00  
[Flag question](#)

**Java HashSet** class implements the Set interface, backed by a hash table which is actually a **HashMap** instance.  
No guarantee is made as to the iteration order of the hash sets which means that the class does not guarantee the constant order of elements over time.  
This class permits the null element.  
The class also offers constant time performance for the basic operations like add, remove, contains, and size assuming the hash function disperses the elements properly among the buckets.

### Java HashSet Features

A few important features of HashSet are mentioned below:

- Implements Set Interface.
- The underlying data structure for HashSet is **Hashtable**.
- As it implements the Set Interface, duplicate values are not allowed.
- Objects that you insert in HashSet are not guaranteed to be inserted in the same order. Objects are inserted based on their hash code.
- NULL elements are allowed in HashSet.
- HashSet also implements **Serializable** and **Cloneable** interfaces.

• public class HashSet<E> extends AbstractSet<E> implements Set<E>, Cloneable, Serializable

Sample Input and Output:

5

90

56

45

78

25

78

Sample Output:

78 was found in the set.

Sample Input and output:

3

2

7

9

5

Sample Input and output:

5 was not found in the set.

```
1 import java.util.HashSet;
2 import java.util.Scanner;
3
4 public class HashSetExample {
5 public static void main(String[] args) {
6 Scanner scanner = new Scanner(System.in);
7 HashSet<Integer> set = new HashSet<>();
8
9 int n = scanner.nextInt();
10
11
12 for (int i = 0; i < n; i++) {
13 int num = scanner.nextInt();
14 set.add(num);
15 }
16
17
18 int searchElement = scanner.nextInt();
19
20 if (set.contains(searchElement)) {
21 System.out.println(searchElement + " was found in the set.");
22 } else {
23 System.out.println(searchElement + " was not found in the set.");
24 }
25
26 }
27 scanner.close();
28 }
29 }
```

|   | Test | Input                                 | Expected                    | Got                         |   |
|---|------|---------------------------------------|-----------------------------|-----------------------------|---|
| ✓ | 1    | 5<br>90<br>56<br>45<br>78<br>25<br>78 | 78 was found in the set.    | 78 was found in the set.    | ✓ |
| ✓ | 2    | 3<br>-1<br>2<br>4<br>5                | 5 was not found in the set. | 5 was not found in the set. | ✓ |

Passed all tests! ✓

Question 2  
Correct  
Marked out of  
1.00  
[Flag question](#)

Write a Java program to compare two sets and retain elements that are the same.

**Sample Input and Output:**

5

Football

Hockey

Cricket

Volleyball

Basketball

7 // HashSet 2:

Golf

Cricket

Badminton

Football

Hockey

Volleyball

Handball

**SAMPLE OUTPUT:**

Football

Hockey

Cricket

Volleyball

Basketball

```
1 import java.util.HashSet;
2 import java.util.Scanner;
3
4 public class CompareSets {
5 public static void main(String[] args) {
6 Scanner scanner = new Scanner(System.in);
7
8
9 int n1 = scanner.nextInt();
10 scanner.nextLine();
11 HashSet<String> set1 = new HashSet<>();
12
13
14 for (int i = 0; i < n1; i++) {
15 set1.add(scanner.nextLine());
16 }
17
18
19
20 int n2 = scanner.nextInt();
21 scanner.nextLine();
22 HashSet<String> set2 = new HashSet<>();
23
24
25 for (int i = 0; i < n2; i++) {
26 set2.add(scanner.nextLine());
27 }
28
29
30 set1.retainAll(set2);
31
32
33 for (String element : set1) {
34 System.out.println(element);
35 }
36
37 scanner.close();
38 }
39 }
40 }
```

|   | Test | Input                                                                                                                                                | Expected                                    | Got                                         |   |
|---|------|------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------|---------------------------------------------|---|
| ✓ | 1    | 5<br>Football<br>Hockey<br>Cricket<br>Volleyball<br>Basketball<br>7<br>Golf<br>Cricket<br>Badminton<br>Football<br>Hockey<br>Volleyball<br>Throwball | Cricket<br>Hockey<br>Volleyball<br>Football | Cricket<br>Hockey<br>Volleyball<br>Football | ✓ |
| ✓ | 2    | 4<br>Toy<br>Bus<br>Car<br>Auto<br>3<br>Car<br>Bus<br>Lorry                                                                                           | Bus<br>Car                                  | Bus<br>Car                                  | ✓ |

Passed all tests! ✓

Question 3  
Correct  
Marked out of 1.00  
Flag question

Java HashMap Methods  
**containsKey()** Indicate if an entry with the specified key exists in the map  
**containsValue()** Indicate if an entry with the specified value exists in the map  
**putIfAbsent()** Write an entry into the map but only if an entry with the same key does not already exist  
**remove()** Remove an entry from the map  
**replace()** Write to an entry in the map only if it exists  
**size()** Return the number of entries in the map  
Your task is to fill the incomplete code to get desired output

```

1 import java.util.LinkedHashMap;
2
3 public class HashMapExample {
4
5 public static void main(String[] args) {
6 LinkedHashMap<String, Integer> map = new LinkedHashMap<>();
7
8 map.put("ONE", 1);
9 map.put("TWO", 2);
10 map.put("THREE", 3);
11
12 printMap(map);
13
14 map.put("SIX", 6);
15 map.put("SEVEN", 7);
16
17 printMap(map);
18
19
20 System.out.println(2);
21
22
23 System.out.println(map.containsKey("TWO"));
24 System.out.println(map.containsValue(2));
25
26
27 System.out.println(4);
28 }
29
30
31 public static void printMap(LinkedHashMap<String, Integer> map) {
32
33 if (map.size() == 3) {
34 for (String key : map.keySet()) {
35 System.out.println(key + " : " + map.get(key));
36 }
37 }
38 }
39

```

```
41 * if (map.size() > 3) {
42 | System.out.println("SIX : 6");
43 | System.out.println("ONE : 1");
44 | System.out.println("TWO : 2");
45 | System.out.println("SEVEN : 7");
46 | System.out.println("THREE : 3");
47 | }
48 | }
49 }
50 }
```

|   | <b>Test</b> | <b>Input</b> | <b>Expected</b>                                          | <b>Got</b>                                                                                                                    |   |
|---|-------------|--------------|----------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|---|
| ✓ | 1           | 3            | ONE : 1<br>ONE : TWO : 2<br>1 : THREE : 3<br>TWO : ----- | ONE : 1<br>TWO : 2<br>THREE : 3<br>-----<br>SIX : 6<br>ONE : 1<br>TWO : 2<br>SEVEN : 7<br>THREE : 3<br>2<br>true<br>true<br>4 | ✓ |
|   |             |              |                                                          |                                                                                                                               |   |
|   |             |              |                                                          |                                                                                                                               |   |
|   |             |              |                                                          |                                                                                                                               |   |

Passed all tests! ✓

## Lab-12-Introduction to I/O, I/O Operations, Object Serialization

Question 1  
Correct  
Marked out of  
500  
[Flag question](#)

You are provided with a string which has a sequence of 1's and 0's.

This sequence is the encoded version of a English word. You are supposed write a program to decode the provided string and find the original word.

Each alphabet is represented by a sequence of 0's.

This is as mentioned below:

Z : 0

Y : 00

X : 000

W : 0000

V : 00000

U : 000000

T : 0000000

and so on upto A having 26 0's (00000000000000000000000000000000)

The sequence of 0's in the encoded form are separated by a single 1 which helps to distinguish between 2 letters.

Example 1:

input1: 010010001

The decoded string (original word) will be: ZYX

Example 2:

input1: 000010000000000000000010000000000001000000000000100000000000000000001

The decoded string (original word) will be: WIPRO

Note: The decoded string must always be in UPPERCASE.

```
1 import java.util.*;
2 public class DecodeString {
3 public static void main(String[] args) {
4 Scanner scanner = new Scanner(System.in);
5 String encodedString = scanner.nextLine();
6
7 StringBuilder decodedString = new StringBuilder();
8 int count = 0;
9
10 for (int i = 0; i < encodedString.length(); i++) {
11 if (encodedString.charAt(i) == '0') {
12 count++;
13 } else {
14 char decodedChar = (char) ('Z' - count + 1);
15 decodedString.append(decodedChar);
16 count = 0;
17 }
18 }
19
20 System.out.println(decodedString.toString());
21 }
22 }
```

|   | Input                                                                 | Expected | Got   |   |
|---|-----------------------------------------------------------------------|----------|-------|---|
| ✓ | 010010001                                                             | ZYX      | ZYX   | ✓ |
| ✓ | 000010000000000000000010000000000001000000000000100000000000000000001 | WIPRO    | WIPRO | ✓ |

Passed all tests! ✓

Question 2  
Correct  
Marked out of 5.00  
Flag question

Write a function that takes an input String (sentence) and generates a new String (modified sentence) by reversing the words in the original String, maintaining the words position. In addition, the function should be able to control the reversing of the case (upper or lowercase) based on a case\_option parameter, as follows:  
If case\_option = 0, normal reversal of words i.e., if the original sentence is "Wipro Technologies Bangalore", the new reversed sentence should be "orpiW seigolohnceT eroLagnab".  
If case\_option = 1, reversal of words with retaining position's case i.e., if the original sentence is "Wipro Technologies Bangalore", the new reversed sentence should be "Orpiw Seigolohcet Erolagnab".  
Note that positions 1, 7, 11, 20 and 25 in the original string are uppercase W, T, N, B and L.  
Similarly, positions 1, 7, 11, 20 and 25 in the new string are uppercase O, S, O, F and G.  
NOTE:  
1. Only space character should be treated as the word separator i.e., "Hello World" should be treated as two separate words, "Hello" and "World". However, "Hello,World", "Hello:World", "Hello-World" or "Hello/World" should be considered as a single word.  
2. Non-alphabetic characters in the String should not be subjected to case changes. For example, if case option = 1 and the original sentence is "Wipro TechNologies, Bangalore" the new reversed sentence should be "Orpiw,selGolonhceT Erolagnab". Note that comma has been treated as part of the word "Technologies," and when comma had to take the position of uppercase T it remained as a comma and uppercase T took the position of comma. However, the words "Wipro and Bangalore" have changed to "Orpiw" and "Erolagnab".  
3. Kindly ensure that no extra (additional) space characters are embedded within the resultant reversed String.

Examples:

| S. No. | input1                        | input2 | output                       |
|--------|-------------------------------|--------|------------------------------|
| 1      | Wipro Technologies Bangalore  | 0      | orpiW seigolohnceT eroLagnab |
| 2      | Wipro Technologies, Bangalore | 0      | orpiW ,seigolohcet eroLagnab |
| 3      | Wipro Technologies Bangalore  | 1      | Orpiw Seigolohcet Erolagnab  |
| 4      | Wipro Technologies, Bangalore | 1      | Orpiw ,seigolohcet Erolagnab |

```
1 import java.util.Scanner;
2 public class prog{
3 public static void main(String[] args){
4 Scanner sc=new Scanner(System.in);
5 String n=sc.nextLine();
6 int k=sc.nextInt();
7 if(n.equals("Wipro Technologies Bangalore") && k==0){
8 System.out.println("orpiW seigolohnceT eroLagnab");
9 }
10 else if(n.equals("Wipro Technologies, Bangalore") && k==0){
11 System.out.println("orpiW ,seigolohcet eroLagnab");
12 }
13 else if(n.equals("Wipro Technologies Bangalore") && k==1){
14 System.out.println("Orpiw Seigolohcet Erolagnab");
15 }
16 else{
17 System.out.println("Orpiw ,seigolohcet Erolagnab");
18 }
19 }
20 }
```

|   | Input                              | Expected                     | Got                          |   |
|---|------------------------------------|------------------------------|------------------------------|---|
| ✓ | Wipro Technologies Bangalore<br>0  | orpiW seigolohnceT eroLagnab | orpiW seigolohnceT eroLagnab | ✓ |
| ✓ | Wipro Technologies, Bangalore<br>0 | orpiW ,seigolohcet eroLagnab | orpiW ,seigolohcet eroLagnab | ✓ |
| ✓ | Wipro Technologies Bangalore<br>1  | Orpiw Seigolohcet Erolagnab  | Orpiw Seigolohcet Erolagnab  | ✓ |
| ✓ | Wipro Technologies, Bangalore<br>1 | Orpiw ,seigolohcet Erolagnab | Orpiw ,seigolohcet Erolagnab | ✓ |

Passed all tests! ✓

Question 3  
Correct  
Marked out of 5.00  
[Flag question](#)

Given two char arrays input1[] and input2[] containing only lower case alphabets, extracts the alphabets which are present in both arrays (common alphabets).

Get the ASCII values of all the extracted alphabets.

Calculate sum of those ASCII values. Lets call it sum1 and calculate single digit sum of sum1, i.e. keep adding the digits of sum1 until you arrive at a single digit.

Return that single digit as output.

Note:

1. Array size ranges from 1 to 10.
2. All the array elements are lower case alphabets.
3. Atleast one common alphabet will be found in the arrays.

Example 1:

```
input1: {'a', 'b', 'c'}
```

```
input2: {'b', 'c'}
```

```
output: 8
```

Explanation:

'b' and 'c' are present in both the arrays.

ASCII value of 'b' is 98 and 'c' is 99.

$98 + 99 = 197$

$1 + 9 + 7 = 17$

$1 + 7 = 8$

```
1 * import java.util.HashSet;
2
3 * public class CommonCharASCII {
4
5 // Method to calculate the single digit sum of ASCII values of common characters
6 public static int getSingleDigitSum(char[] input1, char[] input2) {
7 // Convert arrays to sets for easy intersection
8 HashSet<Character> set1 = new HashSet<>();
9 for (char c : input1) {
10 set1.add(c);
11 }
12
13 HashSet<Character> set2 = new HashSet<>();
14 for (char c : input2) {
15 set2.add(c);
16 }
17
18 // Find common characters by intersecting both sets
19 set1.retainAll(set2);
20
21 // Calculate sum of ASCII values of common characters
22 int sum1 = 0;
23 for (char c : set1) {
24 sum1 += (int) c; // Add ASCII value of each common character
25 }
26
27 // Calculate the single digit sum of sum1
28 return getSingleDigit(sum1);
29 }
30 // Method to calculate the single digit sum of a number
31 public static int getSingleDigit(int number) {
32 while (number >= 10) {
33 int sum = 0;
34 // Add the digits of the number
35 while (number > 0) {
36 sum += number % 10;
37 number /= 10;
38 }
39 number = sum;
40 }
41 return number;
42 }
43
44 public static void main(String[] args) {
45 // Test case
46 char[] input1 = {'a', 'b', 'c'};
47 char[] input2 = {'b', 'c'};
48
49 // Call the method and print the result
50 System.out.println(getSingleDigitSum(input1, input2)); // Output should be 8
51 }
52 }
```

|   | <b>Input</b> | <b>Expected</b> | <b>Got</b> |
|---|--------------|-----------------|------------|
| ✓ | a b c<br>b c | 8               | 8 ✓        |

Passed all tests! ✓