

**Status** Finished**Started** Wednesday, 16 October 2024, 5:36 PM**Completed** Wednesday, 16 October 2024, 5:44 PM**Duration** 8 mins 7 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 Main{
3 ↓   public static void main (String[] args){
4 ↓     Scanner sc=new Scanner(System.in);
5 ↓     int n=sc.nextInt();
6 ↓     if(n%2==0)
7 ↓     {
8 ↓       System.out.println("1");
9 ↓     }
10 ↓    else if(n%2!=0){
11 ↓      System.out.println("2");
12 ↓    }
13 ↓    else
14 ↓    {
15 ↓      System.out.println("0");
16 ↓    }
17 ↓  }
18 ↓ }
```

	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 Main{
3     public static void main(String[] args)
4     {
5         Scanner sc=new Scanner(System.in);
6         int n=sc.nextInt();
7         int l=Math.abs(n%10);
8         System.out.println(l);
9     }
10 }
```

	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: Tile 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 slim 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 Main{
3     public static void main(String[] args)
4     {
5         Scanner sc=new Scanner(System.in);
6         int n1=sc.nextInt();
7         int n2=sc.nextInt();
8         int a=Math.abs(n1%10);
9         int b=Math.abs(n2%10);
10        int c=a+b;
11        System.out.println(c);
12        sc.close();
13    }
14 }
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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, 10:58 AM

**Completed** Sunday, 22 September 2024, 11:40 AM

**Duration** 41 mins 43 secs

---

**Question 1**

Correct

Marked out of 5.00

Consider a sequence of the form 0, 1, 1, 2, 4, 7, 13, 24, 44, 81, 149...

Write a method program which takes as parameter an integer n and prints the nth term of the above sequence. The nth term will fit in an integer value.

Example Input:

5

Output:

4

Example Input:

8

Output:

24

Example Input:

11

Output:

149

**For example:**

Input	Result
5	4
8	24
11	149

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

```

1 import java.util.Scanner;
2 public class CustomSequence
3 {
4     public static int findterm(int n)
5     {
6         if(n==0) return 0;
7         if(n==1||n==2) return 1;
8         int[] sequence=new int[n+1];
9         sequence[0]=0;
10        sequence[1]=1;
11        sequence[2]=1;
12        for(int i=3;i<=n;i++)
13        {
14            sequence[i]=sequence[i-3]+sequence[i-2]+sequence[i-1];
15        }
16        return sequence[n-1];
17    }
18    public static void main(String[] args)
19    {
20        Scanner scanner=new Scanner(System.in);
21        int n=scanner.nextInt();
22        int nthTerm=findterm(n);
23        System.out.println(nthTerm);
24    }
25 }
```

	Input	Expected	Got	
✓	5	4	4	✓

	Input	Expected	Got	
✓	8	24	24	✓
✓	11	149	149	✓

Passed all tests! ✓

**Question 2**

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 public class Timetable{
3     public static void time(int day, boolean vacay){
4         if(vacay == false){
5             if(day == 1 || day == 7){
6                 System.out.println("6:00");
7             }
8             else{
9                 System.out.println("5:00");
10            }
11        }
12        else{
13            if(day == 1|| day == 7){
14                System.out.println("9:00");
15            }
16            else{
17                System.out.println("7:00");
18            }
19        }
20    }
21    public static void main(String[] args){
22        Scanner sc = new Scanner(System.in);
23        String input = sc.nextLine();
24    }
}

```

```
24     String[] parts = input.split(" ");
25     int day = Integer.parseInt(parts[0]);
26     boolean vacay = Boolean.parseBoolean(parts[1]);
27     time(day,vacay);
28 }
29 }
30 }
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	1 false	6:00	6:00	✓
✓	5 false	5:00	5:00	✓
✓	1 true	9:00	9:00	✓

Passed all tests! ✓

//

**Question 3**

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 %)

```

1 // Java program to count trailing 0s in n!
2 import java.io.*;
3 import java.util.Scanner;
4 class prog {
5     // Function to return trailing
6     // 0s in factorial of n
7     static int findTrailingZeros(int n)
8     {
9         if (n < 0) // Negative Number Edge Case
10             return -1;
11
12         // Initialize result
13         int count = 0;
14
15
16         // Keep dividing n by powers
17         // of 5 and update count
18         for (int i = 5; n / i >= 1;i *=5)
19             count += n / i;
20
21         return count;
22     }
23 }
```

```
24 // Driver Code
25 public static void main(String[] args)
26 {
27
28     Scanner sc= new Scanner(System.in);
29     int n = sc.nextInt();
30     System.out.println(findTrailingZeros(n));
31 }
32
33 }
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	3	0	0	✓
✓	60	14	14	✓
✓	100	24	24	✓
✓	1024	253	253	✓

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, 11:40 AM

**Completed** Sunday, 22 September 2024, 12:54 PM

**Duration** 1 hour 13 mins

---

**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 %)

```

1 import java.util.Scanner;
2 import java.util.Arrays;
3 public class Main{
4     public static int largestSum(int n, int arr[]){
5         int maxlen = 0;
6         int currentLength = 0;
7         int maxSum = 0;
8         int currentSum = 0;
9         boolean haspos = false;

```

```

10
11     for(int i = 0;i<n;i++){
12         if(arr[i]>=0){
13             haspos = true;
14             currentSum += arr[i];
15             currentLength++;
16         }
17     else{
18         if(currentLength == maxlen) maxSum += currentSum;
19     else if(currentLength > maxlen){
20             maxlen = currentLength;
21             maxSum = currentSum;
22         }
23         currentSum = 0;
24         currentLength = 0;
25     }
26
27 }
28 if(currentLength == maxlen) maxSum += currentSum;
29 else if(currentLength > maxlen) maxSum = currentSum;
30
31 return haspos? maxSum: -1;
32 }
33 public static void main(String[] args){
34     Scanner sc = new Scanner(System.in);
35     int n = sc.nextInt();
36     sc.nextLine();
37
38     String[] starray = sc.nextLine().split(" ");
39     int[] arr = new int[n];
40     for(int i = 0;i<n;i++){
41         arr[i] = Integer.parseInt(starray[i]);
42     }
43
44     System.out.println(largestSum(n,arr));
45 }
46 }
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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 0<sup>th</sup> index of the array pick up digits as per below:

0<sup>th</sup> index – pick up the units value of the number (in this case is 1).

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

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

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

4<sup>th</sup> 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.Scanner;
2 public class NumberArraySum
3 {
4     public static int calculateSum(int[] input1, int input2)
5     {

```

```

6   int[] extractedArray=new int[input2];
7   for(int i=0;i<input2;i++)
8   {
9       int num=input1[i];
10      int position=i;
11      extractedArray[i]=extractedDigitAtPosition(num,position);
12  }
13  for(int i=0;i<input2;i++)
14  {
15      extractedArray[i]=extractedArray[i]*extractedArray[i];
16  }
17  int sum=0;
18  for(int i=0;i<input2;i++)
19  {
20      sum+=extractedArray[i];
21  }
22  return sum;
23 }
24 public static int extractedDigitAtPosition(int num,int position)
25 {
26     int divisor=(int) Math.pow(10,position);
27     return(num/divisor)%10;
28 }
29 public static void main(String[] args)
30 {
31     Scanner sc=new Scanner(System.in);
32     int input2_1 = sc.nextInt();
33     sc.nextLine();
34     String[] input1Str_1= sc.nextLine().split(" ");
35     int[] input1_1=new int[input2_1];
36     for(int i=0;i<input2_1;i++)
37     {
38         input1_1[i]=Integer.parseInt(input1Str_1[i]);
39     }
40     int result1=calculateSum(input1_1,input2_1);
41     System.out.println(result1);
42 }
43 }
44 }
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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.Scanner;
2 public class ArrayOperattions
3 {
4     public static int[] performoperation(int[] input2,int input1)
5     {
6         int max=findMax(input2);
7         for(int i=0;i<input1;i++)
8         {
9             input2[i]=input2[i]-max;
10        }
11        for(int i=0;i<input1;i++)
12        {
13            input2[i]=input2[i]*max;
14        }
15        return input2;
16    }
17    public static int findMax(int[] array)
18    {
19        int max=array[0];
20        for(int i=1;i<array.length;i++)
21        {
22            if(array[i]>max)
23            {
24                max=array[i];
25            }
26        }
27        return max;
28    }
29    public static void main(String[] args)
30    {
31        Scanner scanner=new Scanner(System.in);
32        int input1=scanner.nextInt();
33        int[]input2=new int[input1];
34        for (int i=0;i<input1;i++)
35        {
36            input2[i]=scanner.nextInt();
37        }
38        int[] result= performoperation(input2,input1);
39        for(int i:result)
40        {
41            System.out.print(i+" ");
42        }
43        scanner.close();
44    }
45 }
```

	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, 8:49 PM

**Completed** Sunday, 22 September 2024, 9:23 PM

**Duration** 34 mins 9 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 rollno;
4
5     public Student(){
6         System.out.println("No-arg constructor is invoked");
7         this.name = null;
8         this.rollno = 0;
9     }
10    public Student(String name){
11        System.out.println("1 arg constructor is invoked");
12        this.name = name;
13        this.rollno = 0;
14    }
15
16    public Student(String name,int rollno){
17        System.out.println("2 arg constructor is invoked");
18        this.name = name;
19        this.rollno = rollno;
20    }
21    public void display(){
22        System.out.println("Name =" +this.name+ " , Roll no = "+this.rollno);
23    }
24    public static void main(String[] args){
25        Student std1 = new Student();
26        Student std2 = new Student("Rajalakshmi");
27        Student std3 = new Student("Lakshmi", 101);
28        std1.display();
29        std2.display();
30        std3.display();
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 =  $\pi r^2$**

**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.util.Scanner;
2 class Circle
3 {
4     private double radius;
5     public Circle(double radius){
6         this.radius = radius;
7
8     }
9     public void setRadius(double radius){
10        // set t
11        this.radius = radius;
12
13    }
14
15    public double getRadius()    {
16        // return the radius
17        return this.radius;
18
19    }
20
21
22    }
23    public double calculateArea() { // complete the below statement
24        return Math.PI*radius*radius;
25    }
26
27    public double calculateCircumference()    {
28        // complete the statement
29        return 2*Math.PI*radius;
30    }
31 }
32 class prog{
33     public static void main(String[] args)  {
34         int r;
35         Scanner sc= new Scanner(System.in);
36         r=sc.nextInt();
37         Circle c= new Circle(r);
38         System.out.println("Area = "+String.format("%.2f", c.calculateArea()));
39         System.out.println("Circumference = "+String.format("%.2f",c.calculateCircumference()));
40         // invoke the calculatecircumference method
41
42
43     }
44 }
```

	<b>Test</b>	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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 public class Mobile{
2     private String manufacturer;
3     private String operating_system;
4     public String color;
5     private int cost;
6
7     public Mobile(String manufacturer, String operating_system, String color, int cost){
8         this.manufacturer = manufacturer;
9         this.operating_system = operating_system;
10        this.color = color;
11        this.cost = cost;
12    }
13    public String getManufacturer(){
14        return manufacturer;
15    }
16    public String getOperatingSystem(){
17        return operating_system;
18    }
19
20    public void setOperatingSystem(String operating_system){
21        this.operating_system= operating_system;
22    }
23
24    public String getColor(){
25        return color;
26    }
27    public void setColor(String color){
28        this.color = color;
29    }
30    public int getCost(){
31        return cost;
32    }
33    public void setCost(int cost){
34        this.cost = cost;
35    }
36
37    public String toString(){
38        return "manufacturer =" + manufacturer + "\noperating_system =" + operating_system +
39        "\ncolor =" + color+
```

```
40     "\ncost = "+cost;
41 }
42
43 public static void main(String[] args){
44     Mobile mobile = new Mobile(" Redmi "," Andriod "," Blue ",34000);
45     System.out.println(mobile.toString());
46 }
47 }
```

	<b>Test</b>	<b>Expected</b>	<b>Got</b>	
✓	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** Monday, 30 September 2024, 5:24 PM

**Completed** Monday, 30 September 2024, 6:27 PM

**Duration** 1 hour 2 mins

---

**Question 1**

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) {}
public toString()
```

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 {
3     protected String collegeName;
4
5     public College(String collegeName) {
6         // initialize the instance variables
7         this.collegeName=collegeName;
8
9     }
10
11    public void admitted() {
12        System.out.println("A student admitted in "+collegeName);
13    }
14 }
15 class Student extends College{
16
17     String studentName;
18     String department;
19
20     public Student(String collegeName, String studentName, String depart) {
21         // initialize the instance variables
22         super(collegeName);
23         this.studentName=studentName;
24         this.department=depart;
25
26     }
27
28
29    public String toString(){
30        // return the details of the student
31        return "CollegeName : "+ collegeName + "\nStudentName : "+ studentName + "\nDepartment : "+ department;
32
33    }
34 }
```

```
35 public class Main {  
36     public static void main (String[] args) {  
37         Student s1 = new Student("REC","Venkatesh","CSE");  
38         s1.admitted();  
39         System.out.println(s1.toString());  
40     }  
41 }
```

	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

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](#)

```
1 class BankAccount {  
2     // Private field to store the account number  
3     private String accountNumber;  
4  
5     // Private field to store the balance  
6     private double balance;  
7  
8     // Constructor to initialize account number and balance  
9     public BankAccount(String accountNumber, double balance)  
10    {  
11        this.accountNumber=accountNumber;  
12        this.balance=balance;  
13    }  
14  
15  
16  
17  
18  
19     // Method to deposit an amount into the account  
20     public void deposit(double amount) {  
21         // Increase the balance by the deposit amount  
22         balance+=amount;  
23     }  
24  
25  
26     // Method to withdraw an amount from the account  
27     public void withdraw(double amount) {  
28         // Check if the balance is sufficient for the withdrawal  
29         if (balance >= amount) {  
30             // Decrease the balance by the withdrawal amount  
31             balance -= amount;  
32         } else {  
33             // Print a message if the balance is insufficient  
34             System.out.println("Insufficient balance");  
35         }  
36     }  
37  
38     // Method to get the current balance  
39     public double getBalance() {  
40         // Return the current balance  
41         return balance;  
42     }  
43 }  
44  
45  
46 class SavingsAccount extends BankAccount {  
47     // Constructor to initialize account number and balance  
48     public SavingsAccount(String accountNumber, double balance)  
49     {
```

```

50     // Call the parent class constructor
51     super(accountNumber,balance);
52

```

	<b>Expected</b>	<b>Got</b>	
✓	<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! ✓

**Question 3**

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 {
3     public Mobile()
4     {
5         System.out.println("Basic Mobile is Manufactured");
6     }
7     public void basicMobile()
8     {
9         System.out.println("Basic Mobile features");
10    }
11 }
12 class CameraMobile extends Mobile
13 {
14     public CameraMobile()
15     {
16         super();
17         System.out.println("Camera Mobile is Manufactured");
18     }
19     public void newFeature()
20     {
21         System.out.println("Camera Mobile with 5MG px");
22     }
23 }
24 class AndroidMobile extends CameraMobile
25 {
26     public AndroidMobile()
27     {
28         super();
29         System.out.println("Android Mobile is Manufactured");
30     }
31     public void androidMobile()
32     {
33         System.out.println("Touch Screen Mobile is Manufactured");
34     }
35 }
```

```
36 }  
37 public class main  
38 {  
39     public static void main(String[] args)  
40     {  
41         AndroidMobile androidMobile=new AndroidMobile();  
42  
43         androidMobile.newFeature();  
44         androidMobile.androidMobile();  
45     }  
46 }
```

	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! ✓

◀ 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** Monday, 30 September 2024, 10:09 PM

**Completed** Monday, 30 September 2024, 10:46 PM

**Duration** 37 mins 30 secs

---

**Question 1**

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 = "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.

**For example:**

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

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

```

1 import java.util.Scanner;
2 public class SringProcessor
3 {
4     public static String processWord(String word)
5     {
6         int len = word.length();
7         int mid = len/2;
8         String middleToBegin = word.substring(0,mid+(len%2));
9         String middleToEnd=word.substring(mid);
10        return new StringBuilder(middleToBegin).reverse().toString()+middleToEnd;

```

```

11 }
12 }
13 public static String processInput(String input1, int input2)
14 {
15     String[] words=input1.split(" ");
16     int firstWordIndex=input2 / 10-1;
17     int secondWordIndex=input2 % 10-1;
18     String firstProcessed=processWord(words[firstWordIndex]);
19     String secondProcessed=processWord(words[secondWordIndex]);
20     return firstProcessed + " " + secondProcessed;
21 }
22 public static void main(String[] args)
23 {
24     Scanner sc=new Scanner(System.in);
25     String input1=sc.nextLine();
26     int input2=sc.nextInt();
27     String result=processInput(input1,input2);
28     System.out.println(result);
29     sc.close();
30 }
31 }
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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! ✓

**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 public class StringManipulation
3 {
4     public static String processStrings(String input1, String input2)
5     {
6         String concatenated = input1 + input2;
7         Set<Character> uniqueChars = new HashSet<>();
8         for(char c : concatenated.toCharArray())
9         {
10             if(Character.isLetter(c))
11             {
12                 uniqueChars.add(c);
13             }
14         }
15         List<Character> sortedChars = new ArrayList<>(uniqueChars);
16         Collections.sort(sortedChars, Collections.reverseOrder());
17         if(sortedChars.isEmpty())
18         {
19             return "null";
20         }
21         StringBuilder result = new StringBuilder();
22         for(char c : sortedChars)
23         {
24             result.append(c);
25         }
26         return result.toString();
27     }
}

```

```
28     public static void main(String[] args)
29     {
30         Scanner sc=new Scanner(System.in);
31         String input1=sc.nextLine();
32         String input2=sc.nextLine();
33         String output=processStrings(input1,input2);
34         System.out.println(output);
35     }
36 }
```

	<b>Test</b>	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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

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 2<sup>nd</sup> 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 25<sup>th</sup> 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.Scanner;
2 public class AlphabetGame
3 {
4     public static char getCharFromPositionDifference(char a,char b)
5     {
6         int posA=a-'a'+1;
7         int posB=b-'a'+1;
8         int difference=Math.abs(posA-posB);
9         return (char)('a'+ difference-1);
10    }
11    public static String processInput(String input)
12    {
13        String[] words=input.split(":");
14        StringBuilder result=new StringBuilder();
15        for(String word:words)
16        {
17            if(word.length()==2)
18            {
19                char firstChar=word.charAt(0);
20                char secondChar=word.charAt(1);
21                if(firstChar==secondChar)
22                {
23                    result.append(firstChar);
24                }
25                else
26                {
27                    result.append(getCharFromPositionDifference(firstChar,secondChar));
28                }
29            }
30            else
31            {
32                System.out.println("Skipping invalid word:"+word);
33            }
34        }
35        return result.toString().toUpperCase();
36    }
37    public static void main(String[] args)
38    {
39        Scanner sc=new Scanner(System.in);
40        String input=sc.nextLine();
41        String output=processInput(input);
42        System.out.println(output);
43    }
44 }
```

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

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** Monday, 30 September 2024, 8:56 PM

**Completed** Monday, 30 September 2024, 9:39 PM

**Duration** 43 mins 8 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 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
19     }
20     public void setVisitingTeam(String name){
21         visitingTeam=name;
22     }
23     }
24     public void homeTeamScored(int points){
25         System.out.println(homeTeam+" "+points+" scored");
26     }
27     public void visitingTeamScored(int points){
28         System.out.println(visitingTeam+" "+points+" scored");
29     }
30     public void winningTeam(int p1, int p2){
31         if(p1>p2)
32             {
33                 System.out.println(homeTeam+" is the winner!");
34             }
35     }
36 }
```

```

35     else if(p1<p2)
36     {
37         System.out.println(visitingTeam+" is the winner!");
38     }
39
40
41     else
42     {
43         System.out.println("It's a tie match.");
44     }
45
46 }
47 }
48 public class Main{
49 public static void main(String[] args){
50     Scanner sc= new Scanner(System.in);
51     String hname=sc.nextLine();
52     String vteam=sc.nextLine();

```

	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

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 interface Playable
3 {
4     void play();
5 }
6 class Football implements Playable
7 {
8     String name;
9     public Football(String name)
10 {
11     this.name=name;
12 }
13 @Override
14 public void play()
15 {
16     System.out.println(name+" is Playing football");
17 }
18 }
19 class Volleyball implements Playable
20 {
21     String name;
22     public Volleyball(String name)
23 {
24     this.name=name;
25 }
26 @Override
27 public void play()
28 {
29     System.out.println(name+" is Playing volleyball");
30 }
31 }
32 class Basketball implements Playable
```

```

33 {
34     String name;
35     public Basketball(String name)
36     {
37         this.name=name;
38     }
39     @Override
40     public void play()
41     {
42         System.out.println(name+" is Playing basketball");
43     }
44 }
45 public class Main
46 {
47     public static void main(String[] args)
48     {
49         Scanner sc=new Scanner(System.in);
50         String FPname=sc.nextLine();
51         String VPname=sc.nextLine();
52         String BPname=sc.nextLine();

```

	<b>Test</b>	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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! ✓

//

**Question 3**

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 parentBank="RBI";
4     double rateOfInterest();
5     default void policyNote()
6     {
7         System.out.println("RBI has a new Policy issued in 2023");
8     }
9     static void regulations()
10    {
11        System.out.println("RBI has updated new regulations in 2024.");
12    }
13 }
14
15 class SBI implements RBI
16 {
17     @Override
18     public double rateOfInterest()
19     {
20         return 7.6;
21     }
22 }
23 class Karur implements RBI
24 {
25     @Override
26     public double rateOfInterest()
27     {
28         return 7.4;
29     }
30 }
31 public class Main
32 {
33     public static void main(String[] args)
34     {
35         RBI rbi=new SBI();
36         rbi.policyNote();
```

```
-- 37     RBI.regulations();  
38     SBI sbi=new SBI();  
39     System.out.println("SBI rate of interest: "+ sbi.rateOfInterest() + " per annum.");  
40     Karur karur=new Karur();  
41     System.out.println("Karur rate of interest: " + karur.rateOfInterest() + " per annum.");  
42 }  
43 }
```

	<b>Test</b>	<b>Expected</b>	<b>Got</b>	
✓	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-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** Friday, 4 October 2024, 6:14 PM

**Completed** Friday, 4 October 2024, 8:17 PM

**Duration** 2 hours 2 mins

---

**Question 1**

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.*;
2 public class VowelStringConcatenation
3 {
4     public static boolean isVowel(char c)
5     {
6         return "AEIOUaeiou".indexOf(c)!=-1;
7     }
8     public static String concatenateVowelStrings(String[] arr)
9     {
10
11         StringBuilder result = new StringBuilder();
12         for(String str : arr)
13         {
14             if(!str.isEmpty() && isVowel(str.charAt(0)) && isVowel(str.charAt(str.length() -1)))
15             {
16                 result.append(str);
17             }
18         }
19     }
20 }
```

```

19     if(result.length()==0)
20     {
21         return "no matches found";
22     }
23     return result.toString().toLowerCase();
24 }
25
26
27 /*StringBuilder result=new StringBuilder();
28 for(String str:arr)
29 {
30
31     boolean allVowels=true;
32     for(int i=0;i<str.length();i++)
33     {
34         if(!isVowel(str.charAt(i)))
35         {
36             allVowels=false;
37             break;
38         }
39     }
40     if(allVowels)
41     {
42         result.append(str);
43     }
44 }
45
46
47 return result.toString().toLowerCase();
48
49
50 }
51 public static boolean isVowel(char c)
52 {

```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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

**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 public class FinalExample
2 {
3     final int MAX_SPEED=120;
4     public final void display()
5     {
6         System.out.println("The maximum speed is: "+ MAX_SPEED+ " km/h");
7     }
8     public static void main(String[] args)
9     {
10        SubExample obj=new SubExample();
11        obj.display();
12        obj.show();
13    }
14 }
15 class SubExample extends FinalExample
16 {
17     public void show()
18     {
19         System.out.println("This is a subclass of FinalExample.");
20     }
21 }
22
23

```

	<b>Test</b>	<b>Expected</b>	<b>Got</b>	
✓	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! ✓

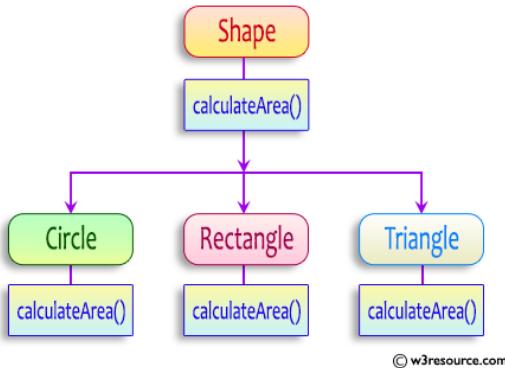
Question 3

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:



```

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 abstract class Shape
3 {
4     public abstract double calculateArea();
5 }
6 class Circle extends Shape
7 {
8     private double radius;
9     public Circle(double radius)
10    {
11        this.radius=radius;
12    }
13 }
  
```

```

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

```

	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! ✓

◀ Lab-08-MCQ

Jump to...

FindStringCode ▶

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

---

**Status** Finished

**Started** Wednesday, 9 October 2024, 2:42 PM

**Completed** Wednesday, 9 October 2024, 2:44 PM

**Duration** 1 min 28 secs

---

**Question 1**

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	8
5 2 1	
2	You entered bad data.
1 g	

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

```
1 import java.util.Scanner;
2 import java.util.InputMismatchException;
3
4 public class ArraySum {
5     public static void main(String[] args) {
6         Scanner scanner = new Scanner(System.in);
7
8         try {
9
10             int n = scanner.nextInt(); // Read the size of the array
11
12             int[] array = new int[n]; // Initialize the array
13
14             int sum = 0;
15             for (int i = 0; i < n; i++) {
16                 array[i] = scanner.nextInt(); // Read integers into the array
17                 sum += array[i]; // Calculate the sum
18             }
19
20             // If no exception occurs, print the total sum
21             System.out.println(sum);
22         } catch (InputMismatchException e) {
23             // Handle the exception if non-integer input is entered
24             System.out.println("You entered bad data.");
25         } finally {
26             scanner.close(); // Close the scanner
27         }
28     }
29 }
30
31 }
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	3 5 2 1	8	8	✓
✓	2 1 g	You entered bad data.	You entered bad data.	✓

Passed all tests! ✓

**Question 2**

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 public class EvenOddChecker {  
2  
3     public static void main(String[] args) {  
4         checkNumber(82);  
5         checkNumber(37);  
6     }  
7  
8     // Method to check if the number is even or odd  
9     public static void checkNumber(int number) {  
10        try {  
11            if (isOdd(number)) {  
12                throw new IllegalArgumentException(number + " is odd.");  
13            } else {  
14                System.out.println(number + " is even.");  
15            }  
16        } catch (IllegalArgumentException e) {  
17            System.out.println("Error: " + e.getMessage());  
18        }  
19    }  
20  
21    // Method to determine if a number is odd  
22    public static boolean isOdd(int number) {  
23        return number % 2 != 0;  
24    }  
25}  
26  
27
```

	<b>Expected</b>	<b>Got</b>	
✓	82 is even. Error: 37 is odd.	82 is even. Error: 37 is odd.	✓

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
3 public class ExceptionHandlingDemo {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6
7         // Read the size of the array
8
9         int size = scanner.nextInt();
10
11        // Create an array of the specified size
12        int[] array = new int[size];
13
14        // Read elements into the array
15
16        for (int i = 0; i < size; i++) {
17            array[i] = scanner.nextInt();
18        }
19
20        // Attempt to access the elements and perform the division
21        try {
22            // This will intentionally access an out-of-bounds index
23            int outOfBoundsAccess = array[3]; // Example access beyond the valid range
24
25            // Now perform the division only if the access is within bounds
26            int result = array[0] / array[1];
27            System.out.println("Result: " + result);
28        } catch (ArithmaticException e) {
29            System.out.println("java.lang.ArithmaticException: " + e.getMessage());
30        } catch (ArrayIndexOutOfBoundsException e) {
31            System.out.println("java.lang.ArrayIndexOutOfBoundsException: " + e.getMessage());
32        }
33
34        // Always executed message
35        System.out.println("I am always executed");
36
37        // Close the scanner

```

```

38     // close the scanner
39     scanner.close();
40 }
41

```

	<b>Test</b>	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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 ►

//

# CS23333-Object Oriented Programming Using Java-2023

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

## Lab-10-Logic Building

 Done

Attempts allowed: 2

Time limit: 2 hours

Grading method: Highest grade

### Your attempts

#### Attempt 1

<b>Status</b>	Finished
<b>Started</b>	Wednesday, 30 October 2024, 10:21 AM
<b>Completed</b>	Wednesday, 30 October 2024, 11:01 AM
<b>Duration</b>	40 mins 41 secs

[Review](#)

[◀ 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** Tuesday, 12 November 2024, 12:22 PM

**Completed** Tuesday, 12 November 2024, 12:32 PM

**Duration** 9 mins 46 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 scanner object for input  
7         Scanner sc = new Scanner(System.in);  
8  
9         // Read the number of elements to insert into the HashSet  
10        int n = sc.nextInt();  
11  
12        // Create a HashSet object to store integers  
13        HashSet<Integer> numbers = new HashSet<>();  
14  
15        // Read 'n' elements and add them to the HashSet  
16        for (int i = 0; i < n; i++) {  
17            numbers.add(sc.nextInt());  
18        }  
19  
20        // Read the number to search for in the HashSet  
21        int skey = sc.nextInt();  
22  
23        // Check if the number exists in the HashSet and print the result  
24        if (numbers.contains(skey)) {  
25            System.out.println(skey + " was found in the set.");  
26        } else {  
27            System.out.println(skey + " was not found in the set.");  
28        }  
29    }  
30}
```

```
27     }  
28         System.out.println(skey + " was not found in the set.");  
29     }  
30     // Close the scanner  
31     sc.close();  
32 }  
33 }  
34 }
```

	<b>Test</b>	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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 SetComparison {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         // Create the first HashSet
9         HashSet<String> set1 = new HashSet<>();
10
11        // Read the size of the first set
12        int n1 = sc.nextInt();
13        sc.nextLine(); // Consume the newline character left by nextInt()
14
15        // Add elements to the first set
16        for (int i = 0; i < n1; i++) {
17            set1.add(sc.nextLine());
18        }
19
20        // Create the second HashSet
21        HashSet<String> set2 = new HashSet<>();
22
23        // Read the size of the second set
24        int n2 = sc.nextInt();
25        sc.nextLine(); // Consume the newline character
26
27        // Add elements to the second set
28        for (int i = 0; i < n2; i++) {
29            set2.add(sc.nextLine());
30        }
31
32        // Perform the intersection of set1 and set2
33        set1.retainAll(set2); // This will retain only the common elements
34
35        // Output the common elements
```

```

55
56     // Output the common elements
57     for (String element : set1) {
58         System.out.println(element);
59     }
60
61     sc.close(); // Close the scanner to prevent memory leaks
62 }
63
64
65

```

	<b>Test</b>	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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 public class Main { // Ensure the class is named "Main"
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     // Inputting key-value pairs into the map
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 in the 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 into anotherMap using put() method
35     anotherMap.put("SIX", 6);
36     anotherMap.put("SEVEN", 7);
37
38     // Inserting key-value pairs from 'map' into 'anotherMap' using putAll() method
39     anotherMap.putAll(map); // Put all entries from map into anotherMap
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
50     // Retrieving a value associated with key 'TWO'
51     Integer value = map.get("TWO");
52     System.out.println(value); // Will print null if "TWO" does not exist
```

	<b>Test</b>	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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](#)

<b>Status</b>	Finished
<b>Started</b>	Friday, 8 November 2024, 5:58 PM
<b>Completed</b>	Friday, 8 November 2024, 6:36 PM
<b>Duration</b>	37 mins 41 secs

**Question 1**

Incorrect

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:**

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 ,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".

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 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 %)

```

1 import java.util.*;
2
3 public class ReverseWordsWithCase {
4
5     public static void main(String[] args) {
6         // Scanner to accept input
7         Scanner sc = new Scanner(System.in);
8
9         // Input sentence and case_option
10        String sentence = sc.nextLine();
11        int caseOption = sc.nextInt();
12
13        // Call function to reverse words and handle case
14        String result = reverseWords(sentence, caseOption);
15
16        // Output the result
    
```

```

17     System.out.println(result);
18 }
19
20 // Function to reverse words based on case_option
21 public static String reverseWords(String sentence, int caseOption) {
22     // Split the sentence into words based on spaces
23     String[] words = sentence.split(" ");
24     StringBuilder result = new StringBuilder();
25
26     // Process each word
27     for (String word : words) {
28         String reversedWord = reverseWord(word, caseOption);
29         result.append(reversedWord).append(" ");
30     }
31
32     // Remove trailing space
33     return result.toString().trim();
34 }
35
36 // Function to reverse a single word and apply case_option
37 public static String reverseWord(String word, int caseOption) {
38     StringBuilder reversedWord = new StringBuilder(word);
39
40     // Reverse the word
41     reversedWord.reverse();
42
43     // Apply the case option
44     if (caseOption == 0) {
45         return reverseCase(reversedWord.toString()); // Reverse case for case_option 0
46     } else if (caseOption == 1) {
47         return capitalizeFirstLetterAndLastLetter(reversedWord.toString()); // Handle case_option 1
48     }
49
50     return reversedWord.toString(); // Default case, should not happen
51 }
52

```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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 Seigolonhcet Erolagnab	Orpiw Seigolonhcet Erolagnab	✓
✗	Wipro Technologies, Bangalore 1	Orpiw ,seigolonhceT Erolagnab	Orpiw ,seigolonhceT Erolagnab	✗

Your code must pass all tests to earn any marks. Try again.

Show differences

## Question 2

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 (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:

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 Decoder {
4
5     public static void main(String[] args) {
6         Scanner scanner = new Scanner(System.in);
7
8         // Taking input string from user
9         String encoded = scanner.nextLine();
10
11        // Decoding the string
12        String decoded = decode(encoded);
13
14        // Printing the decoded word
15        System.out.println( decoded);
16
17        scanner.close();
18    }
19
20    private static String decode(String encoded) {
21        // Split the string by '1', which separates the sequences of 0's
22        String[] zeroSequences = encoded.split("1");
23
24        StringBuilder decoded = new StringBuilder();
25
26        // Iterate through each zero sequence
```

```
27 for (String seq : zeroSequences) {  
28     if (!seq.isEmpty()) {  
29         // Calculate the length of the zero sequence  
30         int length = seq.length();  
31  
32         // Map the length of zeros to the corresponding alphabet letter  
33         char letter = (char) ('Z' - (length - 1)); // 'Z' corresponds to 1 zero  
34         decoded.append(letter);  
35     }  
36 }  
37  
38 return decoded.toString();  
39 }  
40 }  
41 }
```

Passed all tests! ✓



**Question 3**

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 %)

```

1 import java.util.HashSet;
2
3 public class CommonCharactersSum {
4
5     public static void main(String[] args) {
6         // Example input
7         char[] input1 = {'a', 'b', 'c'};
8         char[] input2 = {'b', 'c'};
9
10        // Function call to process and get the final result
11        int result = getSingleDigitSumOfCommonASCIIValues(input1, input2);
12
13        // Output the result
14        System.out.println(result);
15    }
16
17    public static int getSingleDigitSumOfCommonASCIIValues(char[] input1, char[] input2) {
18        // Step 1: Find common characters using a set
19        HashSet<Character> set1 = new HashSet<>();
20        for (char c : input1) {
21            set1.add(c);
22        }
23
24        HashSet<Character> commonChars = new HashSet<>();
25        for (char c : input2) {
26            if (set1.contains(c)) {
27                commonChars.add(c);
28            }
29        }
    }

```

```
30 // Step 2: Calculate the sum of ASCII values of common characters
31 int sum1 = 0;
32 for (char c : commonChars) {
33     sum1 += (int) c; // Get ASCII value of the character
34 }
35
36 // Step 3: Calculate the single digit sum (digital root)
37 return digitalRoot(sum1);
38 }
39
40 // Method to calculate the digital root (single-digit sum)
41 public static int digitalRoot(int n) {
42     if (n == 0) {
43         return 0;
44     }
45     return 1 + (n - 1) % 9;
46 }
47
48 }
```

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

Passed all tests! ✓

◀ Lab-12-MCQ

Jump to...

Identify possible words ►

//