

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-01-Java Architecture, Language Basics](#) / [Lab-01-Logic Building](#)

Status Finished

Started Saturday, 21 September 2024, 8:59 PM

Completed Saturday, 21 September 2024, 9:24 PM

Duration 24 mins 57 secs

Question 1

Correct

Marked out of 5.00

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.Scanner;
2 public class OddEvenCheck {
3     public static void main(String[] args) {
4         int number = new Scanner(System.in).nextInt();
5         System.out.println(number != 0 && number % 2 != 0? 2 : 1);
6     }
7 }
8 }
```

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

Passed all tests! ✓

Question 2

Correct

Marked out of 5.00

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.Scanner;
2 public class LastDigit {
3     public static void main(String[] args){
4         int number = new Scanner(System.in).nextInt();
5         System.out.println(Math.abs(number % 10));
6     }
7 }
```

	Input	Expected	Got	
✓	197	7	7	✓
✓	-197	7	7	✓

Passed all tests! ✓

Question 3

Correct

Marked out of 5.00

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: Tle 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
267 154	11
267 -154	11
-267 154	11
-267 -154	11

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 public class LastDigitSum {
3     public static void main(String[] args){
4         Scanner scanner = new Scanner(System.in);
5         int num1= scanner.nextInt();
6         int num2= scanner.nextInt();
7         int sum = Math.abs(num1 %10) + Math.abs(num2 %10);
8         System.out.println(sum);
9     }
10 }
```

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

Passed all tests! ✓



◀ Lab-01-MCQ

Jump to...

Is Even? ►

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-02-Flow Control Statements](#) / [Lab-02-Logic Building](#)

Status Finished

Started Sunday, 22 September 2024, 9:03 PM

Completed Sunday, 22 September 2024, 9:24 PM

Duration 21 mins 42 secs

Question 1

Correct

Marked out of 5.00

Write a program that takes as parameter an integer n.

You have to print the number of zeros at the end of the factorial of n.

For example, $3! = 6$. The number of zeros are 0. $5! = 120$. The number of zeros at the end are 1.

Note: $n! < 10^5$

Example Input:

3

Output:

0

Example Input:

60

Output:

14

Example Input:

100

Output:

24

Example Input:

1024

Output:

253

For example:

Input	Result
3	0
60	14
100	24
1024	253

Answer: (penalty regime: 0 %)

Reset answer

```

1 import java.util.Scanner;
2
3 public class FactorialTrailingZeros {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         int n = sc.nextInt();
7         System.out.println(countTrailingZeros(n));
8     }
9
10    public static int countTrailingZeros(int n) {
11        int count = 0;
12        while (n >= 5) {
13            n /= 5;
14            count += n;
15        }
16        return count;

```

```
15 }  
16 }  
17 }  
18 }  
19 }
```

	Input	Expected	Got	
✓	3	0	0	✓
✓	60	14	14	✓
✓	100	24	24	✓
✓	1024	253	253	✓

Passed all tests! ✓



Question 2

Correct

Marked out of 5.00

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

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2
3 public class NumberToWords {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         String[] words = {"Zero", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine"};
7         String number = sc.nextLine();
8         for (char digit : number.toCharArray()) {
9             System.out.print(words[Character.getNumericValue(digit)] + " ");
10        }
11    }
12 }
```

	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! ✓

Question 3

Correct

Marked out of 5.00

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

For example:

Input	Result
1 false	6:00
5 false	5:00
1 true	9:00

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2
3 public class WakeUpTime {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         int day = scanner.nextInt();
7         boolean vacation = scanner.nextBoolean();
8
9         if (vacation) {
10            if (day == 1 || day == 7) {
11                System.out.println("9:00");
12            } else {
13                System.out.println("7:00");
14            }
15        } else {
16            if (day == 1 || day == 7) {

```

```
17     System.out.println("6:00");
18 } else {
19     System.out.println("5:00");
20 }
21 }
22 }
23 }
```

	Input	Expected	Got	
✓	1 false	6:00	6:00	✓
✓	5 false	5:00	5:00	✓
✓	1 true	9:00	9:00	✓

Passed all tests! ✓

◀ Lab-02-MCQ

Jump to...

Lab-03-MCQ ►

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-03-Arrays](#) / [Lab-03-Logic Building](#)

Status	Finished
Started	Sunday, 22 September 2024, 10:01 PM
Completed	Sunday, 22 September 2024, 10:16 PM
Duration	15 mins 3 secs

Question 1

Correct

Marked out of 5.00

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 = $12 + 18 + 18 + 14 = 63$.

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.

Example 3:

input1 = 16

input2 = {-58, 32, 26, 92, -10, -4, 12, 0, 12, -2, 4, 32, -9, -7, 78, -79}

Expected output = 174

Explanation:

The input array contains four sequences of POSITIVE numbers, i.e. "32, 26, 92", "12, 0, 12", "4, 32", and "78". The first and second sequences "32, 26, 92" and "12, 0, 12" are the longest of the four as they contain 4 elements each. Therefore, the expected output = sum of the longest sequence of POSITIVE numbers = $(32 + 26 + 92) + (12 + 0 + 12) = 174$.

For example:

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

Answer: (penalty regime: 0 %)

```

5   int c=0,s=0,d=0;
6   int n=sc.nextInt();
7   int[] arr=new int[n];
8   int count[] =new int[n];
9   int sum[] =new int[n];
10  for(int i=0;i<n;i++){
11      int a=sc.nextInt();
12      if(a<0){
13          arr[i]=-1;
14      }
15  }

```

```

16         arr[i]=a;
17     }
18 }
19 for(int i=0;i<n;i++){
20     if(arr[i]>=0)
21     {
22         c=i+1-d;
23         s+=arr[i];
24     }
25 else{
26     d=i;
27     s=0;
28     continue;
29 }
30 count[i]=c;
31 sum[i]=s;
32 }
33 int max=count[0],z=0;
34 for(int i=0;i<count.length;i++){
35     if(count[i]>max)
36     {
37         max=count[i];
38         z=i;
39     }
40 }
41 for(int i=0;i<count.length;i++){
42     if(max==count[i]&&z!=i)
43     {
44         sum[z]+=sum[i];
45     }
46 }
47 if(sum[z]==0)
48 {
49     System.out.println("-1");
50 }
51 else{
52     System.out.println(sum[z]);
53 }
54 }
55 }
56

```

	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

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.

Example 2:

input1: 5 and input1: {1, 5, 423, 310, 61540}

Step 1:

Generating the new array based on position, we get the below array:

{1, 0, 4, 0, 6}

In this case, the value in input1 at index 1 and 3 is less than the value required to be picked up based on position, so we use a 0.

Step 2:

{1, 0, 16, 0, 36}

Step 3:

The final result = 53.

For example:

Input	Result
5 1 51 436 7860 41236	107
5 1 5 423 310 61540	53

Answer: (penalty regime: 0 %)

```

1 ↓ import java.util.*;
2 ↓ public class co{
3     public static void main(String args[])
4     {
5         Scanner sc=new Scanner(System.in);
6         int n=sc.nextInt();

```

```
7   int[] arr=new int[n];
8   int num=1;
9   for(int i=0;i<n;i++){
10      int x=sc.nextInt();
11      x/=num;
12      x%=10;
13      num*=10;
14      arr[i]=x;
15   }
16   int p=0;
17   for(int i=0;i<n;i++){
18      arr[i]=arr[i]*arr[i];
19      p+=arr[i];
20   }
21   System.out.println(p);
22 }
23 }
24 }
```

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

Passed all tests! ✓

Question 3

Correct

Marked out of 5.00

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:

$$\{(-8 \times 9), (-4 \times 9), (3 \times 9), (0 \times 9)\} = \{-72, -36, -27, 0\}$$

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:

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

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

Example 3:

input1 = 2 (represents the number of elements in the input1 array)

input2 = {-9, 9}

Expected Output = {-162, 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:

$$\{(-9 - 9), (9 - 9)\} = \{-18, 0\}$$

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

$$\{(-18 \times 9), (0 \times 9)\} = \{-162, 0\}$$

So, the expected output is the resultant array {-162, 0}.

Note: The input array will contain not more than 100 elements

For example:

Input	Result
4 1 5 6 9	-72 -36 -27 0

Input	Result
5 10 87 63 42 2	-6699 0 -2088 -3915 -7395
2 -9 9	-162 0

Answer: (penalty regime: 0 %)

```

1 import java.util.*;
2 public class main{
3     public static void main(String args[])
4     {
5         Scanner sc=new Scanner(System.in);
6         int n=sc.nextInt();
7         int[] arr=new int[n];
8         int max=0;
9         for(int i=0;i<n;i++){
10             int x=sc.nextInt();
11             arr[i]=x;
12             if(max<x)
13             {
14                 max=x;
15             }
16         }
17         for(int i=0;i<n;i++){
18             arr[i]-=max;
19             arr[i]*=max;
20         }
21         for(int i=0;i<n;i++){
22             System.out.print(arr[i]+" ");
23         }
24     }
25 }
26

```

	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! ✓

◀ Lab-03-MCQ

Jump to...

Simple Encoded Array ►

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-04-Classes and Objects](#) / [Lab-04-Logic Building](#)

Status Finished

Started Sunday, 22 September 2024, 10:17 PM

Completed Sunday, 22 September 2024, 10:37 PM

Duration 19 mins 35 secs

Question 1

Correct

Marked out of 5.00

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****For example:**

Test	Result
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

Answer: (penalty regime: 0 %)

```

1 public class Student{
2     private String name;
3     private int roll;
4     Student(){
5         this.name=null;
6         this.roll=0;
7     }
8     Student(String name){
9         this.name=name;
10
11         this.roll=0;
12     }
13     Student(String name,int roll){
14         this.name=name;
15         this.roll=roll;
16     }
17     public void display(){
18         System.out.println("Name =" +name+ " , Roll no = "+roll);
19     }
20     public static void main(String args[])
21     {
22         Student s=new Student();
23         Student s1=new Student("Rajalakshmi");
24         Student s2=new Student("Lakshmi", 101);
25         System.out.println("No-arg constructor is invoked\n1 arg constructor is invoked\n2 arg constructor is inv
26         s.display();
27         s1.display();
28         s2.display();
29     }
30 }
31

```

	Test	Expected	Got	
✓	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	✓

Passed all tests! ✓

Question 2

Correct

Marked out of 5.00

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

For example:

Test	Input	Result
1	4	Area = 50.27 Circumference = 25.13

Answer: (penalty regime: 0 %)

[Reset answer](#)

```

1 import java.io.*;
2 import java.util.*;
3 class Circle
4 {
5     private double radius;
6     public Circle(double radius){
7         // set the instance variable radius
8         this.radius=radius;
9     }
10    public void setRadius(double radius){
11        // set the radius
12        this.radius=radius;
13    }
14    public double getRadius() {
15        // return the radius
16        return radius;
17    }
18    public double calculateArea() { // complete the below statement
19        return Math.PI*radius*radius;
20    }
21    public double calculateCircumference() {
22        // complete the statement
23        return 2*Math.PI*radius;
24    }
25 }
26 class prog{
27     public static void main(String[] args) {
28         int r;
29         Scanner sc= new Scanner(System.in);
30         r=sc.nextInt();
31         Circle c= new Circle(r);
32         System.out.println("Area = "+String.format("%.2f", c.calculateArea()));
33         // invoke the calculatecircumference method
34         System.out.println("Circumference = "+String.format("%.2f",c.calculateCircumference()));
35     }
36 }
```

	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! ✓

Question 3

Correct

Marked out of 5.00

Create a Class Mobile with the attributes listed below,

```
private String manufacturer;
private String operating_system;
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.

For example:

Test	Result
1	manufacturer = Redmi operating_system = Andriod color = Blue cost = 34000

Answer: (penalty regime: 0 %)

```
1 import java.util.*;
2 public class Mobile{
3     private String manufacturer;
4     private String operating_system;
5     public String color;
6     public int cost;
7     Mobile(String m,String os,String col,int cost){
8         this.manufacturer=m;
9         this.operating_system=os;
10        this.color=col;
11        this.cost=cost;
12    }
13    public String getM(){
14        return manufacturer;
15    }
16    public String getOS(){
17        return operating_system;
18    }
19    public String getC(){
20        return color;
21    }
22    public int getcost(){
23        return cost;
24    }
25    public String toString(){
26        return "manufacturer = "+getM()+'\n'+"operating_system = "+getOS()+'\n'+"color = "+getC()+'\n'+"cost = "+cost;
27    }
28    public static void main(String args[])
29    {
30        Mobile b=new Mobile("Redmi","Andriod","Blue",34000);
31        System.out.println(b.toString());
32    }
33}
34
35
```

	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! ✓

◀ Lab-04-MCQ

Jump to...

Number of Primes in a specified range ►

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-05-Inheritance](#) / [Lab-05-Logic Building](#)

Status Finished

Started Saturday, 5 October 2024, 11:08 PM

Completed Saturday, 5 October 2024, 11:37 PM

Duration 28 mins 40 secs

Question 1

Correct

Marked out of 5.00

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

For example:

Result

```
Basic Mobile is Manufactured
Camera Mobile is Manufactured
Android Mobile is Manufactured
Camera Mobile with 5MG px
Touch Screen Mobile is Manufactured
```

Answer: (penalty regime: 0 %)

```
1 class Mobile {
2     Mobile(String name) {
3         basicmobile(name);
4     }
5
6     void basicmobile(String name) {
7         System.out.println(name + " is Manufactured");
8     }
9 }
10
11 class CameraMobile extends Mobile {
12     CameraMobile(String name) {
13         super(name);
14     }
15
16     void newfeature() {
17         System.out.println("Camera Mobile with 5MG px");
18     }
19 }
20
21 class AndroidMobile extends CameraMobile {
22     AndroidMobile(String name) {
23         super(name);
24     }
25
26     void androidMobile() {
27         System.out.println("Touch Screen Mobile is Manufactured");
```

```
28     }
29 }
30
31 public class Main {
32     public static void main(String[] args) {
33         Mobile mobile = new Mobile("Basic Mobile");
34         CameraMobile cam = new CameraMobile("Camera Mobile");
35         AndroidMobile am = new AndroidMobile("Android Mobile");
36
37         cam.newfeature();
38         am.androidMobile();
39     }
40 }
```

	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 2

Correct

Marked out of 5.00

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
```

Answer: (penalty regime: 0 %)

[Reset answer](#)

```

12     }
13
14     public void withdraw(double amount) {
15         if (balance >= amount) {
16             balance -= amount;
17         } else {
18             System.out.println("Insufficient balance");
19         }
20     }
21
22     public double getBalance() {
23         return balance;
24     }
25 }
26
27 class SavingsAccount extends BankAccount {
28     public SavingsAccount(String accountNumber, double balance) {
29         super(accountNumber, balance);
30     }
31
32     @Override
33     public void withdraw(double amount) {
34         if (getBalance() - amount < 100) {
35             System.out.println("Minimum balance of $100 required!");
36         } else {
37             super.withdraw(amount);
38         }
39     }
40 }
41
42 public class Main {
43     public static void main(String[] args) {
44         System.out.println("Create a Bank Account object (A/c No. BA1234) with initial b
45         BankAccount BA1234 = new BankAccount("BA1234", 500);
46
47         System.out.println("Deposit $1000 into account BA1234:");
48         BA1234.deposit(1000);
49         System.out.println("New balance after depositing $1000: $" + BA1234.getBalance())
50
51         System.out.println("Withdraw $600 from account BA1234:");
52         BA1234.withdraw(600);
```

```

53     System.out.println("New balance after withdrawing $600: $" + BA1234.getBalance())
54
55     System.out.println("Create a SavingsAccount object (A/c No. SA1000) with initial
56     SavingsAccount SA1000 = new SavingsAccount("SA1000", 300);
57
58     System.out.println("Try to withdraw $250 from SA1000!");
59     SA1000.withdraw(250);
60     System.out.println("Balance after trying to withdraw $250: $" + SA1000.getBalance()
61 }
62

```

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

Passed all tests! ✓

Question 3

Correct

Marked out of 5.00

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) {}
```

Expected Output:

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

For example:**Result**

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

Answer: (penalty regime: 0 %)**Reset answer**

```
1 class College {
2     String collegeName;
3
4     public College(String collegeName) {
5         this.collegeName = collegeName;
6     }
7
8     public void admitted() {
9         System.out.println("A student admitted in " + collegeName);
10    }
11 }
12
13 class Student extends College {
14     String studentName;
15     String department;
16
17     public Student(String collegeName, String studentName, String department) {
18         super(collegeName);
19         this.studentName = studentName;
20         this.department = department;
21     }
22
23     public String toString() {
24         return "CollegeName : " + collegeName + "\n" +
25                 "StudentName : " + studentName + "\n" + // Note the space after "StudentNa
26                 "Department : " + department;
27 }
```

```
27 }  
28 }  
29  
30 class CSE extends Student {  
31     public CSE(String collegeName, String studentName) {  
32         super(collegeName, studentName, "CSE");  
33     }  
34 }  
35  
36 public class Main {  
37     public static void main(String[] args) {  
38         CSE student = new CSE("REC", "Venkatesh");  
39         student.admitted();  
40         System.out.println(student);  
41     }  
42 }
```

	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! ✓

◀ Lab-05-MCQ

Jump to...

Is Palindrome Number? ►

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-06-String, StringBuffer](#) / [Lab-06-Logic Building](#)

Status Finished

Started Sunday, 6 October 2024, 11:17 AM

Completed Sunday, 6 October 2024, 11:21 AM

Duration 3 mins 58 secs

Question 1

Correct

Marked out of 5.00

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 = ww:ii:pp:rr:oo

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 = zx:za:ee

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

For example:

Input	Result
ww:ii:pp:rr:oo	WIPRO
zx:za:ee	BYE

Answer: (penalty regime: 0 %)

```

1 import java.util.*;
2
3 public class Main {
4
5     public static void main(String[] args) {
6
7         Scanner obj = new Scanner(System.in);
8         String s = obj.nextLine();
9         String[] arr = s.split(":");
10
11        for (int i = 0; i < arr.length; i++) {
12            check(arr[i]);
13        }
14
15    }
16
17    public static void check(String y) {
18
19        char a = y.charAt(0);
20        char b = y.charAt(1);
21
22        if (a == b) {
23            System.out.print(Character.toUpperCase(a));
24        } else {
25            int n1 = (int) a;
26            int n2 = (int) b;
27            int n3 = (Math.abs(n1 - n2)) + 96;
28            char c = (char) n3;
29            System.out.print(Character.toUpperCase(c));
30        }
31    }
32 }
```

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

Passed all tests! ✓

Question 2

Correct

Marked out of 5.00

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

For example:

Test	Input	Result
1	apple orange	rponlgea
2	fruits are good	utsroigfeda

Answer: (penalty regime: 0 %)

```

1 import java.util.*;
2
3 public class Main {
4
5     public static void main(String[] args) {
6
7         Scanner obj = new Scanner(System.in);
8
9         String s1 = obj.nextLine();
10        String s2 = obj.nextLine();
11
12        String s3 = s1 + s2;
13        String rs = s3.replace(" ", "");
14
15        if (rs.isEmpty()) {
16            System.out.println("null");
17            return;
18        }
19    }
  
```

```

20
21 ↓ character[] carr = new Character[rs.length()];
22     for (int i = 0; i < carr.length; i++) {
23         carr[i] = rs.charAt(i);
24     }
25
26     Arrays.sort(carr, Collections.reverseOrder());
27
28     Set<Character> charset = new LinkedHashSet<>();
29
30 ↓     // Adding sorted characters to the set to remove duplicates
31     for (char c : carr) {
32         charset.add(c);
33     }
34
35 ↓     // Printing the unique sorted characters
36     for (char c : charset) {
37         System.out.print(c);
38     }
39 }
```

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

Passed all tests! ✓

Question 3

Correct

Marked out of 5.00

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: Note that 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 = "naMng 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.

For example:

Input	Result
Today is a Nice Day 41	iNce doTday
Fruits like Mango and Apple are common but Grapes are rare 39	naMng arGpes

Answer: (penalty regime: 0 %)

```
1 import java.util.*;  
2  
3
```

```

3  public class Main {
4
5  public static void main(String[] args) {
6
7      Scanner obj = new Scanner(System.in);
8
9      String s = obj.nextLine();
10     int n = obj.nextInt();
11     int a = n / 10;
12     int b = n % 10;
13     String[] arr = s.split(" ");
14     process(arr, a);
15     System.out.print(" ");
16     process(arr, b);
17
18     obj.close();
19 }
20
21 public static void process(String[] arr, int x) {
22
23     int c = (arr[x - 1].length()) / 2;
24
25     if ((arr[x - 1].length()) % 2 == 0) { // even
26
27         for (int i = c - 1; i >= 0; i--) { // mid-left
28             System.out.print(arr[x - 1].charAt(i));
29         }
30
31         for (int j = c; j < c * 2; j++) { // mid-right
32             System.out.print(arr[x - 1].charAt(j));
33         }
34
35     } else { // odd
36
37         for (int i = c; i >= 0; i--) { // mid-left
38             System.out.print(arr[x - 1].charAt(i));
39         }
40
41         for (int j = c; j <= c * 2; j++) { // mid-right
42             System.out.print(arr[x - 1].charAt(j));
43         }
44     }
45 }
46 }
```

	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-06-MCQ](#)

Jump to...

[Return second word in Uppercase ►](#)

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-07-Interfaces](#) / [Lab-07-Logic Building](#)

Status Finished

Started Saturday, 5 October 2024, 9:23 PM

Completed Saturday, 5 October 2024, 9:55 PM

Duration 32 mins 30 secs

Question 1

Correct

Marked out of 5.00

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!

For example:

Test	Input	Result
1	Rajalakshmi Saveetha 22 21	Rajalakshmi 22 scored Saveetha 21 scored Rajalakshmi is the winner!

Answer: (penalty regime: 0 %)

[Reset answer](#)

```
1 import java.util.Scanner;
2
3 interface Sports {
4     void setHomeTeam(String name);
5     void setVisitingTeam(String name);
6 }
7
8 interface Football extends Sports {
9     void homeTeamScored(int points);
10    void visitingTeamScored(int points);
11 }
12
13 class College implements Football {
14     private String homeTeam;
15     private String visitingTeam;
16     private int homeTeamScore;
17     private int visitingTeamScore;
18
19     @Override
20     public void setHomeTeam(String name) {
21         this.homeTeam = name;
22     }
23
24     @Override
25     public void setVisitingTeam(String name) {
26         this.visitingTeam = name;
27     }
}
```

```

27
28
29     @Override
30     public void homeTeamScored(int points) {
31         this.homeTeamScore = points;
32     }
33
34     @Override
35     public void visitingTeamScored(int points) {
36         this.visitingTeamScore = points;
37     }
38
39     public void displayResult() {
40         System.out.println(homeTeam + " " + homeTeamScore + " scored");
41         System.out.println(visitingTeam + " " + visitingTeamScore + " scored");
42
43         if (homeTeamScore > visitingTeamScore) {
44             System.out.println(homeTeam + " is the winner!");
45         } else if (homeTeamScore < visitingTeamScore) {
46             System.out.println(visitingTeam + " is the winner!");
47         } else {
48             System.out.println("It's a tie match.");
49         }
50     }
51 }
52

```

	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 2

Correct

Marked out of 5.00

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.

For example:

Test	Result
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.

Answer: (penalty regime: 0 %)

```
1 interface RBI {
2
3     String pb = "RBI";
4
5     void roi();
6
7     default void policyNote() {
8         System.out.println("RBI has a new Policy issued in 2023");
9     }
10
11    static void regulations() {
12        System.out.println("RBI has updated new regulations in 2024.");
13    }
14}
15
16 class SBI implements RBI {
17
18     public void roi() {
19         System.out.println("SBI rate of interest: 7.6 per annum.");
20     }
21 }
22
23 class Karur implements RBI {
24
25     public void roi() {
26         System.out.println("Karur rate of interest: 7.4 per annum.");
27     }
28 }
```

```
29  
30 public class Main {  
31     public static void main(String[] args) {  
32         SBI s = new SBI();  
33         Karur k = new Karur();  
34  
35         s.policyNote();  
36         RBI.regulations();  
37         s.roi();  
38         k.roi();  
39     }  
40 }
```

	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! ✓

Question 3

Correct

Marked out of 5.00

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:

```
Sadvin is Playing football
Sanjay is Playing volleyball
Sruthi is Playing basketball
```

For example:

Test	Input	Result
1	Sadvin Sanjay Sruthi	Sadvin 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

Answer: (penalty regime: 0 %)

```
1 import java.util.Scanner;
2
3 interface playable {
4     void play();
5 }
6
7 class football implements playable {
8     String n;
9
10    public football(String n) {
11        this.n = n;
12    }
13
14    public void play() {
15        System.out.println(n + " is Playing football");
16    }
17 }
18
19 class volleyball implements playable {
20     String n;
21
22    public volleyball(String n) {
23        this.n = n;
24    }
25 }
```

```

25
26     public void play() {
27         System.out.println(n + " is Playing volleyball");
28     }
29 }
30
31 class basketball implements playable {
32     String n;
33
34     public basketball(String n) {
35         this.n = n;
36     }
37
38     public void play() {
39         System.out.println(n + " is Playing basketball");
40     }
41 }
42
43 public class Main {
44     public static void main(String[] args) {
45         Scanner obj = new Scanner(System.in);
46
47         String s1 = obj.next();
48         String s2 = obj.next();
49         String s3 = obj.next();
50
51         football f1 = new football(s1);
52         volleyball f2 = new volleyball(s2);

```

	Test	Input	Expected	Got	
✓	1	Sadhvin Sanjay Sruthi	Sadhvin is Playing football Sanjay is Playing volleyball Sruthi is Playing basketball	Sadhvin 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! ✓

◀ Lab-07-MCQ

Jump to...

Generate series and find Nth element ➔

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-08 - Polymorphism, Abstract Classes, final Keyword](#) / [Lab-08-Logic Building](#)

Status Finished

Started Sunday, 6 October 2024, 11:22 AM

Completed Sunday, 6 October 2024, 11:45 AM

Duration 23 mins 15 secs

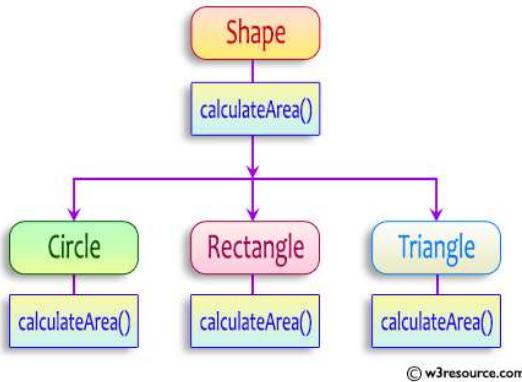
Question 1

Correct

Marked out of 5.00

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:



© w3resource.com

```

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

```

OUTPUT:

Area of a circle :50.27
Area of a Rectangle :30.00
Area of a Triangle :6.00

For example:

Test	Input	Result
1	4 5 6 4 3	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

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2
3 abstract class Shape {
4     public abstract double calculateArea();

```

```

5 }
6
7 class Circle extends Shape {
8     private double radius;
9
10 public Circle(double radius) {
11     this.radius = radius;
12 }
13
14 @Override
15 public double calculateArea() {
16     return Math.PI * radius * radius;
17 }
18 }
19
20 class Rectangle extends Shape {
21     private double length;
22     private double breadth;
23
24 public Rectangle(double length, double breadth) {
25     this.length = length;
26     this.breadth = breadth;
27 }
28
29 @Override
30 public double calculateArea() {
31     return length * breadth;
32 }
33 }
34
35 class Triangle extends Shape {
36     private double base;
37     private double height;
38
39 public Triangle(double base, double height) {
40     this.base = base;
41     this.height = height;
42 }
43
44 @Override
45 public double calculateArea() {
46     return 0.5 * base * height;
47 }
48 }
49
50 public class Main {
51     public static void main(String[] args) {
52         Scanner sc = new Scanner(System.in);

```

	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 2

Correct

Marked out of 5.00

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

For example:

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

Answer: (penalty regime: 0 %)

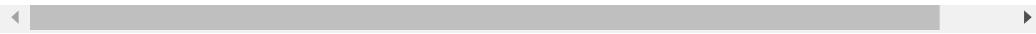
```

1 import java.util.Scanner;
2
3 public class Main {
4     // Method to check if a character is a vowel
5     public static boolean isVowel(char c) {
6         switch (Character.toLowerCase(c)) {
7             case 'a':
8             case 'e':
9             case 'i':
10            case 'o':

```

```

11     case 'u':
12         return true;
13     default:
14         return false;
15     }
16 }
17
18 public static void main(String[] args) {
19     Scanner in = new Scanner(System.in);
20
21     // Read the number of strings
22     int n = in.nextInt();
23     in.nextLine(); // Consume the newline character after the integer
24
25     // Read the strings from the next line
26     String input = in.nextLine(); // Read the entire line of strings
27     String[] a = input.split(" "); // Split the input into an array
28
29     StringBuilder result = new StringBuilder(); // Use StringBuilder for concatenation
30
31     for (String i : a) {
32         i = i.toLowerCase(); // Convert the string to lowercase
33         boolean a1 = isVowel(i.charAt(0)); // Check if first character is a vowel
34         boolean a2 = isVowel(i.charAt(i.length() - 1)); // Check if last character is a vowel
35
36         if (a1 && a2) {
37             result.append(i); // Concatenate the string
38         }
39     }
40
41     // Check if any matching strings were found
42     if (result.length() > 0) {
43         System.out.println(result.toString()); // Print the concatenated result
44     } else {
45         System.out.println("no matches found"); // No matches found
46     }
47
48     in.close(); // Close the scanner
49 }
50 }
```



	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 3

Correct

Marked out of 5.00

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
}`

Given a Java Program that contains the bug in it, your task is to clear the bug to the output.

you should delete any piece of code.

For example:

Test	Result
1	The maximum speed is: 120 km/h This is a subclass of FinalExample.

Answer: (penalty regime: 0 %)

[Reset answer](#)

```
1 class FinalExample {
2     // Final variable
3     final int maxSpeed = 120;
4
5     // Final method
6     public final void displayMaxSpeed() {
7         System.out.println("The maximum speed is: " + maxSpeed + " km/h");
8     }
9 }
10
11 class Subclass extends FinalExample {
12     // You cannot override the final method from FinalExample
13
14     // You can create new methods here
15     public void showDetails() {
16         System.out.println("This is a subclass of FinalExample.");
17     }
18 }
19
20 public class Main {
21     public static void main(String[] args) {
22         FinalExample obj = new FinalExample();
23         obj.displayMaxSpeed();
```

```
24 |
25     Subclass subObj = new Subclass();
26     subObj.showDetails();
27 }
28 }
```

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

Passed all tests! ✓

◀ Lab-08-MCQ

Jump to...

FindStringCode ►

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-09-Exception Handling](#) / [Lab-09-Logic Building](#)

Status Finished

Started Wednesday, 16 October 2024, 6:45 PM

Completed Wednesday, 16 October 2024, 7:06 PM

Duration 21 mins 15 secs

Question 1

Correct

Marked out of 5.00

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.  
Error: 37 is odd.
```

Answer: (penalty regime: 0 %)

[Reset answer](#)

```
1 class prog {  
2  
3     public static void main(String[] args) {  
4  
5         int n = 82;  
6  
7         trynumber(n);  
8  
9         n = 37;  
10        trynumber(n);  
11  
12        // call the trynumber(n);  
13    }  
14  
15    public static void trynumber(int n) {  
16  
17        try {  
18            checkEvenNumber(n);  
19            System.out.println(n + " is even.");  
20        } catch (ArithmaticException e) { // Changed 'I' to '{'  
21            System.out.println("Error: " + e.getMessage()); // Moved this line inside the  
22        }  
23    }  
24  
25    public static void checkEvenNumber(int number) { // Changed '(' to '{'  
26  
27        if (number % 2 != 0)  
28            throw new ArithmaticException(number + " is odd."); // Changed number" to number  
29    }  
30  
31}
```

	Expected	Got	
✓	82 is even. Error: 37 is odd.	82 is even. Error: 37 is odd.	✓

Passed all tests! ✓

Question 2

Correct

Marked out of 5.00

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.

```
/* Define try-catch block to save user input in the array "name"
 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.
```

For example:

Input	Result
3 5 2 1	8
2 1 g	You entered bad data.

Answer: (penalty regime: 0 %)

```
1 import java.util.Scanner;
2 import java.util.InputMismatchException;
3
4 class prog {
5
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8
9         int length = sc.nextInt();
10
11        // create an array to save user input
12        int[] name = new int[length]; // Changed 'name new int[length];' to 'int[] name = '
13
14        int sum = 0; // save the total sum of the array.
15
16        /* Define try-catch block to save user input in the array "name"
17         If there is an exception then catch the exception otherwise print the total sum
18
19         for (int i = 0; i < length; i++) {
20
21             try {
22                 name[i] = sc.nextInt();
23                 sum = sum + name[i];
24             } catch (InputMismatchException e) {
25                 System.out.println("You entered bad data.");
26                 return;
27             }
28         }
29
30         System.out.println(sum);
31     }
32 }
```

```
--  
27 }  
28 }  
29  
30     System.out.println(sum);  
31 }  
32 }
```

	Input	Expected	Got	
✓	3 5 2 1	8	8	✓
✓	2 1 g	You entered bad data.	You entered bad data.	✓

Passed all tests! ✓

Question 3

Correct

Marked out of 5.00

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

Output

java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3

I am always executed

For example:

Test	Input	Result
1	6 1 0 4 1 2 8	java.lang.ArithmaticException: / by zero I am always executed

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 public class Main {
3     public static void main(String args[]){
4         Scanner s=new Scanner(System.in);
5         int l=s.nextInt();
6         int[] a=new int[l];
7         for(int i=0;i<l;i++){
8             a[i]=s.nextInt();
9         }
10        try{
11            if(a[0]/a[1]==-1)
12                throw new ArithmaticException();
13            if(a[1]==-1)
14                throw new ArrayIndexOutOfBoundsException();
15        }
16        catch(ArithmaticException e)
17        {
18            System.out.println(e);
19        }
20        catch(ArrayIndexOutOfBoundsException e1){
21            System.out.println(e1);
22        }
23        finally{
24            System.out.println("I am always executed");
25        }
26    s.close();
27 }
28 }
```

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

Passed all tests! ✓

◀ Lab-09-MCQ

Jump to...

The "Nambiar Number" Generator ►

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-10- Collection- List](#) / [Lab-10-Logic Building](#)

Status Finished

Started Monday, 4 November 2024, 5:44 PM

Completed Monday, 4 November 2024, 6:34 PM

Duration 49 mins 26 secs

Question 1

Correct

Marked out of 1.00

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.

Answer: (penalty regime: 0 %)

```

1 import java.util.ArrayList;
2 import java.util.Scanner;
3
4 public class FirstLastElement {
5     public static void main(String[] args) {
6         Scanner scanner = new Scanner(System.in);
7         ArrayList<Integer> arrayList = new ArrayList<>();
8
9         // Read the number of elements (sample input requirement)
10        int n = scanner.nextInt();
11
12        // Read elements into the ArrayList
13        for (int i = 0; i < n; i++) {
14            arrayList.add(scanner.nextInt());
15        }
16
17        // Print the ArrayList and the first and last elements in the required format
18        System.out.println("ArrayList: " + arrayList);
19        if (!arrayList.isEmpty()) {
20            int first = arrayList.get(0);
21            int last = arrayList.get(arrayList.size() - 1);
22            System.out.println("First : " + first + ", Last : " + last);
23        }
24
25        scanner.close();
26    }
27 }
```

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

Passed all tests! ✓

Question 2

Correct

Marked out of 1.00

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.

Answer: (penalty regime: 0 %)

[Reset answer](#)

```
1 import java.util.ArrayList;
2 import java.util.Scanner;
3
4 public class Prog {
5
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         int n = sc.nextInt(); // Fixed the syntax error: '=' was missing
9
10        ArrayList<Integer> list = new ArrayList<>();
11
12        for (int i = 0; i < n; i++) {
13            list.add(sc.nextInt());
14        }
15
16        // Printing initial value of ArrayList
17        System.out.println("ArrayList: " + list);
18
19        // Replacing the element at index 1 with 100
20        list.set(1, 100);
21
22        // Getting the index of first occurrence of 100
23        int indexOf100 = list.indexOf(100); // Fixed the variable name from 'index0f100' to 'indexOf100'
24        System.out.println("Index of 100 = " + indexOf100);
25
26        // Getting the index of last occurrence of 100
27        int lastIndexOf100 = list.lastIndexOf(100);
28        System.out.println("LastIndex of 100 = " + lastIndexOf100); // Fixed the syntax error
29
30        // Check whether 200 is in the list or not
31        System.out.println(list.contains(200));
32
33        // Print ArrayList size
34        System.out.println("Size Of ArrayList = " + list.size());
35
36        // Inserting 500 at index 1
37        list.add(1, 500);
38
39        // Removing the second occurrence of 100
40        list.remove(lastIndexOf100);
41
42        System.out.println("ArrayList: " + list);
43    }
44}
```

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

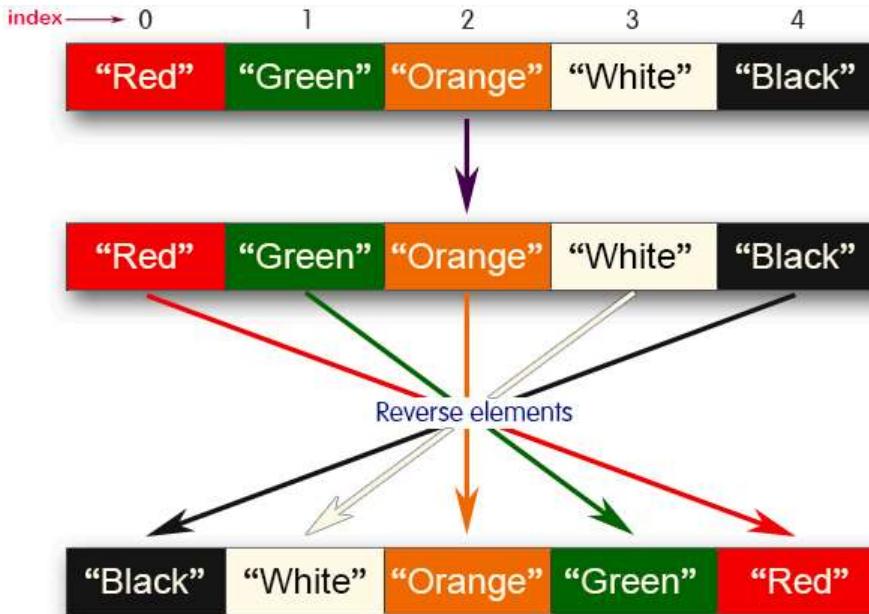
Passed all tests! ✓

Question 3

Correct

Marked out of 1.00

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]

Answer: (penalty regime: 0 %)

```

1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.Scanner;
4
5 public class ReverseArrayList {
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner(System.in);
8         ArrayList<String> arrayList = new ArrayList<>();
9
10        // Read the number of elements
11        int n = scanner.nextInt();
12        scanner.nextLine(); // Consume the newline after the integer input
13
14        // Read elements into the ArrayList
15        for (int i = 0; i < n; i++) {
16            arrayList.add(scanner.nextLine());
17        }
18
19        // Print the list before reversing
20        System.out.println("List before reversing :");
21        System.out.println(arrayList);
22
23        // Reverse the ArrayList
24        Collections.reverse(arrayList);
25

```

```

26     // Print the list after reversing
27     System.out.println("List after reversing :");
28     System.out.println(arrayList);
29
30     scanner.close();
31 }
32 }
```

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

Passed all tests! ✓

◀ Lab-10-MCQ

Jump to...

Lab-11-MCQ ►

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-11-Set, Map](#) / [Lab-11-Logic Building](#)

Status	Finished
Started	Saturday, 16 November 2024, 6:05 PM
Completed	Saturday, 16 November 2024, 6:07 PM
Duration	1 min 47 secs

Question 1

Correct

Marked out of 1.00

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.

Answer: (penalty regime: 0 %)

[Reset answer](#)

```
1 import java.util.HashSet;
2 import java.util.Scanner;
3
4 public class HashSetExample {
5     public static void main(String[] args) {
6         // Create a HashSet to store integers
7         HashSet<Integer> set = new HashSet<>();
8         Scanner scanner = new Scanner(System.in);
9
10        // Input the number of elements to add to the HashSet
11
12        int n = scanner.nextInt();
13
14        // Add elements to the HashSet
15
16        for (int i = 0; i < n; i++) {
17            int num = scanner.nextInt();
18            set.add(num);
```

```
19 }  
20  
21 // Input the element to search for  
22  
23 int searchElement = scanner.nextInt();  
24  
25 // Check if the element is in the set  
26 if (set.contains(searchElement)) {  
    System.out.println(searchElement + " was found in the set.");  
} else {  
    System.out.println(searchElement + " was not found in the set.");  
}  
31  
32 scanner.close();  
33 }  
34 }
```

	Test	Input	Expected	Got	
✓	1	5 90 56 45 78 25 78	78 was found in the set.	78 was found in the set.	✓
✓	2	3 -1 2 4 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

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

Answer: (penalty regime: 0 %)

```
1 import java.util.HashSet;
2 import java.util.Scanner;
3
4 public class CommonElementsInSets {
5     public static void main(String[] args) {
6         Scanner scanner = new Scanner(System.in);
7
8         // Read first set
9         int n1 = scanner.nextInt();
10        scanner.nextLine(); // consume newline
11        HashSet<String> set1 = new HashSet<>();
12        for (int i = 0; i < n1; i++) {
13            set1.add(scanner.nextLine());
14        }
15
16        // Read second set
17        int n2 = scanner.nextInt();
18        scanner.nextLine(); // consume newline
19        HashSet<String> set2 = new HashSet<>();
20        for (int i = 0; i < n2; i++) {
21            set2.add(scanner.nextLine());
22        }
23
24        // Retain common elements in set1
25        set1.retainAll(set2);
26
27        // Output the common elements
```

```

28     for (String element : set1) {
29         System.out.println(element);
30     }
31
32     scanner.close();
33 }
34 }
```

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

Passed all tests! ✓

Question 3

Correct

Marked out of 1.00

Java HashMap Methods

containsKey() Indicate if an entry with the specified key exists in the mapcontainsValue() Indicate if an entry with the specified value exists in the mapputIfAbsent() Write an entry into the map but only if an entry with the same key does not already existremove() Remove an entry from the mapreplace() Write to an entry in the map only if it existssize() Return the number of entries in the map

Your task is to fill the incomplete code to get desired output

Answer: (penalty regime: 0 %)[Reset answer](#)

```
1 import java.util.HashMap;
2 import java.util.Map.Entry;
3 import java.util.Set;
4 import java.util.Scanner;
5
6 class prog {
7     public static void main(String[] args) {
8         // Creating HashMap with default initial capacity and load factor
9         HashMap<String, Integer> map = new HashMap<String, Integer>();
10
11     String name;
12     int num;
13     Scanner sc = new Scanner(System.in);
14     int n = sc.nextInt();
15
16     // Populating the map with input values
17     for (int i = 0; i < n; i++) {
18         name = sc.next();
19         num = sc.nextInt();
20         map.put(name, num);
21     }
22
23     // Printing key-value pairs of the first map
24     Set<Entry<String, Integer>> entrySet = map.entrySet();
25     for (Entry<String, Integer> entry : entrySet) {
26         System.out.println(entry.getKey() + " : " + entry.getValue());
27     }
28
29     System.out.println("-----");
30
31     // Creating another HashMap
32     HashMap<String, Integer> anotherMap = new HashMap<String, Integer>();
33
34     // Inserting key-value pairs to anotherMap using put() method
35     anotherMap.put("SIX", 6);
36     anotherMap.put("SEVEN", 7);
37
38     // Inserting key-value pairs of map to anotherMap using putAll() method
39     anotherMap.putAll(map); // Fills in the missing code
40
41     // Printing key-value pairs of anotherMap
42     entrySet = anotherMap.entrySet();
43     for (Entry<String, Integer> entry : entrySet) {
44         System.out.println(entry.getKey() + " : " + entry.getValue());
45     }
46
47     // Adds key-value pair 'FIVE-5' only if it is not present in map
48     map.putIfAbsent("FIVE", 5);
```

```
49 // Retrieving a value associated with key 'TWO'  
50 int value = map.get("TWO");  
51 System.out.println(value);  
52
```

	Test	Input	Expected	Got	
✓	1	3 ONE 1 TWO ----- 2 THREE 3	ONE : 1 TWO : 2 THREE : 3 ----- SIX : 6 ONE : 1 TWO : 2 SEVEN : 7 THREE : 3 2 true true 4	ONE : 1 TWO : 2 THREE : 3 ----- SIX : 6 ONE : 1 TWO : 2 SEVEN : 7 THREE : 3 2 true true 4	✓

Passed all tests! ✓

◀ Lab-11-MCQ

Jump to...

TreeSet example ►

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-12-Introduction to I/O, I/O Operations, Object Serialization](#) / [Lab-12-Logic Building](#)

Status Finished

Started Saturday, 16 November 2024, 6:42 PM

Completed Saturday, 16 November 2024, 7:06 PM

Duration 24 mins 14 secs

Question 1

Correct

Marked out of 5.00

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.

$$98 + 99 = 197$$

$$1 + 9 + 7 = 17$$

$$1 + 7 = 8$$

For example:

Input	Result
a b c	8
b c	

Answer: (penalty regime: 0 %)

```

8     char[] input2 = {'b', 'c'};
9
10    // Call the function to get the final single digit
11    int result = getSingleDigitSum(input1, input2);
12    System.out.println(result); // Output: 8
13
14
15  public static int getSingleDigitSum(char[] input1, char[] input2) {
16      // Use a HashSet to store common characters
17      HashSet<Character> set1 = new HashSet<>();
18      HashSet<Character> commonChars = new HashSet<>();
19
20      // Add characters of input1 to set1
21      for (char c : input1) {
22          set1.add(c);
23      }
24
25      // Find common characters between input1 and input2
26      for (char c : input2) {
27          if (set1.contains(c)) {
28              commonChars.add(c);
29          }
30      }
31
32      // Calculate the sum of ASCII values of common characters
33      int sum = 0;
34      for (Character c : commonChars) {
35          sum += c - 'a' + 1;
36      }
37
38      // Get the single digit sum of the result
39      int result = sum;
40      while (result > 9) {
41          result = result / 10 + result % 10;
42      }
43
44      return result;
45  }

```

```
29     }
30 }
31
32 // Calculate the sum of ASCII values of common characters
33 int sum1 = 0;
34 for (char c : commonChars) {
35     sum1 += (int) c;
36 }
37
38 // Calculate the single digit sum
39 return getSingleDigit(sum1);
40 }
41
42 // Function to calculate the single digit sum of a number
43 public static int getSingleDigit(int sum) {
44     while (sum >= 10) {
45         sum = sumDigits(sum);
46     }
47     return sum;
48 }
49
50 // Function to calculate the sum of digits of a number
51 public static int sumDigits(int num) {
52     int sum = 0;
53     while (num > 0) {
54         sum += num % 10;
55         num /= 10;
56     }
57     return sum;
58 }
59 }
```

	Input	Expected	Got	
✓	a b c b c	8	8	✓

Passed all tests! ✓

Question 2

Correct

Marked out of 5.00

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 seigoloNhceT 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 SeigOlOnhcet 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, E and G.

NOTE:

- 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.

- 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 ,seiGolonhcet 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".

- 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 seigolonhceT erolagnaB
2	Wipro Technologies, Bangalore	0	orpiW ,seigolonhceT erolagnaB
3	Wipro Technologies Bangalore	1	Orpiw Seigolonhcet Erolagnab
4	Wipro Technologies, Bangalore	1	Orpiw ,seigolonhceT Erolagnab

For example:

Input	Result
Wipro Technologies Bangalore 0	orpiW seigolonhceT erolagnaB
Wipro Technologies, Bangalore 0	orpiW ,seigolonhceT erolagnaB
Wipro Technologies Bangalore 1	Orpiw Seigolonhcet Erolagnab
Wipro Technologies, Bangalore 1	Orpiw ,seigolonhceT Erolagnab

Answer: (penalty regime: 0 %)

```

10
11      // Loop through each word
12      for (String word : words) {
13          StringBuilder reversedWord = new StringBuilder();
14
15          // Reverse the word
16          StringBuilder tempWord = new StringBuilder(word);
17          tempWord.reverse();
18
19      }
20
21  }
```

```

18
19     // If caseOption is 1, retain the original positions of upper/lower case let
20     if (caseOption == 1) {
21         for (int i = 0; i < word.length(); i++) {
22             char originalChar = word.charAt(i);
23             char reversedChar = tempWord.charAt(i);
24
25             // If original character is uppercase, keep it uppercase in the reversed word
26             if (Character.isUpperCase(originalChar)) {
27                 reversedWord.append(Character.toUpperCase(reversedChar));
28             } else if (Character.isLowerCase(originalChar)) {
29                 reversedWord.append(Character.toLowerCase(reversedChar));
30             } else {
31                 // If non-alphabetical, keep it as is
32                 reversedWord.append(reversedChar);
33             }
34         }
35     } else {
36         // If caseOption is 0, just append the reversed characters
37         reversedWord = tempWord;
38     }
39
40     // Append the reversed word and space it correctly
41     result.append(reversedWord).append(" ");
42 }
43
44 // Remove the trailing space and return the result
45 return result.toString().trim();
46 }
47
48 public static void main(String[] args) {
49     Scanner scanner = new Scanner(System.in);
50
51     // Read sentence from the user
52     String sentence = scanner.nextLine();
53
54     // Read case option (0 or 1)
55     int caseOption = scanner.nextInt();
56
57     // Process the input and print the result
58     System.out.println(reverseWords(sentence, caseOption));
59 }
}

```

	Input	Expected	Got	
✓	Wipro Technologies Bangalore 0	orpiW seigolonhceT erolagnaB	orpiW seigolonhceT erolagnaB	✓
✓	Wipro Technologies, Bangalore 0	orpiW ,seigolonhceT erolagnaB	orpiW ,seigolonhceT erolagnaB	✓
✓	Wipro Technologies Bangalore 1	Orpiw Seigolonhcef Erolagnab	Orpiw Seigolonhcef Erolagnab	✓
✓	Wipro Technologies, Bangalore 1	Orpiw ,seigolonhceT Erolagnab	Orpiw ,seigolonhceT Erolagnab	✓

Passed all tests! ✓

Question 3

Correct

Marked out of 5.00

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 0s.

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 (0000000000000000000000000000000).

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:

The decoded string (original word) will be: WIPRO

Note: The decoded string must always be in UPPER case

For example:

Input	Result
010010001	ZYX
000010000000000000000000000000001000000000000100000000000010000000000000001	WIPRO

Answer: (penalty regime: 0 %)

```
1 import java.util.Scanner;
2
3 public class DecodeString {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         String encodedString = sc.nextLine(); // Read the encoded string
7
8         // Split the encoded string by '1' to separate sequences of '0's
9         String[] sequences = encodedString.split("1");
10        StringBuilder decodedWord = new StringBuilder();
11
12        for (String sequence : sequences) {
13            if (!sequence.isEmpty()) {
14                // Count the number of '0's in the sequence
15                int count = sequence.length();
16                // Find the corresponding letter: Z is 0 '0's, so (26 - count) gives the
17                char letter = (char) ('Z' - (count - 1));
18                decodedWord.append(letter);
19            }
20        }
21    }
22}
```

```
20     }
21
22     // Output the decoded word
23     System.out.println(decodedWord.toString());
24 }
25 }
```

	Input	Expected	Got	
✓	010010001	ZYX	ZYX	✓
✓	00001000000000000000000010000000000100000000010000000000001	WIPRO	WIPRO	✓

Passed all tests! ✓

◀ Lab-12-MCQ

Jump to...

Identify possible words ►