RAJALAKSHMI ENGINEERING COLLEGE
THANDALAM - 602 105

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

CS23333 Object Oriented Programming Using Java

Laboratory Record Notebook

Name: VAISHNAVI R S

Year / Branch / Section: 2nd year / B.Tech AIML – 'C'

University Register No: 2116231501178

College Roll No: 231501178

Semester: IIIrd Semester

Academic Year: 2023 - 2024

RAJALAKSHMI ENGINEERING COLLEGE

An Autonomous Institution
Affiliated to Anna University, Chennai,
Rajalakshmi Nagar, Thandalam – 602 105

BONAFIDE CERTIFICATE

Name: VAISHNAVI R S

Academic Year:2024-2025 Semester: III Branch: B.Tech- AIML

Register No. 231501178

Certified that this is the bonafide record of work done by the above student in the CS23333- Object OrientedProgramming Using Java Laboratory during the academic year 2024- 2025

Signature of Faculty in-charge

Submitted for the Practical Examination held on 2/11/2024.....

Internal Examiner External Examiner

Status Finished

Started Thursday, 19 September 2024, 11:12 AM
Completed Thursday, 19 September 2024, 11:22 AM

Duration 10 mins 41 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.io.*;
  import java.util.*;
2
```

```
 3   public class Odd{
 4public static void main(String[] args)
 5  {
 6Scanner sc=new Scanner(System.in);
 7int a=sc.nextInt();
 8if(a%2==1 || a%2==-1)
 9  {
10System.out.println(2);
11}
12else if(a%2==0)
13  {
14System.out.println(1);
15}
16else if(a==0)
17  {
18System.out.println(1);
19}
20}
21 }
```

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

Passed all tests! 1

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.io.*;
   import java.util.*;
2
3     import java.math.*;
4       public class Last{
5     public static void main(String[] args)
6  {
er sc=new Scanner(System.in);
c.nextInt();
h.abs(a);
m.out.println(a%10);
11}
12 }
```

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

Passed all tests! 1

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
-2	11

67 15 4	
-2 67 -1 54	11

Answer: (penalty regime: 0 %)

```
import
java.io.*;
import
java.util.*;
import
java.math.*;
public class
add{
    public static void main(String[] args)
    {
        Scanner sc=new
        Scanner(System.in); int
        a=sc.nextInt();
        int
        b=sc.nextInt();

        a=Math.abs(a
        );
        b=Math.abs(b);
        int c=(a%10)+(b%10);
        System.out.println(c);
    }
}
```

4 15	

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

Passed all tests! 1

Lab-01-MCQ

Jump to...

Is Even?

Status Finished

Started Saturday, 21 September 2024, 10:12 AM
Completed Saturday, 21 September 2024, 10:57 AM

Duration 45 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

```
// Java program to count trailing 0s in n!
import java.io.*;
import java.util.*;
class prog {
    // Function to return trailing
    // 0s in factorial of n
    static int findTrailingZeros(int n)
    {
        int count=0;
        if (n < 0) // Negative Number Edge Case

            return -1;



        // Initialize result



        // Keep dividing n by powers

        // of 5 and update count

        for (int i = 5; n / i >= 1;i*=5)

            count += n / i;
```

```
19
20
21          return count;
22      }
23
```

```
24  // Driver Code
25  public static void
    main(String[] args)
26  {
27      int n ;
28      Scanner sc= new
        Scanner(System.in);
29      n=sc.nextInt();
30      int x=findTrailingZeros(n);
31      System.out.println(x);
32  }
33
34
```

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

Passed all tests! 1

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 %)

```
import
java.io.*;
import
java.util.*;
public class
Num{
    public static void main(String[] args)
    {
        Scanner sc=new
        Scanner(System.in); int
        n=sc.nextInt();
        String
        st=Integer.toString(n);
        char[]
        arr=st.toCharArray();
        for(int i=0;i<arr.length;i+
        +)
        {
            switch(arr[i])
            {
                case '0':
                    System.out.print("Zer
                    o "); break;
                case '1':
                    System.out.print("On
                    e "); break;
                case '2':
                    System.out.print("Tw
                    o "); break;
                case '3':
                    System.out.print("Thr
                    ee "); break;
```

```
                case '4':
                    System.out.print("Fo
                    ur "); break;
                case '5':
                    System.out.print("Fiv
                    e "); break;
                case '6':
                    System.out.print("Six
                     "); break;
                case '7':
                    System.out.print("Se
                    ven "); break;
                case '8':
                    System.out.print("Eig
                    ht "); break;
                case '9':
                    System.out.print("Nine ");
```

```
4
3
4
```

	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! 1

Question 3

Correct

Marked out of 5.00

Consider the following sequence:

1st term: 1

2nd term: 1 2 1

3rd term: 1 2 1 3 1 2 1

4th term: 1 2 1 3 1 2 1 4 1 2 1 3 1 2 1

And so on. Write a program that takes as parameter an integer n and prints the nth terms of this sequence. Example Input:

1

Output:

1

Example Input:

4

Output:

1 2 1 3 1 2 1 4 1 2 1 3 1 2 1

For example:

Input	Result
1	1
2	1 2 1
3	1 2 1 3 1 2 1
4	1 2 1 3 1 2 1 4 1 2 1 3 1 2 1

Answer: (penalty regime: 0 %)

```
1 import java.io.*;
```

```
    2    import java.util.*;
    3      public class pattern{
    4    public static void main(String[] args)
    5  {
er sc=new Scanner(System.in);
c.nextInt();
res="1";
i=1;i<n;i++)
   10  {
   11res+=" "+(i+1)+" "+res;
   12}
   13System.out.println(res);
   14}
   15 }
```

	Input	Expected	Got	
✓	1	1	1	✓
✓	2	1 2 1	1 2 1	✓

	Input	Expected	Got	
✓	3	1 2 1 3 1 2 1	1 2 1 3 1 2 1	✓
✓	4	1 2 1 3 1 2 1 4 1 2 1 3 1 2 1	1 2 1 3 1 2 1 4 1 2 1 3 1 2 1	✓

Passed all tests! 1

Lab-02-MCQ

Jump to...

Lab-03-MCQ

Status Finished

Started Sunday, 22 September 2024, 8:33 PM
Completed Sunday, 22 September 2024, 9:43 PM

Duration 1 hour 9 mins

Question 1
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.io.*;
   import
   java.util.*;
2  public class arraysp{
       public static void main(String[] args)
3      {
           Scanner sc=new Scanner(System.in);
4
5
```

```
7
8   int sum=0;
9   int n=sc.nextInt();
10  int[] arr=new int[n];
11  for(int i=0;i<n;i++)
12  {
13      arr[i]=sc.nextInt();
14  }
15  int[] p=new int[n];
16  for(int i=0;i<n;i++)
17  {
18      p[i]=(arr[i]/(int)
        Math.pow(10,i)) %10;
19  }
20  for(int i:p)
21  {
22      sum+=i*i;
    }
```

23			System.out.println(sum);
24		}	
25	}		

	Input	Expected	Got	
✓	5 1 51 436	107	10	✓

Passed all tests! 1

Question 2

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

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

Example 2:

input1 = 5 (represents the number of

elements in the input1 array) input2 = {10,

87, 63, 42, 2}

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

Explanation:

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

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

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

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

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

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

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 x 9), (0 x 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 %)

```
import
java.io.*;
import
java.util.*;
public class arraychange{
    public static void main(String[] args)
    {
        Scanner sc=new
        Scanner(System.in); int
        n=sc.nextInt();
        int[] arr= new
        int[n]; for(int
        i=0;i<n;i++)
        {
            arr[i]=sc.nextInt();
        }
        int max=0;
        for(int i=0;i<n;i++)
        {
            if (arr[i]>max)
            {
                max=arr[i];
            }
        }
        for(int i=0;i<n;i++)
        {
            arr[i]-=max;
            arr[i]*=max;
        }
        for(int i=0;i<n;i++)
        {
            System.out.print(arr[i]+ " ");
        }
    }
}
```

```
import
java.io.*;
import
java.util.*;
public class arraychange{
    public static void main(String[] args)
    {
        Scanner sc=new
        Scanner(System.in); int
        n=sc.nextInt();
        int[] arr= new
        int[n]; for(int
        i=0;i<n;i++)
        {
            arr[i]=sc.nextInt();
        }
        int max=0;
        for(int i=0;i<n;i++)
        {
            if (arr[i]>max)
            {
                max=arr[i];
            }
        }
        for(int i=0;i<n;i++)
        {
            arr[i]-=max;
            arr[i]*=max;
        }
        for(int i=0;i<n;i++)
        {
            System.out.print(arr[i]+ " ");
        }
    }
}
```

	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! 1

Question 3

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.io.*;
  import
  java.util.*;
2 public class arraypos{
      public static void main(String[] args)
3     {
          Scanner sc=new
          Scanner(System.in); int
          n=sc.nextInt();
4         int[] arr=new
          int[n]; int
5         maxl=0;
          int cl=0;


6
```

```
11          int csum=0;
12          int tsum=0;
13          for(int i=0;i<n;i++)
14          {
                arr[i]=sc.nextInt(
                );
15          }
16          for(int i=0;i<n;i++)
17          {
18              if(arr[i]>0)
                {
                    cl++;
19                  csum+=arr[i];
20              }
                else
                {
21                  if(cl>maxl)
22                  {
23                      maxl=cl;
24                      tsum=csu
25                      m;
                    }
                    else
26                  if(cl==maxl)
27                  {
                        tsum+=cs
                        um;
```

	Input	Expected	Got	
✓	16 -12 -16 12 18 18 14 -4 -12 -13 32 34 -5 66 78 78 -79	62	62	✓

Passed all tests! 1

Lab-03-MCQ

Jump to...

Simple Encoded Array

Status Finished

Started Sunday, 22 September 2024, 10:32 PM
Completed Sunday, 22 September 2024, 11:31 PM

Duration 58 mins 48 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(Stri

ng 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 %)

```
39 }
40
```

	Test	Expected	Got	
✓	1	No-arg constructor is invoked	No-arg constructor is invoked	✓
		1 arg constructor is invoked 2 arg constructor is invoked	1 arg constructor is invoked 2 arg constructor is invoked	
		Name =null , Roll no = 0	Name =null , Roll no = 0	
		Name =Rajalakshmi , Roll no = 0 Name =Lakshmi , Roll no = 101	Name =Rajalakshmi , Roll no = 0 Name =Lakshmi , Roll no = 101	

Passed all tests! 1

Question 2

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

getManufactur

er(){ 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 %)

```
40 }
```

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

Passed all tests! 1

Question 3

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 = πr2

Circumferen

ce = 2πr

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          this.radius=radius;
8
9
```

```
10      }
11      public void setRadius(double radius){
12          this.radius=radius;
13
14
15      }
16      public double getRadius(){
17          return radius;
18
19
20      }
21      public double calculateArea() { // complete the below statement
22          return Math.PI*radius*radius;
23
24      }
25      public double calculateCircumference(){
26          return 2*Math.PI*radius;
2       }
```

```
7
28  }
29  class prog{
30      public static void main(String[] args) {
31          int r;
32          Scanner sc= new Scanner(System.in);
33          r=sc.nextInt();
34          Circle c= new Circle(r);
35          System.out.println("Area = "+String.format("%.2f", c.calculateArea()));
36          System.out.println("Circumference = " +String.format("%.
            2f",c.calculateCircumference()));
37
38
39      }
40  }
41
```

	Tes	Input	Expected	Got	

	t				
✓	1	4	Area = 50.27 Circumferenc e = 25.13	Area = 50.27 Circumferenc e = 25.13	✓
✓	2	6	Area = 113.10 Circumferenc e = 37.70	Area = 113.10 Circumferenc e = 37.70	✓
✓	3	2	Area = 12.57 Circumferenc e = 12.57	Area = 12.57 Circumferenc e = 12.57	✓

Passed all tests! 1

Lab-04-MCQ

Jump to...

Number of Primes in a specified range

Status Finished

Started Sunday, 6 October 2024, 7:02 PM
Completed Sunday, 6 October 2024, 7:07 PM

Duration 5 mins 27 secs

Question 1
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

Answer: (penalty regime: 0 %)

Reset answer

```
1  class BankAccount {
2      private String accountNumber;
3      private double balance;
4
5      public BankAccount(String accountNumber, double initialBalance) {
6          this.accountNumber = accountNumber;
7          this.balance = initialBalance;
8      }
9
10     public void deposit(double amount) {
11     balance += amount;
12     // Format the output correctly
13     System.out.println("New balance after depositing $" + (amount % 1 == 0 ?
       String.format("%.0f", amount) : Strin
14 }
15
```

```
16
17      public void withdraw(double amount) {
18          if (balance >= amount) {
19              balance -= amount;
20              // Format the output correctly
21              System.out.println("New balance after withdrawing $" + (amount % 1 == 0 ?
                String.format("%.0f", amount
22          } else {
23              System.out.println("Insufficient funds!");
24          }
25      }
26
27      public double getBalance() {
28          return balance;
29      }
30  }
31
32  class SavingsAccount extends BankAccount {
33      private final double minimumBalance = 100.0;
```

BankAccount BA1234 = new

```
34
35      public SavingsAccount(String accountNumber, double initialBalance) {
36          super(accountNumber, initialBalance);
37      }
38
39      @Override
40      public void withdraw(double amount) {
41          if (getBalance() - amount >= minimumBalance) {
42              super.withdraw(amount);
43          } else {
44              System.out.println("Minimum balance of $" + String.format("%.0f",
                minimumBalance) + " required!");
45          }
46      }
47  }
48
49  public class Main {
50      public static void main(String[] args) {
```

5 System.out.println("Create a Bank Account object (A/c No. BA1234) with

	Expected	Got	
✓	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	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	✓

Passed all tests! 1

Question 2

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

collegeNam

e; public

College() { }

public

admitted()

{ } Student:

String

studentNa

me; String

departmen

t;

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 :

Answer: (penalty regime: 0 %)

Reset answer

```
class College {
    protected String collegeName;

    public College(String collegeName) {
        this.collegeName = collegeName;
    }

    public void admitted() {
        System.out.println("A student admitted in " + collegeName);
    }
}

class Student extends College {
    String studentName;

    String department;

    public Student(String collegeName, String studentName, String department) {
        super(collegeName);

        this.studentName = studentName;
```

```
9
20          this.department = department;
21      }
22
23      @Override
24      public String toString() {
25          return "CollegeName : " + collegeName + "\n" +
26                  "StudentName : " + studentName + "\n" +
27                  "Department : " + department;
28      }
29  }
30
31  public class sample {
32      public static void main(String[] args) {
33          Student s1 = new Student("REC", "Venkatesh", "CSE");
34          s1.admitted(); // Print "A student admitted in REC"
35          System.out.println(s1);
```

```
36}
37 }
```

	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! 1

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

Answer: (penalty regime: 0 %)

```
2       public Mobile() {
3           System.out.println("Basic Mobile is Manufactured");
4       }
5
6       public void basicMobile() {
7           System.out.println("Basic Mobile functionality");
8       }
9   }
10
11  class CameraMobile extends Mobile {
12      public CameraMobile() {
36  }
37  }
```

	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! 1

Lab-05-MCQ

Jump to...

Is Palindrome Number?

Status Finished

Started Sunday, 6 October 2024, 7:09 PM
Completed Sunday, 6 October 2024, 7:12 PM

Duration 3 mins 36 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:o

o 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
Explanat
ion
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.Scanner;
2
3   public class Main {
4       public static void main(String[] args)
5       {
6           Scanner sc = new Scanner(System.in);
7           String s = sc.nextLine();
8           String[] words = s.split(":");
9           StringBuilder output = new StringBuilder();
10          for (String i : words)
11          {
1               char ch1 = i.charAt(0);
```

```
2
13          char ch2 = i.charAt(1);
14
15          if (ch1 == ch2)
16          {
17  output.append(Character.toUpperCase(ch1))
                                                ;
18          }
19          else
20          {
21              int pos1 = ch1 - 'a' + 1;
22              int pos2 = ch2 - 'a' + 1;
23
24              int max = Math.max(pos1, pos2);
25              int min = Math.min(pos1, pos2);
26
27              int position = max - min;
28      char result = (char) ('A' + position - 1);
2
```

```
9
30                  output.append(result);
31              }
32          }
33
34          System.out.println(output.toString());
35      }
36  }
```

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

Passed all tests! 1

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 oran	rponlgea

	ge	
2	fruits are good	utsroi gfeda

Answer: (penalty regime: 0 %)

```
1  import java.util.*;
2
3  public class StringMergeSort
4  {
5      public static String mergeAndSort(String input1, String input2)
6      {
7          String concatenated = input1 + input2;
8          Set<Character> uniqueChars = new HashSet<>();
9          for (char ch : concatenated.toCharArray())
10         {
11             if (ch != ' ')
12             {
13                 uniqueChars.add(ch);
14             }
15         }
16
```

```
17
18          List<Character> sortedList = new ArrayList<>(uniqueChars);
19          Collections.sort(sortedList, Collections.reverseOrder());
20
21          StringBuilder result = new StringBuilder();
22          for (char ch : sortedList)
23          {
24              result.append(ch);
25          }
26          return result.length() > 0 ? result.toString() : "null";
27      }
```

```
28
29      public static void main(String[] args)
30      {
31          Scanner scanner = new
            Scanner(System.in);
32
```

```
2
3
3
4        String input1 = scanner.nextLine();
5
6        String input2 = scanner.nextLine();
7
8        String result =
         mergeAndSort(input1, input2);
9        System.out.println(result);
0        scanner.close();
1    }
2}
```

	Test	Input	Expected	Got	
✓	1	apple orange	rponlgea	rponlgea	✓
✓	2	fruit	utsroi	utsroi	✓

		s are goo d	gfeda	gfeda	
✓	3		null	null	✓

Passed all tests! 1

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 = "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	naMng o arGpes

Answer: (penalty regime: 0 %)

```
1  import java.util.Scanner;
2
3  public class WordProcessor {
```

```
4     public static void main(String[]
                                    args) {
5                 Scanner sc = new
                  Scanner(System.in);
6
7         String input = sc.nextLine();
8         int number = sc.nextInt();
9         String[] words = input.split(" ");
10
11        int pos1 = number / 10;
12        int pos2 = number % 10;
13
14        pos1--;
15        pos2--;
16
17        String result1 = processWord(words[pos1]);
18        String result2 = processWord(words[pos2]);
19
20        String result = result1 + " " + result2;
2         System.out.println(result);
```

```
1
2
2     }
2
3
2
4     private static String processWord(String word) {
2
5         int len = word.length();
2
6         int mid = len / 2;
2
7
2
8         String middleToBegin;
2
9         String middleToEnd;
3
0
3
1         if (len % 2 == 0)
3
2         {
3
3             middleToBegin = new StringBuilder(word.substring(0,
              mid)).reverse().toString();
3
4             middleToEnd = word.substring(mid);
3
5         }
3
3         else
```

```
    {
        middleToBegin = new StringBuilder(word.substring(0, mid + 1)).reverse().toString();
        middleToEnd = word.substring(mid);
    }
    return middleToBegin + middleToEnd;
  }
}
```

	Input	Expected	Got	
✓	Today is a Nice Day 41	iNce doTday	iNce doTday	✓

Passed all tests! 1

Lab-06-MCQ

Jump to...

Return second word in Uppercase

Status Finished

Started Sunday, 6 October 2024, 7:13 PM
Completed Sunday, 6 October 2024, 7:17 PM

Duration 4 mins 48 secs

Question 1

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:

```
Sadhvin is Playing
football Sanjay is
```

For example:

Tes	Input	Result

t		
1	Sadhvin Sanjay Sruthi	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

Answer: (penalty regime: 0 %)

```
1  import java.util.Scanner;
2
3  interface Playable
4  {
5      void play();
6  }
7
8  class Football implements Playable {
9      String name;
10
1      public Football(String name)
```

```
1
12      {
13          this.name = name;
14      }
15
16      public void play()
17      {
18          System.out.println(name + " is Playing football");
19      }
20  }
21
22  class Volleyball implements Playable
23  {
24      String name;
25
26      public Volleyball(String name)
27      {
28          this.name = name;
```

```
    }

    public void play()
    {
        System.out.println(name + " is Playing volleyball");
    }
}

class Basketball
{
    String name;
    public Basketball(String name)
    {
        this.name = name;
    }

    public void play()
}

public class
```

	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	Vijay is Playing football Arun is	Vijay is Playing football Arun is	✓

Passed all tests! 1

Question 2
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:
Rajal
aksh
mi
Save
etha
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  {
5      public void setHomeTeam(String name);
6      public void setVisitingTeam(String name);
7  }
8
9  interface Football extends Sports
10 {
11     public void homeTeamScored(int points);
12
13     public void visitingTeamScored(int points);
```

```
2
13  }
14
15  class College implements Football
16  {
17      String homeTeam;
18      String visitingTeam;
19
20      public void setHomeTeam(String name)
21      {
22          homeTeam = name;
23      }
24
25      public void setVisitingTeam(String name)
26      {
27          visitingTeam = name;
28      }
29
```

```
30      public void homeTeamScored(int points)
31      {
32          System.out.println(homeTeam + " " + points + " scored");
33      }
34
35      public void visitingTeamScored(int points)
36      {
37          System.out.println(visitingTeam + " " + points +
            " scored");
38      }
39
40      public void winningTeam(int homeTeamPoints, int
        visitingTeamPoints)
41      {
42          if (homeTeamPoints > visitingTeamPoints)
43          {
```

```
44          System.out.println(homeTeam + " is the winner!");
45      }
46      else if (homeTeamPoints < visitingTeamPoints)
47      {
48          System.out.println(visitingTeam + " is the winner!");
49      }
50      else
51      {
52          System.out.println("It's a tie match.");
```

	Test	Input	Expected	Got	
✓	1	Rajalakshmi	Rajalakshmi 22 scored	Rajalakshmi 22 scored	✓
		Saveetha	Saveetha 21 scored	Saveetha 21 scored	
		22	Rajalakshmi is the winner!	Rajalakshmi is the winner!	
		21			
✓	2	Anna	Anna 21 scored	Anna 21 scored	✓
		Balaji	Balaji 21 scored	Balaji 21 scored	

		21	It's a tie match.	It's a tie match.	
		21			
✓	3	SRM	SRM 20 scored	SRM 20 scored	✓
		VIT	VIT 21 scored	VIT 21 scored	
		20	VIT is the winner!	VIT is the winner!	
		21			

Passed all tests! 1

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 %)

```
interface RBI
{
    String parentBank = "RBI";

    double rateOfInterest();

     default void policyNote()
    {
        System.out.println("RBI has a new Policy issued in 2023");
    }

    static void regulations()
    {
```

```
14          System.out.println("RBI has updated new regulations in 2024.");
15      }
16  }
17
18  class SBI implements RBI
19  {
20      public double rateOfInterest()
21      {
22          return 7.6;
23      }
24  }
25
26  class Karur implements RBI
27  {
28      public double rateOfInterest()
29      {
30          return 7.4;
31      }
```

```
32  }
33
34  public class test
35  {
36      public static void main(String[] args)
37      {
```

```
38          SBI sbiBank = new SBI();
39          Karur karurBank = new Karur();
40
41          sbiBank.policyNote();
42          RBI.regulations();
43
44          System.out.println("SBI rate of interest: " + sbiBank.rateOfInterest() + "
            per annum.");
45          System.out.println("Karur rate of interest: " +
            karurBank.rateOfInterest() + " per annum.");
46      }
47
```

	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! 1

Lab-07-MCQ

Jump to...

Generate series and find Nth element

Status Finished

Started Wednesday, 16 October 2024, 8:25 PM

Completed Wednesday, 16 October 2024, 8:30 PM

Duration 5 mins 6 secs

Question 1

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). 1 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

```
class FinalExample {


    final int maxSpeed = 120;


    public final void displayMaxSpeed() {
        System.out.println("The maximum speed is: " + maxSpeed + " km/h");
    }
}


class SubClass extends FinalExample {


    public void showDetails() {
        System.out.println("This is a subclass of FinalExample.");

    }
}


class prog {
```

```
20      public static void main(String[] args) {
21          FinalExample obj = new FinalExample();
22          obj.displayMaxSpeed(); // This will print the maximum speed
23
24          SubClass subObj = new SubClass();
25          subObj.showDetails(); // This will print the subclass details
26      }
27  }
```

	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! 1

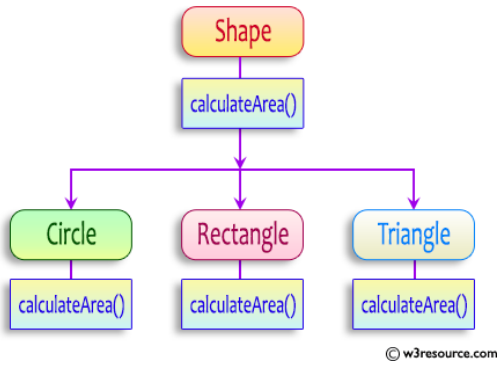
Question 2

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	Area of a circle: 50.27
	5	Area of a Rectangle: 30.00
	6	Area of a Triangle: 6.00
	4	
	3	
2	7	Area of a circle: 153.94
	4.5	Area of a Rectangle: 29.25

	6.5	Area of a Triangle: 4.32
	2.4	
	3.6	

Answer: (penalty regime: 0 %)

```
import java.util.Scanner;


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

class Circle extends Shape {
    private double radius;

    public Circle(double radius) {
        this.radius = radius;
    }
```

```

    @Override
    public double calculateArea() {
        return Math.PI * radius * radius;
```

```
6
17      }
18  }
19
20  class Rectangle extends Shape {
21      private double length;
22      private double breadth;
23      public Rectangle(double length,
24      double breadth) {
25          this.length = length;
26          this.breadth = breadth;
27      }
28
29      @Override
        public double
30          calculateArea()
            { return length *
            breadth;
```

```
31
32      }
33  }
34
35  class Triangle extends Shape {
36      private double base;
37      private double height;
38      public Triangle(double base, double
39  height) {
40          this.base = base;
41          this.height = height;
42      }
43
44
45      @Override
    public double
```

```
46      calculateArea() { return
            0.5 * base * height;
47
48  }
49  }
50
```

	Test	Input	Expected	Got	
✓	1	4	Area of a circle: 50.27	Area of a circle: 50.27	✓
		5	Area of a Rectangle: 30.00	Area of a Rectangle: 30.00	
		6	Area of a	Area of a	

Passed all tests! 1

Question 3

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 VowelEndStrings {
4      public static void main(String[] args)
5      {
6          Scanner sc = new Scanner(System.in);
7  19{ int n = sc.nextInt();
```

```
20         if ("aeiouAEIOU".indexOf(i.charAt(0)) != -1 &&
           "aeiouAEIOU".indexOf(i.charAt(i.length() - 1)) != -1)
21         {
22             s += i;
23             found = true;
24         }
25     }
26
27     if (found)
```

```
28          {
29              System.out.println(s.toLowerCase());
30          }
31          else
32          {
33              System.out.println("no matches found");
34          }
35
36          sc.close();
37      }
38  }
```

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

	Ate Ace Girl			

Passed all tests! 1

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, 8:31 PM
Completed Wednesday, 16 October 2024, 8:37 PM

Duration 6 mins 17 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 5 2 1	8
2 1 g	You entered bad data.

Answer: (penalty regime: 0 %)

Reset answer

```
import java.util.Scanner;
import java.util.InputMismatchException;
  class prog {
  public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
int length = sc.nextInt();
int[] name = new int[length];
int sum=0;
try
{
for(int i=0;i<length;i++){
name[i] = sc.nextInt();
sum+=name[i];
}
System.out.println(sum);
}
catch(InputMismatchException e)
{
System.out.println("You entered bad data.");
}
}
```

	Input	Expected	Got	
✓	3 5 2 1	8	8	✓

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

Passed all tests! 1

Question 2
Correct
Marked out of 5.00

Write a Java program to handle ArithmeticException 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.ArithmeticException: / 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	java.lang.ArithmeticException: / by zero
	1 0 4 1 2 8	I am always executed

Answer: (penalty regime: 0 %)

```
1  import java.util.Scanner;
2
3  public class I
4  {
5      public static void main(String[] args)
6      {
7          Scanner sc = new Scanner(System.in);
8
9          int n = sc.nextInt();
10         int[] arr = new int[n];
11         for (int i = 0; i < n; i++) {
12             arr[i] = sc.nextInt();
13         }
```

```
14
15          try
16          {
17              int result = arr[0] / arr[1];
18
19
20              System.out.println(arr[3]);
21          }
22          catch (ArithmeticException e)
23          {
24              System.out.println("java.lang.ArithmeticException: " + e.getMessage());
25          }
26          catch (ArrayIndexOutOfBoundsException e)
27          {
28              System.out.println("java.lang.ArrayIndexOutOfBoundsException: " +
                e.getMessage());
29          }
30          finally
31          {
```

```
                System.out.println("I am always executed");
            }
        }
}
```

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

Passed all tests! 1

Question 3
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.
```

Fill the preloaded answer to get the expected output.

For example:

Result
82 is even.

Answer: (penalty regime: 0 %)

Reset answer

```
1  class prog
2  {
3      public static void main(String[] args)
4      {
5          int n = 82;
6          trynumber(n);
7          n = 37;
8          trynumber(n);
9      }
1
```

```
0
1
1       public static void trynumber(int n)
12      {
1
3           try
14          {
1
5               checkEvenNumber(n); // Call the checkEvenNumber() method
1
6               System.out.println(n + " is even.");
1
7           }
1
8           catch (IllegalArgumentException e)
19          {
2
0               System.out.println("Error: " + e.getMessage());
2
1           }
2
2       }
2
3
2
4       public static void checkEvenNumber(int number)
25      {
2
6           if (number % 2 != 0)
27          {
2               throw new IllegalArgumentException(number + " is odd.");
```

```
8
29          }
30      }
31  }
```

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

Passed all tests! 1

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, 8:28 AM
Completed Monday, 4 November 2024, 8:50 AM

Duration 21 mins 47 secs

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
```

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 %)

```
import
java.util.*;
public class
Main{
    public static void main(String[] args){
        Scanner scanner=new
        Scanner(System.in); int
        n=scanner.nextInt();
        ArrayList<Integer>arrayList=new
        ArrayList<>(); for(int i=0;i<n;i++)
        {
            arrayList.add(scanner.nextInt());
        }
        if(!arrayList.isEmpty())
        {
            int first=arrayList.get(0);
            int
            last=arrayList.get(arrayList.size()-1);
             System.out.println("ArrayList:
            "+arrayList);
            System.out.println("First : "+first+", Last : "+last);
        }
        else
        {
            System.out.println("The ArrayList is empty:");
        }
    }
}
```

```
13
14
15
16
17
18
19
20
21
22
23
```

	Test	Input	Expected	Got	
✓	1	6	ArrayList: [30, 20, 40, 50, 10, 80]	ArrayList: [30, 20, 40, 50, 10, 80]	✓

		30	First : 30, Last : 80	First : 30, Last : 80	
		20			
		40			
		50			
		10			
		80			
✓	2	4	ArrayList: [5, 15, 25, 35]	ArrayList: [5, 15, 25, 35]	✓
		5	First : 5, Last : 35	First : 5, Last : 35	
		15			
		25			
		35			

Passed all tests! 1

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.lastInd

exOf())

list.contain

s()

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.*;
2		import java.io.*;
3		
4		class prog {
5		public static void main(String[] args)
6		{

```
7           Scanner sc= new Scanner(System.in);
8           int n = sc.nextInt();
9
10          ArrayList<Integer> list = new ArrayList<Integer>();
11        for(int i = 0; i<n;i++){
12            list.add(sc.nextInt());
13        }
14        System.out.println("ArrayList: " + list);
15        list.set(1,100);
16        System.out.println("Index of 100 = "+list.indexOf(100));
17
18        //Getting the index of last occurrence of 100
19        System.out.println("LastIndex of 100 = "+list.lastIndexOf(100));
20        // Check whether 200 is in the list or not
21        System.out.println(list.contains(200)); //Output : false
22         // Print ArrayList size
23        System.out.println("Size Of ArrayList = "+ list.size());
24        //Inserting 500 at index 1
25        list.add(1,500);// code here
26         //Removing an element from position 3
27        list.remove(3);// code here
28        System.out.print("ArrayList: " + list);
29    }
30  }
```

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

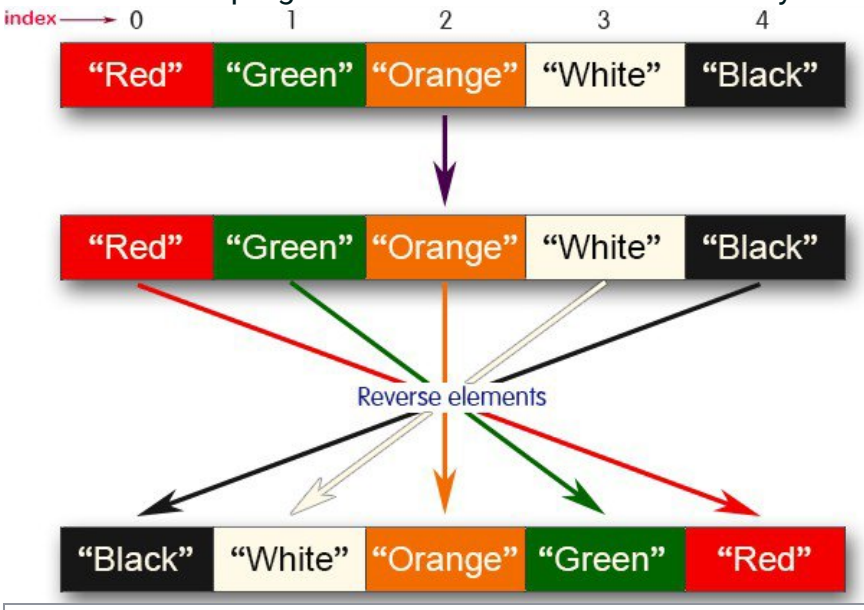
Passed all tests! 1

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
Orang
e
White
Black

Answer: (penalty regime: 0 %)

```
import java.util.*;
public class ReverseArrayList{
    public static void main(String[] args){
        Scanner scanner=new Scanner(System.in);
        ArrayList<String>colorList=new
        ArrayList<>(); int
        n=scanner.nextInt();
        scanner.nextL
        ine(); for(int
        i=0;i<n;i++)
        {
            String
            color=scanner.nextLine();
            colorList.add(color);
        }
        System.out.println("List before
        reversing :");
        System.out.println(colorList);
        Collections.reverse(colorList);
        System.out.println("List after
        reversing :");
        System.out.println(colorList);
    }
}
```

6 17 18 19	

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

Passed all tests! 1

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 Friday, 8 November 2024, 5:24 PM
Completed Friday, 8 November 2024, 5:55 PM

Duration 31 mins 1 sec

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. 1 NULL elements are allowed in HashSet.

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

HashSet also implements Serializable and Cloneable interfaces.

Answer: (penalty regime: 0 %)

Reset answer

```
1  import java.util.HashSet;
2  import java.util.Scanner;
3  class prog {
4    public static void main(String[] args) {
5      Scanner sc= new Scanner(System.in);
6      int n = sc.nextInt();
7      // Create a HashSet object called numbers
8      HashSet<Integer> numbers= new HashSet<>();
9
10     // Add values to the set
11     for(int i=0;i<n;i++)
12     {
13         numbers.add(sc.nextInt());
```

```
14      }
15      int skey=sc.nextInt();
16
17      // Show which numbers between 1 and 10 are in the set
18      if(numbers.contains(skey))
19      {
20          System.out.println(skey+ " was found in the set.");
21      }
22      else {
23          System.out.println(skey + " was not found in the set.");
24       }
25      }
26  }
```

	Test	Input	Expected	Got	
✓	1	5	78 was found in the set.	78 was found in the set.	✓
		9			

		0			
		5 6			
		4 5			
		7 8			
		2 5			
		7 8			
✓	2	3	5 was not found in the set.	5 was not found in the set.	✓
		-1			
		2			
		4			
		5			

Passed all tests! 1

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

Foot

ball

Hoc

key

Cric

ket

Voll

eyba

ll

Bas

ketb

all

7// HashSet 2:

Golf
Cric
ket
Bad
mint
on
Foot
ball
Hoc
key
Voll
eyba
ll
Han
dbal
l

SAMPLE OUTPUT:

Foot
ball
Hoc
key
Cric
ket
Voll
eyba
ll
Bas
ketb
all

Answer: (penalty regime: 0 %)

```
import
java.util.HashSet;
 import
java.util.Scanner;
class prog{
    public static void main(String[] args)
    {
        Scanner sc=new
        Scanner(System.in); int
        n1=sc.nextInt();
        sc.nextLine();
        HashSet<String> set1= new
        HashSet<>(); for (int
        i=0;i<n1;i++)
        {
            set1.add(sc.nextLine());
        }
        int
        n2=sc.nextInt();
        sc.nextLine();
        HashSet<String> set2=new
        HashSet<>(); for(int
        i=0;i<n2;i++)
        {
            set2.add(sc.nextLine());
        }
        set1.retainAll(se
        t2); for(String
        sport:set1)
        {
            System.out.println(sport);
        }
    }
}
```

	Test	Input	Expected	Got	
✓	1	5	Cricket	Cricket	✓
		Football	Hockey	Hockey	

		Hockey	Volleyball	Volleyball	
		Cricket	Football	Football	
		Volleyball			
		Basketball			
		7			
		Golf			
		Cricket			
		Badminton			
		Football			
		Hockey			
		Volleyball			
		Throwball			
✓	2	4	Bus	Bus	✓
		Toy	Car	Car	
		Bus			
		Car			
		Auto			
		3			
		Car			
		Bus			
		Lorry			

Passed all tests! 1

Question 3

Correct

Marked out of 1.00

Java HashMap Methods

containsKey() Indicate if an entry with the specified key exists in the map

containsValue() Indicate if an entry with the specified value exists in the map

putIfAbsent() Write an entry into the map but only if an entry with the same key does not already exist remove() Remove an entry from the map

replace() Write to an entry in the map only if it exists size() Return the number of entries in the map
Your task is to fill the incomplete code to get desired output

Answer: (penalty regime: 0 %)

Reset | answer

```
import java.util.HashMap;
import java.util.Map.Entry;
import java.util.Set;
import java.util.Scanner;
 class prog
{
    public static void main(String[] args)
    {
        //Creating HashMap with default initial capacity and load factor
        HashMap<String, Integer> map = new HashMap<String, Integer>();

        String name;

        int num;

        Scanner sc= new Scanner(System.in);

        int n=sc.nextInt();

         for(int i =0;i<n;i++)

         {

             name=sc.next();

             num= sc.nextInt();

             map.put(name,num);

         }

        //Printing key-value pairs

        Set<Entry<String, Integer>> entrySet = map.entrySet();


```

```
        for (Entry<String, Integer> entry : entrySet)
        {
            System.out.println(entry.getKey()+" : "+entry.getValue());
        }
         System.out.println("");
        //Creating another HashMap
        HashMap<String, Integer> anotherMap = new HashMap<String, Integer>();
        //Inserting key-value pairs to anotherMap using put() method
        anotherMap.put("SIX", 6);
        anotherMap.put("SEVEN", 7);
        //Inserting key-value pairs of map to anotherMap using putAll() method
        anotherMap.putAll(map); // code here
        //Printing key-value pairs of anotherMap
        entrySet = anotherMap.entrySet();
        for (Entry<String, Integer> entry : entrySet)
        {
            System.out.println(entry.getKey()+" : "+entry.getValue());
        }


        //Adds key-value pair 'FIVE-5' only if it is not present in map


        map.putIfAbsent("FIVE", 5);


        //Retrieving a value associated with key 'TWO'

```

```
48
49          int value = map.get("TWO");
50          System.out.println(value);
51
52           //Checking whether key 'ONE' exist in map
```

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

Passed all tests! 1

Lab-11-MCQ

Status Finished

Started Sunday, 10 November 2024, 11:31 AM
Completed Sunday, 10 November 2024, 11:55 AM

Duration 23 mins 50 secs

Question 1

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:

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 %)

```
import java.util.*;
public class SentenceReversal{
    public static void main(String[] args)
    {
        Scanner sc=new
        Scanner(System.in);
        String
        sentence=sc.nextLine();
        int caseOption=sc.nextInt();
        if(caseOption!=0 && caseOption!=1)
        {
            return;
        }
        String result=reverseWordWithCaseOption(sentence,caseOption);
        System.out.println(result);
    }
    public static String reverseWordWithCaseOption(String sentence,int caseOption)
    {
        String[] words=sentence.split(" ");

        StringBuilder result=new StringBuilder();

        for(String word : words)

        {

            StringBuilder reversedWord=new StringBuilder();

            StringBuilder tempWord=new
            StringBuilder(word).reverse();

            if(caseOption==0)

            {

                reversedWord.append(tempWord);
```

```
26
27          }
28          else
29          {
30              for(int i=0;i<word.length();i++)
31              {
32                  char originalChar=word.charAt(i);
33                  char reversedChar=tempWord.charAt(i);
34                  if(Character.isUpperCase(originalChar))
35                  {
36  reversedWord.append(Character.toUpperCase(reversedChar));
37                  }
38                  else if(Character.isLowerCase(originalChar))
39                  {
```

```
40                  reversedWord.append(Character.toLowerCase(reversedChar));
41              }
42              else
43              {
44                  reversedWord.append(reversedChar);
45              }
46          }
47      }
48      result.append(reversedWord).append(" ");
49  }
50  return result.toString().trim();
51  }
52  }
```

	Input	Expected	Got	
✓	Wipro Technologies Bangalore 0	orpiW seigolonhceT erolagnaB	orpiW seigolonhceT erolagnaB	✓
✓	Wipro Technologies, Bangalore 0	orpiW ,seigolonhce T erolagnaB	orpiW ,seigolonhce T erolagnaB	✓

Passed all tests! 1

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 (00000000000000000000000000).

The sequence of 0's in the encoded form are separated by a single 1

which helps to distinguish between 2 letters. Example 1:

input1: 010010001

The decoded string (original

word) will be: ZYX Example 2:

input1: 00001000000000000000001000000000001000000000100000000000001

The decoded string (original word) will be: WIPRO

Note: The decoded string must always be in UPPER case.

For example:

Input	Result
010010001	ZYX
00001000000000000000001000000000001000000000010000000000001	WIPRO

Answer: (penalty regime: 0 %)

```
import java.util.*;
public class BinaryDecoder{
    public static void main(String[] args)
    {
        Scanner sc=new
        Scanner(System.in);
        String
        encoded=sc.nextLine();
        String[] sequences= encoded.split("1");
        StringBuilder decodedWord=new
        StringBuilder(); for(String seq:sequences)
        {
            if(!seq.isEmpty())
            {
                int
                letterPos=seq.length
                (); if(letterPos<=26)
                {
                    char decodedChar=(char)('Z'-(letterPos-1));
                    decodedWord.append(decodedChar);
                }
            }
        }
        System.out.println(decodedWord.toString());
    }
}
```

```
12
13
14
15
16
17
18
19
20
21
22
```

	Input	Expected	Got	
✓	010010001	ZYX	ZYX	✓
✓	00001000000000000000000100000000 00010000000010000000000001	WIPRO	WIPR	✓

Passed all tests! 1

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

Explanat

ion:

‘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 c	8

Answer: (penalty regime: 0 %)

```
import java.io.*;
import java.util.*;
public class commonAlphabets{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        String input1=sc.nextLine().replace(" ,","");
        char[] array1=input1.toCharArray();
        String input2=sc.nextLine().replace(" ","");
        char[] array2=input2.toCharArray();

        int result=calculateSingleDigitSum(array1,array2);

        System.out.println(result);


    }
    private static int calculateSingleDigitSum(char[] input1,char[] input2)

    {
        HashSet<Character> set1=new HashSet<>();
```

```
7
18      for(char c : input1)
19      {
20          set1.add(c);
21      }
22      int sum1=0;
23      for(char c: input2)
24      {
25          if(set1.contains(c))
26          {
27              sum1+=(int) c;
28          }
29      }
30      return getDigitalRoot(sum1);
```

```
31  }
32  private static int
    getDigitalRoot(int sum)
33  {
```

```
        if(sum==0)
        {
            return 0;
        }
        else
        {
            return 1+ ((sum-1)%9);
        }
    }
}
```

	Input	Expected	Got	
✓		8	8	✓

Passed all tests! 1

Lab-12-MCQ

Jump to...

Identify possible words