

Status Finished**Started** Tuesday, 1 October 2024, 8:14 AM**Completed** Tuesday, 1 October 2024, 8:29 AM**Duration** 14 mins 46 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 a=new Scanner(System.in);
5 ↓     int b=a.nextInt();
6 ↓     if(b%2!=0 || b==0){
7 ↓       System.out.print(2);
8     }
9     else if(b%2==0 && b!=0)
10    {
11      System.out.print(1);
12    }
13  }
14 }
15 }
```

	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         Scanner input=new Scanner(System.in);
5         int a=input.nextInt();
6         if(a<0){
7             a=a*-1;
8         }
9         System.out.print(a%10);
10    }
11 }
```

	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         Scanner a=new Scanner(System.in);
5         int b=a.nextInt();
6         int d=a.nextInt();
7         if(b<0){
8             b=b*(-1);
9         }
10        if(d<0){
11            d=d*(-1);
12        }
13        System.out.print(d%10+b%10);
14    }
15 }
```

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

Passed all tests! ✓

◀ Lab-01-MCQ

Jump to...

Is Even? ►

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

Status Finished

Started Tuesday, 1 October 2024, 9:03 AM

Completed Tuesday, 1 October 2024, 8:10 PM

Duration 11 hours 6 mins

Question 1

Correct

Marked out of 5.00

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

Logic to print number in words in Java programming.

Example**Input**

1234

Output

One Two Three Four

Input:

16

Output:

one six

For example:

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

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 public class Main{
3     public static String value(int n){
4         switch(n){
5             case 0:
6                 return "Zero";
7             case 1:
8                 return "One";
9             case 2:
10                return "Two";
11            case 3:
12                return "Three";
13            case 4:
14                return "Four";
15            case 5:
16                return "Five";
17            case 6:
18                return "Six";
19            case 7:
20                return "Seven";
21            case 8:
22                return "Eight";
23            case 9:
24                return "Nine";
25            default:
26                return "invalid";
27        }
28    }
29
30
31    public static void main(String args[]){
32        Scanner input=new Scanner(System.in);
33        int a=input.nextInt();
34        System.out.print(value(a/10)+" "+value(a%10));
35
36    }
37}
38

```

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

Passed all tests! ✓

Question 2

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.util.Scanner;
2 public class Main{
3     public static String g(int n){
4         if(n==1) return "1";
5         String p=g(n-1);
6         return p+" "+n+" "+p;
7     }
8     public static void main(String args[]){
9         Scanner input=new Scanner(System.in);
10        int n=input.nextInt();
11        System.out.println(g(n));
12    }
13}
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! ✓

Question 3

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 Main{
3     public static void main(String args[]){
4         Scanner a=new Scanner(System.in);
5         int b=a.nextInt();
6         int s[]={0,1,1,2,4,7,13,24,44,81,149};
7         if(b>0)s[0]=0;
8         if(b>1)s[1]=1;
9         if(b>2)s[2]=1;
10        if(b>3)s[3]=2;
11        if(b>4)s[4]=4;
12        for(int i=5;i<b;i++){
13            s[i]=s[i-1]+s[i-2]+s[i-3];
14        }
15    }
16    System.out.print(s[b-1]);
17 }
18
19
20 }
21 }
22

```

	Input	Expected	Got	
✓	5	4	4	✓
✓	8	24	24	✓

	Input	Expected	Got	
✓	11	149	149	✓

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 Tuesday, 1 October 2024, 8:10 PM

Completed Tuesday, 1 October 2024, 8:54 PM

Duration 43 mins 36 secs

Question 1

Correct

Marked out of 5.00

Given an array of numbers, you are expected to return the sum of the longest sequence of POSITIVE numbers in the array.

If there are NO positive numbers in the array, you are expected to return -1.

In this question's scope, the number 0 should be considered as positive.

Note: If there are more than one group of elements in the array having the longest sequence of POSITIVE numbers, you are expected to return the total sum of all those POSITIVE numbers (see example 3 below).

input1 represents the number of elements in the array.

input2 represents the array of integers.

Example 1:

input1 = 16

input2 = {-12, -16, 12, 18, 18, 14, -4, -12, -13, 32, 34, -5, 66, 78, 78, -79}

Expected output = 62

Explanation:

The input array contains four sequences of POSITIVE numbers, i.e. "12, 18, 18, 14", "12", "32, 34", and "66, 78, 78". The first sequence "12, 18, 18, 14" is the longest of the four as it contains 4 elements. Therefore, the expected output = sum of the longest sequence of POSITIVE numbers = $12 + 18 + 18 + 14 = 63$.

Example 2:

input1 = 11

input2 = {-22, -24, 16, -1, -17, -19, -37, -25, -19, -93, -61}

Expected output = -1

Explanation:

There are NO positive numbers in the input array. Therefore, the expected output for such cases = -1.

Example 3:

input1 = 16

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

Expected output = 174

Explanation:

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

For example:

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

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 public class l{
3     public static int s(int n3,int[] n2){
4         int m=0;
5         int ms=0;
6         int c=0;
7         int cs=0;
8         for(int n:n2){
9             if(n>=0){

```

```

10         c++;
11         cs+=n;
12     }
13     else{
14         if(c>m){
15             m=c;
16             ms=cs;
17         }
18         else if(c==m){
19             ms+=cs;
20         }
21         c=0;
22         cs=0;
23     }
24 }
25 }
26 if(c>m){
27     m=c;
28     ms=cs;
29 }
30 else if(c==m){
31     ms+=cs;
32 }
33 }
34 return m>0 ? ms:-1;
35 }
36
37 public static void main(String args[]){
38     Scanner a=new Scanner(System.in);
39     int n1=a.nextInt();
40     int b[]={};
41     for(int i=0;i<n1;i++){
42         b[i]=a.nextInt();
43     }
44     int r=s(n1,b);
45     System.out.print(r);
46 }
47 }
48 }
```

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

Passed all tests! ✓

Question 2

Correct

Marked out of 5.00

You are provided with a set of numbers (array of numbers).

You have to generate the sum of specific numbers based on its position in the array set provided to you.

This is explained below:

Example 1:

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

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

Step 1:

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

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

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

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

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

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

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

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

Step 2:

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

{1, 25, 16, 49, 16}

Step 3:

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

Note:

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

Example 2:

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

Step 1:

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

{1, 0, 4, 0, 6}

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

Step 2:

{1, 0, 16, 0, 36}

Step 3:

The final result = 53.

For example:

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

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 public class s{
3     public static int c(int n1,int n2[]){
4         int[] p=new int[n1];
5         for(int i=0;i<n1;i++){

```

```

6     int m=n2[i];
7     p[i]=g(m,i);
8   }
9   int[] s=new int[n1];
10  for(int i=0;i<n1;i++){
11    s[i]=p[i]*p[i];
12  }
13  int sum=0;
14  for(int v:s){
15    sum+=v;
16  }
17  return sum;
18 }
19 public static int g(int num,int pos){
20   int d=(int) ((num/Math.pow(10,pos))%10);
21   return d;
22 }
23
24 public static void main(String args[]){
25   Scanner a=new Scanner(System.in);
26   int n=a.nextInt();
27   int b[]={};
28   for(int i=0;i<n;i++){
29     b[i]=a.nextInt();
30   }
31   int r=c(n,b);
32   System.out.print(r);
33 }
34 }
```

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

Passed all tests! ✓

Question 3

Correct

Marked out of 5.00

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

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

After the operations are done, return the resultant array.

Example 1:

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

`input2 = {1, 5, 6, 9}`

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

Explanation:

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

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

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

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

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

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

Example 2:

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

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

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

Explanation:

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

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

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

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

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

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

Example 3:

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

`input2 = {-9, 9}`

Expected Output = `{-162, 0}`

Explanation:

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

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

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

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

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

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

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

For example:

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

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

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 public class Main{
3     public static int[] p(int n,int[] n2){
4         int m=Integer.MIN_VALUE;
5         for(int num:n2){
6             if(num>m){
7                 m=num;
8             }
9         }
10        int[] s=new int[n];
11        for(int i=0;i<n;i++){
12            s[i]=n2[i]-m;
13        }
14        int[] r=new int[n];
15        for(int i=0;i<n;i++){
16            r[i]=s[i]*m;
17        }
18        return r;
19    }
20    public static void main(String args[]){
21        Scanner a=new Scanner(System.in);
22        int n=a.nextInt();
23        int b[]={};
24        for(int i=0;i<n;i++){
25            b[i]=a.nextInt();
26        }
27        int[] r=p(n,b);
28        for(int v:r){
29            System.out.print(v+" ");
30        }
31    }
32}
33

```

	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 Wednesday, 2 October 2024, 1:31 PM

Completed Wednesday, 2 October 2024, 2:24 PM

Duration 53 mins 19 secs

Question 1

Correct

Marked out of 5.00

Create a Class Mobile with the attributes listed below,

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

Define a Parameterized constructor to initialize the above instance variables.

Define getter and setter methods for the attributes above.

for example : setter method for manufacturer is

```
void setManufacturer(String manufacturer){
    this.manufacturer= manufacturer;
}
```

```
String getManufacturer(){
    return manufacturer;
}
```

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

For example:

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

Answer: (penalty regime: 0 %)

```
1 import java.util.Scanner;
2 public class Main{
3     private String manufacter;
4     private String operating_system;
5     public String color;
6     private int cost;
7     Main(String manufacter,String operating_system,String color,int cost){
8         this.manufacter=manufacter;
9         this.operating_system=operating_system;
10        this.color=color;
11        this.cost=cost;
12    }
13    public void display(){
14        System.out.printf("manufacturer = %s\noperating_system = %s\ncolor = %s\ncost = %d",this.manufacter,this.operating_system,this.color,this.cost);
15    }
16    public static void main(String args[]){
17        Scanner a=new Scanner(System.in);
18        Main b=new Main("Redmi","Andriod","Blue",34000);
19        b.display();
20    }
21 }
22 }
```

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

Passed all tests! ✓

Question 2

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     public Student(){
5         System.out.println("No-arg constructor is invoked");
6         this.name=null;
7         this.rollno=0;
8     }
9     public Student(String name){
10        System.out.println("1 arg constructor is invoked");
11        this.name=name;
12        this.rollno=0;
13    }
14    public Student(String name,int rollno){
15        System.out.println("2 arg constructor is invoked");
16        this.name=name;
17        this.rollno=rollno;
18    }
19    public void display(){
20        System.out.println("Name =" +(name !=null ?name: "null")+", Roll no = "+rollno);
21    }
22    public static void main(String args[]){
23        Student s1=new Student();
24        Student s2=new Student("Rajalakshmi");
25        Student s3=new Student("Lakshmi",101);
26        s1.display();
27        s2.display();
28        s3.display();
29    }
30 }
31 
```

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

Passed all tests! ✓

Question 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 = πr^2

Circumference = $2\pi r$

Input:

2

Output:

Area = 12.57

Circumference = 12.57

For example:

Test	Input	Result
1	4	Area = 50.27 Circumference = 25.13

Answer: (penalty regime: 0 %)

[Reset answer](#)

```

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

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

Passed all tests! ✓

◀ Lab-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 Thursday, 3 October 2024, 8:16 PM

Completed Thursday, 3 October 2024, 9:04 PM

Duration 47 mins 53 secs

Question 1

Correct

Marked out of 5.00

Create a class Mobile with constructor and a method basicMobile().

Create a subclass CameraMobile which extends Mobile class , with constructor and a method newFeature().

Create a subclass AndroidMobile which extends CameraMobile, with constructor and a method androidMobile().

display the details of the Android Mobile class by creating the instance. .

```
class Mobile{
```

```
}
```

```
class CameraMobile extends Mobile {
```

```
}
```

```
class AndroidMobile extends CameraMobile {
```

```
}
```

expected output:

Basic Mobile is Manufactured

Camera Mobile is Manufactured

Android Mobile is Manufactured

Camera Mobile with 5MG px

Touch Screen Mobile is Manufactured

For example:

Result

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

Answer: (penalty regime: 0 %)

```
1. class Mobile{
2.     public Mobile(){
3.         System.out.println("Basic Mobile is Manufactured");
4.     }
5. }
6. class CameraMobile extends Mobile{
7.     public CameraMobile(){
8.         System.out.println("Camera Mobile is Manufactured");
9.     }
10.    public void newFeature(){
11.        System.out.println("Camera Mobile with 5MG px");
12.    }
13. }
14. class AndroidMobile extends CameraMobile{
15.     public AndroidMobile(){
16.         System.out.println("Android Mobile is Manufactured");
17.     }
18. }
19.    public void androidMobile(){
20.        System.out.println("Touch Screen Mobile is Manufactured");
21.    }
22. }
23. class prog{
24.     public static void main(String[] args){
25.         AndroidMobile o=new AndroidMobile();
26.         o.newFeature();
27.         o.androidMobile();
28.     }
29. }
```

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

Passed all tests! ✓

Question 2

Correct

Marked out of 5.00

create a class 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 %)

```
1 class College
2 {
3     protected String collegeName;
4
5     public College(String c) {
6         // initialize the instance variables
7         collegeName=c;
8     }
9
10    public void admitted() {
11        System.out.println("A student admitted in "+collegeName);
12    }
13 }
14 class Student extends College{
15
16     String stu;
17     String dep;
18
19     public Student(String collegeNameo, String studentNameo, String deproto) {
20         // initialize the instance variables
21         super(collegeNameo);
22         stu=studentNameo;
23         dep=deproto;
24     }
25 }
26
27    public String toString(){
28        // return the details of the stu
29        return "CollegeName : "+collegeName+"\nStudentName : "+stu+"\nDepartment : "+dep;
30    }
31 }
32 }
33 class prog {
34     public static void main (String[] args) {
35         Student s1 = new Student("REC", "Venkatesh", "CSE");
36                         // invoke the admitted() method
37         s1.admitted();
38         System.out.println(s1.toString());
39     }
40 }
```

	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 3

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        this.accountNumber=accountNumber;  
11        this.balance=balance;  
12    }  
13  
14  
15  
16  
17     // Method to deposit an amount into the account  
18     public void deposit(double amount) {  
19         // Increase the balance by the deposit amount  
20         balance+=amount;  
21     }  
22  
23  
24     // Method to withdraw an amount from the account  
25     public void withdraw(double amount) {  
26         // Check if the balance is sufficient for the withdrawal  
27         if (balance >= amount) {  
28             // Decrease the balance by the withdrawal amount  
29             balance -= amount;  
30         } else {  
31             // Print a message if the balance is insufficient  
32             System.out.println("Insufficient balance");  
33         }  
34     }  
35  
36     // Method to get the current balance  
37     public double getBalance() {  
38         // Return the current balance  
39         return balance;  
40     }  
41 }  
42 }  
43  
44 class SavingsAccount extends BankAccount {  
45     // Constructor to initialize account number and balance  
46     public SavingsAccount(String accountNumber, double balance) {  
47         // Call the parent class constructor  
48         super(accountNumber,balance);  
49     }  
50  
51     // Override the withdraw method from the parent class
```

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

Passed all tests! ✓

◀ Lab-05-MCQ

Jump to...

Is Palindrome Number? ►

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

Status Finished

Started Thursday, 3 October 2024, 7:20 PM

Completed Thursday, 3 October 2024, 8:09 PM

Duration 48 mins 23 secs

Question 1

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.Scanner;
2 import java.util.Arrays;
3 import java.util.HashSet;
4 import java.util.Set;
5 public class P{
6     public static String cal(String a1, String b){
7         if((a1==null||a1.trim().isEmpty())&&(b==null||b.trim().isEmpty())) return "null";
8         String c=a1+b;
9         Set<Character> u=new HashSet<>();
10        for(char c1:c.toCharArray()){
11            if(Character.isAlphabetic(c1)){
12                u.add(c1);
13            }
14        }
15        char[] ca=new char[u.size()];
16        int i=0;
17        for(char c2:u){
18            ca[i++]=c2;
19        }
20        Arrays.sort(ca);
21        return new StringBuilder(new String(ca)).reverse().toString();
22    }
23    public static void main(String args[]){
24        Scanner a=new Scanner(System.in);
25        String i1=a.nextLine();
26        String i2=a.nextLine();
27        System.out.println(cal(i1,i2));
}

```

```
28 }  
29 }
```

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

Passed all tests! ✓



Question 2

Correct

Marked out of 5.00

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 import java.util.Arrays;
3 import java.lang.String;
4 public class Prog{
5     public static String pw(String input,int index){
6         String[] words=input.split(" ");
7         int f=(index/10)-1;
8         int ss=(index%10)-1;
9         String fp=pws(words[f]);
10        String sn=pws(words[ss]);

```

```

11     return fp+ " " +sp;
12 }
13 public static String pws(String o){
14     int l=o.length();
15     int mid=l/2;
16     String l1,f1;
17 if(l%2==0){
18     f1=o.substring(0,mid);
19     f1=new StringBuilder(f1).reverse().toString();
20     l1=o.substring(mid);
21     return f1+l1;
22 }
23 else{
24     f1=o.substring(0,mid+1);
25     f1=new StringBuilder(f1).reverse().toString();
26     l1=o.substring(mid);
27     return f1+l1;
28 }
29 }
30 public static void main(String[] args){
31     Scanner a=new Scanner(System.in);
32     String s=a.nextLine();
33     int n=a.nextInt();
34     String r=pw(s,n);
35     System.out.println(r);
36 }
37 }
```

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

Passed all tests! ✓

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 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 public class Main{
3     public static void main(String args[]){
4         Scanner a=new Scanner(System.in);
5         String input=a.nextLine();
6         String[] w=input.split(":");
7         StringBuilder r=new StringBuilder();
8         for(String word:w){
9             char c1=word.charAt(0);
10            char c2=word.charAt(1);
11            if(c1==c2){
12                r.append(Character.toUpperCase(c1));
13            }
14            else{
15                int p=c1-'a'+1;
16                int p1=c2-'a'+1;
17                int d=Math.abs(p-p1);
18                char newc=(char)('a'+d-1);
19                r.append(Character.toUpperCase(newc));
20            }
21        }
22        System.out.print(r.toString());
23    }
24 }
```

	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 Sunday, 6 October 2024, 9:23 AM

Completed Sunday, 6 October 2024, 10:36 AM

Duration 1 hour 12 mins

Question 1

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

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

Passed all tests! ✓

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     void play();
4 }
5 class Football implements Playable{
6     private String name;
7     public Football(String name){
8         this.name=name;
9     }
10    public void play(){
11        System.out.println(name+ " is Playing football");
12    }
13 }
14 class Volleyball implements Playable{
15     private String name;
16     public Volleyball(String name){
17         this.name=name;
18     }
19     public void play(){
20         System.out.println(name + " is Playing volleyball");
21     }
22 }
23 class Basketball implements Playable{
24     private String name;
25     public Basketball(String name){
26         this.name=name;
27     }
28     public void play(){
29         System.out.println(name + " is Playing basketball");
30     }
31 }
32 public class Main{
33     public static void main(String args[]){
34         Scanner a=new Scanner(System.in);
35         Playable f=new Football(a.nextLine());
36         Playable v=new Volleyball(a.nextLine());
37         Playable b=new Basketball(a.nextLine());
38         f.play();
39         v.play();
```

```
40     b.play();  
41 }  
42 }  
43 }
```

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

Passed all tests! ✓

Question 3

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 v(int points);
9 }
10 class College implements Football {
11     private String homeTeam;
12     private String visitingTeam;
13     private int homeScore;
14     private int visitingScore;
15     public void setHomeTeam(String name){
16         this.homeTeam=name;
17     }
18     public void setVisitingTeam(String name){
19         this.visitingTeam=name;
20     }
21     public void homeTeamScored(int points){
22         this.homeScore+=points;
23         System.out.println(homeTeam+" "+points+" scored");
24     }
25     public void v(int points){
26         this.visitingScore+=points;
27         System.out.println(visitingTeam+" "+points+" scored");
28     }
29     public void winningTeam(){
30         if(homeScore>visitingScore)
31             System.out.println(homeTeam + " is the winner!");
32         else if(homeScore<visitingScore){
33             System.out.println(visitingTeam + " is the winner!");
34         }
35
36     else{
37         System.out.println("It's a tie match.");
38     }
39 }
40 }
```

```

42 class prog{
43     public static void main(String[] args){
44         String hname;
45         Scanner sc= new Scanner(System.in);
46         hname=sc.nextLine();
47         String vteam=sc.next();
48         int hpoints=sc.nextInt();
49         int vpoints=sc.nextInt();
50         College G= new College();
51         G.setHomeTeam(hname);
52         G.setVisitingTeam(vteam);

```

	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! ✓

◀ Lab-07-MCQ

Jump to...

Generate series and find Nth element ►

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

Status Finished

Started Sunday, 6 October 2024, 5:54 PM

Completed Sunday, 6 October 2024, 6:08 PM

Duration 13 mins 56 secs

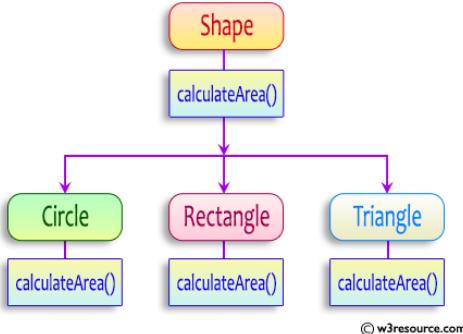
Question 1

Correct

Marked out of 5.00

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

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



```

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 interface Playable {
3     void play();
4 }
5 class Football implements Playable {
6     String name;
7     public Football(String name) {
8         this.name = name;
9     }
10    @Override
11    public void play() {
12        System.out.println(name + " is playing football");
13    }
14 }
15 class Volleyball implements Playable {
16     String name;
17     public Volleyball(String name) {
18         this.name = name;
19     }
20 }

```

```

19     }
20     @Override
21     public void play() {
22         System.out.println(name + " is playing volleyball");
23     }
24 }
25 abstract class Shape {
26     public abstract double calculateArea();
27 }
28 class Rectangle extends Shape {
29     double length;
30     double breadth;
31     public Rectangle(double length, double breadth) {
32         this.length = length;
33         this.breadth = breadth;
34     }
35     @Override
36     public double calculateArea() {
37         return length * breadth;
38     }
39 }
40 class Triangle extends Shape {
41     double base;
42     double height;
43     public Triangle(double base, double height) {
44         this.base = base;
45         this.height = height;
46     }
47     @Override
48     public double calculateArea() {
49         return 0.5 * base * height;
50     }
51 }
52 class Circle extends Shape {

```

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

Passed all tests! ✓

Question 2

Correct

Marked out of 5.00

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

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

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

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

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

input2: String array.

Example 1:

input1: 3

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

output: oreoapple

Example 2:

input1: 2

input2: {"Mango", "banana"}

output: no matches found

Explanation:

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

Hence the output is no matches found.

Example 3:

input1: 3

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

output: ateace

For example:

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

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2
3 abstract class Operations {
4     public boolean isVowel(char c) {
5         return c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u';
6     }
7
8     public boolean hasVowelEnds(String s) {
9         return (isVowel(s.charAt(0)) && isVowel(s.charAt(s.length() - 1)));
10    }
11 }
12
13 class Concatenate extends Operations {
14     public String ans = "";
15
16     Concatenate(String[] words) {
17         for (String s : words) {
18             if (hasVowelEnds(s.toLowerCase())) {
19                 ans += s.toLowerCase();
20             }
21         }
22     }
23
24     public String getAnswer() {
25         if (ans.isEmpty()) return "no matches found";
26         return ans.trim();
27     }

```

```
27     }
28 }
29
30 class prog {
31     public static void main(String args[]) {
32         Scanner scan = new Scanner(System.in);
33         int n = scan.nextInt();
34         scan.nextLine();
35         String[] words = scan.nextLine().split(" ");
36         Concatenate cn = new Concatenate(words);
37         System.out.println(cn.getAnswer());
38         scan.close();
39     }
40 }
41
```

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

Passed all tests! ✓

Question 3

Correct

Marked out of 5.00

1. Final Variable:

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

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

2. Final Method:

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

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

3. Final Class:

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

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

you should delete any piece of code.

For example:

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

Answer: (penalty regime: 0 %)

[Reset answer](#)

```

1  class FinalExample {
2
3
4     int maxSpeed = 120;
5
6     // Final method
7     public final void displayMaxSpeed() {
8         System.out.println("The maximum speed is: " + maxSpeed + " km/h");
9     }
10}
11
12 class SubClass extends FinalExample {
13
14     /* public void displayMaxSpeed() {
15         System.out.println("Cannot override a final method");
16     }*/
17
18     // You can create new methods here
19     public void showDetails() {
20         System.out.println("This is a subclass of FinalExample.");
21     }
22 }
23
24 class prog {
25     public static void main(String[] args) {
26         FinalExample obj = new FinalExample();
27         obj.displayMaxSpeed();
28
29         SubClass subObj = new SubClass();
30         subObj.showDetails();
31     }
32 }
33

```

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

Passed all tests! ✓

[◀ Lab-08-MCQ](#)

Jump to...

[FindStringCode ►](#)

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

Status Finished

Started Thursday, 17 October 2024, 8:20 PM

Completed Thursday, 17 October 2024, 8:51 PM

Duration 31 mins 1 sec

Question 1

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

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

Passed all tests! ✓

Question 2

Correct

Marked out of 5.00

In the following program, an array of integer data is to be initialized.

During the initialization, if a user enters a value other than an integer, it will throw an InputMismatchException exception.

On the occurrence of such an exception, your program should print "You entered bad data."

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

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

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

Sample Input:

```
3  
5 2 1
```

Sample Output:

```
8
```

Sample Input:

```
2  
1 g
```

Sample Output:

```
You entered bad data.
```

For example:

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

Answer: (penalty regime: 0 %)

```
1 import java.util.Scanner;  
2 import java.util.InputMismatchException;  
3 class prog {  
4     public static void main(String[] args) {  
5         Scanner sc = new Scanner(System.in);  
6         int length = sc.nextInt();  
7         // create an array to save user input  
8         int[] name = new int[length];  
9         int sum=0;//save the total sum of the array.  
10  
11     /* Define try-catch block to save user input in the array "name"  
12     If there is an exception then catch the exception otherwise print  
13     the total sum of the array. */  
14         try  
15         {  
16             for(int i=0;i<length;i++){  
17                 name[i]=sc.nextInt();  
18                 sum+=name[i];  
19             }  
20             System.out.print(sum);  
21  
22         }  
23         catch(InputMismatchException e)  
24         {  
25             System.out.print("You entered bad data.");  
26         }  
27     }  
28  
29  
30  
31  
32  
33
```

34	}
35	}

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

Passed all tests! ✓

//

Question 3

Correct

Marked out of 5.00

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

Sample input and Output:

```
82 is even.  
Error: 37 is odd.
```

Fill the preloaded answer to get the expected output.

For example:

Result
82 is even. Error: 37 is odd.

Answer: (penalty regime: 0 %)

[Reset answer](#)

```

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

```

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

Passed all tests! ✓

[◀ Lab-09-MCQ](#)

Jump to...

[The “Nambiar Number” Generator ►](#)

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

Status Finished

Started Saturday, 2 November 2024, 5:36 PM

Completed Saturday, 2 November 2024, 6:00 PM

Duration 24 mins 34 secs

Question 1

Correct

Marked out of 1.00

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

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

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

Approach:

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

Answer: (penalty regime: 0 %)

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

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

Passed all tests! ✓

Question 2

Correct

Marked out of 1.00

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

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

The above methods are used for the below Java program.

Answer: (penalty regime: 0 %)

[Reset answer](#)

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

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

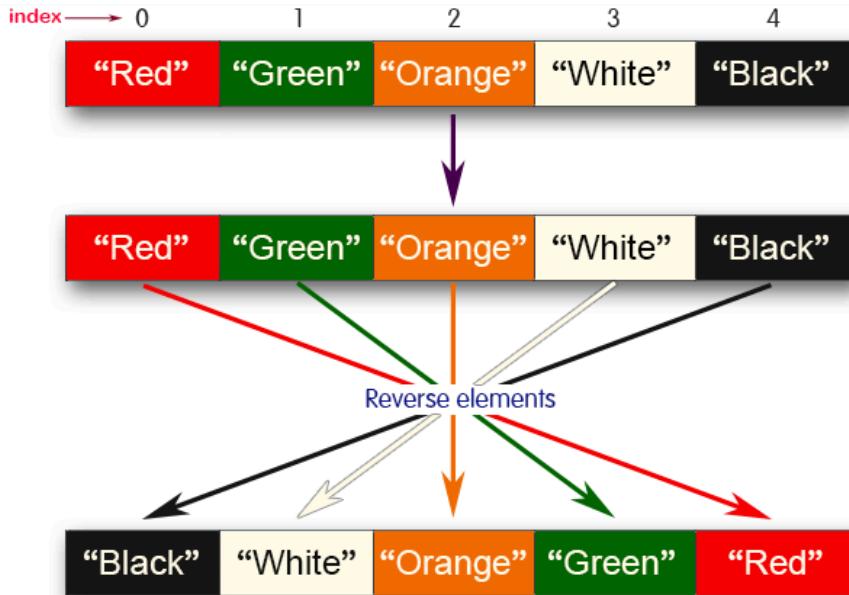
Passed all tests! ✓

Question 3

Correct

Marked out of 1.00

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



Sample input and Output:

Red
Green
Orange
White
Black

Sample output

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

Answer: (penalty regime: 0 %)

```

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

```

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

Passed all tests! ✓

◀ Lab-10-MCQ

Jump to...

Lab-11-MCQ ►

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

Status Finished

Started Sunday, 10 November 2024, 9:26 PM

Completed Sunday, 10 November 2024, 9:47 PM

Duration 20 mins 49 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 public 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         // Add values to the set
10        for(int i=0;i<n;i++)
11            numbers.add(sc.nextInt());
12
13        int skey=sc.nextInt();
14
15        // Show which numbers between 1 and 10 are in the set
16
17        if(numbers.contains(skey)){
18            System.out.println(skey+ " was found in the set.");
19        }
20        else {
21            System.out.println(skey + " was not found in the set.");
22        }
23    }
24 }
25 }
```

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

Passed all tests! ✓

Question 2

Correct

Marked out of 1.00

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

Sample Input and Output:

5
Football
Hockey
Cricket
Volleyball
Basketball

7 // HashSet 2:
Golf
Cricket
Badminton
Football
Hockey
Volleyball
Handball

SAMPLE OUTPUT:

Football
Hockey
Cricket
Volleyball
Basketball

Answer: (penalty regime: 0 %)

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

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

Passed all tests! ✓

Question 3

Correct

Marked out of 1.00

Java HashMap Methods

[containsKey\(\)](#). Indicate if an entry with the specified key exists in the 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](#)

```

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 Prog {
7     public static void main(String[] args) {
8         // Creating HashMap with default initial capacity and load factor
9         HashMap<String, Integer> map = new HashMap<String, Integer>();
10
11     String name;
12     int num;
13     Scanner sc = new Scanner(System.in);
14     int n = sc.nextInt();
15
16     // Taking input and inserting 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
24     Set<Entry<String, Integer>> entrySet = map.entrySet();
25     for (Entry<String, Integer> entry : entrySet) {
26         System.out.println(entry.getKey() + " : " + entry.getValue());
27     }
28
29     System.out.println("-----");
30
31     // Creating another HashMap
32     HashMap<String, Integer> anotherMap = new HashMap<String, Integer>();
33
34     // Inserting key-value pairs to anotherMap using put() method
35     anotherMap.put("SIX", 6);
36     anotherMap.put("SEVEN", 7);
37
38     // Inserting key-value pairs of map to anotherMap using putAll() method
39     anotherMap.putAll(map);
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     if (value != null) {

```

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

Passed all tests! ✓

◀ Lab-11-MCQ

Jump to...

TreeSet example ►

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

Status Finished

Started Tuesday, 12 November 2024, 7:05 PM

Completed Tuesday, 12 November 2024, 7:33 PM

Duration 27 mins 59 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 %)

```

1 import java.util.Scanner;
2
3 public class ReverseWords {
4     public static void main(String[] args) {
5
6         Scanner scanner = new Scanner(System.in);
7
8
9         String sentence = scanner.nextLine();
10
11
12
13         int caseOption = scanner.nextInt();
14
15
16         String reversedSentence = reverseWords(sentence, caseOption);
17
18
19         System.out.println(reversedSentence);
20
21
22         scanner.close();
23     }
24 }
```

```

25
26     public static String reverseWords(String sentence, int case_option) {
27
28         String[] words = sentence.split(" ");
29         StringBuilder result = new StringBuilder();
30
31         for (String word : words) {
32             String reversedWord = reverseWord(word, case_option);
33             result.append(reversedWord).append(" ");
34         }
35
36
37         return result.toString().trim();
38     }
39
40     private static String reverseWord(String word, int case_option) {
41
42         char[] chars = word.toCharArray();
43         char[] reversedChars = new char[chars.length];
44
45
46         for (int i = 0; i < chars.length; i++) {
47             reversedChars[i] = chars[chars.length - 1 - i];
48         }
49         if (case_option == 1) {
50             for (int i = 0; i < chars.length; i++) {
51                 // Check if the original character was uppercase
52                 if (Character.isUpperCase(chars[i])) {

```

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

Passed all tests! ✓

//

Question 2

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 import java.util.Scanner;
3
4 public class CommonAlphabets {
5     public static void main(String[] args) {
6
7         Scanner scanner = new Scanner(System.in);
8
9
10
11        String input1Str = scanner.nextLine();
12        char[] input1 = input1Str.replace(" ", "").toCharArray(); // Convert the string to char array
13
14
15        String input2Str = scanner.nextLine();
16        char[] input2 = input2Str.replace(" ", "").toCharArray(); // Convert the string to char array
17
18
19        int result = getSingleDigitSum(input1, input2);
20        System.out.println(result);
21
22
23        scanner.close();
24    }
25
26
27    public static int getSingleDigitSum(char[] input1, char[] input2) {
28
29        HashSet<Character> commonChars = new HashSet<>();
30
31
32        HashSet<Character> input1Set = new HashSet<>();
33        for (char c : input1) {
34            input1Set.add(c);
35        }
36    }

```

```
~ 37
38     for (char c : input2) {
39         if (input1Set.contains(c)) {
40             commonChars.add(c);
41         }
42     }
43
44
45     int sum1 = 0;
46     for (char c : commonChars) {
47         sum1 += (int) c;
48     }
49
50
51     return getSingleDigit(sum1);
52 }
```

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

Passed all tests! ✓

//

Question 3

Correct

Marked out of 5.00

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

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

Each alphabet is represented by a sequence of 0s.

This is as mentioned below:

Z:0

Y: 00

X: 000

w : 0000

V: 00000

U : 000000

T : 0000000

and so on upto A having 26 0's (00000000000000000000000000).

The sequencer

Example 1:

input1: 010010001

The decoder

Example 2:

Input 1: 000010000000000010000000

The decoded string (original word) will be: WIPRO

Answer: (penalty regime: 0 %)

```
1 import java.util.Scanner;
2
3 public class DecodeString {
4     public static void main(String[] args) {
5
6         Scanner scanner = new Scanner(System.in);
7         String input = scanner.nextLine();
8
9
10        String decodedWord = decode(input);
11
12
13        System.out.println(decodedWord);
14
15
16        scanner.close();
17    }
18
19    public static String decode(String input) {
20
21        String[] segments = input.split("1");
22        StringBuilder decodedWord = new StringBuilder();
23
24
25        for (String segment : segments) {
26            if (segment.isEmpty()) continue;
27
28            int zeroCount = segment.length();
29            char decodedChar = (char) ('A' + (26 - zeroCount));
30            decodedWord.append(decodedChar);
31
32        }
33
34        return decodedWord.toString();
35    }
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
50
51
52
53
54
55
56
57
58
59
5
60
61
62
63
64
65
66
67
68
69
6
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
10
110
111
112
113
114
115
116
117
118
119
11
120
121
122
123
124
125
126
127
128
129
12
130
131
132
133
134
135
136
137
138
13
139
140
141
142
143
144
145
146
147
14
148
149
150
151
152
153
154
155
156
157
15
158
159
160
161
162
163
164
165
166
16
167
168
169
170
171
172
173
174
175
176
17
177
178
179
180
181
182
183
184
185
186
18
187
188
189
190
191
192
193
194
195
196
19
197
198
199
19
199
200
201
202
203
204
205
206
207
208
209
20
210
211
212
213
214
215
216
217
218
219
21
220
221
222
223
224
225
226
227
228
229
22
230
231
232
233
234
235
236
237
238
239
23
240
241
242
243
244
245
246
247
248
249
24
250
251
252
253
254
255
256
257
258
259
25
260
261
262
263
264
265
266
267
268
269
26
270
271
272
273
274
275
276
277
278
279
27
280
281
282
283
284
285
286
287
288
289
28
290
291
292
293
294
295
296
297
298
299
29
299
300
301
302
303
304
305
306
307
308
309
30
310
311
312
313
314
315
316
317
318
319
31
320
321
322
323
324
325
326
327
328
329
32
330
331
332
333
334
335
336
337
338
339
33
340
341
342
343
344
345
346
347
348
349
34
350
351
352
353
354
355
356
357
358
359
35
360
361
362
363
364
365
366
367
368
369
36
370
371
372
373
374
375
376
377
378
379
37
380
381
382
383
384
385
386
387
388
389
38
390
391
392
393
394
395
396
397
398
399
39
399
400
401
402
403
404
405
406
407
408
409
40
410
411
412
413
414
415
416
417
418
419
41
420
421
422
423
424
425
426
427
428
429
42
430
431
432
433
434
435
436
437
438
439
43
440
441
442
443
444
445
446
447
448
449
44
450
451
452
453
454
455
456
457
458
459
45
460
461
462
463
464
465
466
467
468
469
46
470
471
472
473
474
475
476
477
478
479
47
480
481
482
483
484
485
486
487
488
489
48
490
491
492
493
494
495
496
497
498
499
49
499
500
501
502
503
504
505
506
507
508
509
50
510
511
512
513
514
515
516
517
518
519
51
520
521
522
523
524
525
526
527
528
529
52
530
531
532
533
534
535
536
537
538
539
53
540
541
542
543
544
545
546
547
548
549
54
550
551
552
553
554
555
556
557
558
559
55
560
561
562
563
564
565
566
567
568
569
56
570
571
572
573
574
575
576
577
578
579
57
580
581
582
583
584
585
586
587
588
589
58
590
591
592
593
594
595
596
597
598
599
59
599
600
601
602
603
604
605
606
607
608
609
60
610
611
612
613
614
615
616
617
618
619
61
620
621
622
623
624
625
626
627
628
629
62
630
631
632
633
634
635
636
637
638
639
63
640
641
642
643
644
645
646
647
648
649
64
650
651
652
653
654
655
656
657
658
659
65
660
661
662
663
664
665
666
667
668
669
66
670
671
672
673
674
675
676
677
678
679
67
680
681
682
683
684
685
686
687
688
689
68
690
691
692
693
694
695
696
697
698
699
69
699
700
701
702
703
704
705
706
707
708
709
70
710
711
712
713
714
715
716
717
718
719
71
720
721
722
723
724
725
726
727
728
729
72
730
731
732
733
734
735
736
737
738
739
73
740
741
742
743
744
745
746
747
748
749
74
750
751
752
753
754
755
756
757
758
759
75
760
761
762
763
764
765
766
767
768
769
76
770
771
772
773
774
775
776
777
778
779
77
780
781
782
783
784
785
786
787
788
789
78
790
791
792
793
794
795
796
797
798
799
79
799
800
801
802
803
804
805
806
807
808
809
80
810
811
812
813
814
815
816
817
818
819
81
820
821
822
823
824
825
826
827
828
829
82
830
831
832
833
834
835
836
837
838
839
83
840
841
842
843
844
845
846
847
848
849
84
850
851
852
853
854
855
856
857
858
859
85
860
861
862
863
864
865
866
867
868
869
86
870
871
872
873
874
875
876
877
878
879
87
880
881
882
883
884
885
886
887
888
889
88
890
891
892
893
894
895
896
897
898
899
89
899
900
901
902
903
904
905
906
907
908
909
90
910
911
912
913
914
915
916
917
918
919
91
920
921
922
923
924
925
926
927
928
929
92
930
931
932
933
934
935
936
937
938
939
93
940
941
942
943
944
945
946
947
948
949
94
950
951
952
953
954
955
956
957
958
959
95
960
961
962
963
964
965
966
967
968
969
96
970
971
972
973
974
975
976
977
978
979
97
980
981
982
983
984
985
986
987
988
989
98
990
991
992
993
994
995
996
997
998
999
99
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
100
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
101
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
102
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
103
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
104
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
105
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
106
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
107
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
108
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
109
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
110
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
111
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
112
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
113
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
114
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
115
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
116
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
117
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
118
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
119
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
120
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
121
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
122
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
123
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
124
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
125
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
126
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
127
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
128
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
129
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
130
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
131
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
132
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
133
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
134
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
135
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
136
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
137
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
138
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
139
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
140
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
141
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
142
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
143
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
144
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
145
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
146
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
147
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
148
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
149
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
150
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
151
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
152
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
153
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
154
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
155
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
156
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
157
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
158
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
159
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
160
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
161
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
162
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
163
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
164
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
165
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
166
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
167
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
168
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
169
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
170
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
171
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
172
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
173
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
174
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
175
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
176
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
177
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
178
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
179
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
180
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
181
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
182
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
183
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
184
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
185
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
186
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
187
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
188
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
189
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
190
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
191
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
192
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
193
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
194
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
195
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
196
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
197
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
198
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
199
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
200
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
201
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
202
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
203
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
204
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
205
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
206
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
207
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
208
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
209
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
210
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
211
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
212
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
213
2139
2140
2141
2142
2143

```

35 | }
36

Passed all tests! ✓

◀ Lab-12-MCQ

Jump to...

Identify possible words ►