OBJECT ORIENTED PROGRAMMING USING JAVA

LAB MANUAL

Lab-01-Java Architecture, Language Basics

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 %)

```
import java.util.*;
class prog{
    public static void main(String args[]){
        Scanner s=new Scanner(System.in);
        int n=s.nextInt();

        if(n%2==0)
            System.out.println("1");
        else
            System.out.println("2");
        }
    }
```

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

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

The last digit should be returned as a positive number.

For example,

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

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

For example:

Input	Result
197	7
-197	7

Answer: (penalty regime: 0 %)

```
import java.util.*;
class prog{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        int n=s.nextInt();
        System.out.println(Math.abs(n%10));
    }
}
```

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

Passed all tests! ✓

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 %)

```
import java.util.*;
class prog{
    public static void main(String args[]){
        Scanner s=new Scanner(System.in);
        int n1=s.nextInt();
        int n2=s.nextInt();
        System.out.println(Math.abs(n1%10) + Math.abs(n2%10));}}
```

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

Passed all tests! ✓

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

Logic to print number in words in Java programming.

Example

Input

1234

Output

One Two Three Four

Input:

16

Output:

one six

For example:

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

Answer: (penalty regime: 0 %)

```
import java.util.*;
class prog{
    public static void main(String args[]){
        Scanner s=new Scanner(System.in);
        String n=s.nextLine();
        for(int i=0;i<n.length();i++){
            char dig=n.charAt(i);
            switch(dig){
                case '1':
                    System.out.print("One ");
                    break;
                case '3':
                    System.out.print("Three ");
                    break;
                case '4':
                    System.out.print("Four ");
                    break;
                case '5':
                    System.out.print("Five ");
                    break;
                case '7':
                    System.out.print("Seven ");
                    break;
                case '8':
                    System.out.print("Eight ");
                    break;
            }
        }
    }
}
```

Lab-02-Flow Control Statements

	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! ✓

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

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

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

Input Format

Input containing an integer and a boolean value.

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

You have to print the time you should get up.

Example Input:

1 false

Output:

6:00

Example Input:

5 false

Output:

5:00

Example Input:

1 true

Output:

9:00

For example:

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

Answer: (penalty regime: 0 %)

```
import java.util.*;
class prog{
    public static void main(String args[]){
        Scanner s=new Scanner(System.in);
        int day=s.nextInt();
        Boolean vacation =s.nextBoolean();
        if(vacation == false){
            if(day==1 || day==7)
                System.out.println("6:00");
            else
                System.out.println("5:00");
        }
        else{
            if(day==1||day==7)
                System.out.println("9:00");
            else
                System.out.println("7:00");

        }


    }
}
```

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

Passed all tests! ✓

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

The movie's success formula depends on 2 parameters:

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

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

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

Write a program that takes 2 integers:

the first integer is the acting power

second integer is the critic's rating.

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

Example input:

9 5

Output:

Yes

Example input:

1 9

Output:

No

Example input:

6 4

Output:

Maybe

For example:

Input	Result
9 5	Yes
1 9	No
6 4	Maybe

```
import java.util.*;
class prog{
    public static void main(String[] args){
        Scanner s=new Scanner(System.in);
        int actp=s.nextInt();
        int cri=s.nextInt();

        if(actp<2 || cri>=9){
            System.out.println("No");
        }
        else if(actp>=8  || cri >=5)
            System.out.println("Yes");
        else
            System.out.println("Maybe");

        /*if(actp>8)
            System.out.println("Yes");
        else if(critrate<2)
            System.out.println("No");
        else
            System.out.println("Maybe");
        */
        }
    }
```

	Input	Expected	Got	
✓	9 5	Yes	Yes	✓
✓	1 9	No	No	✓
✓	6 4	Maybe	Maybe	✓

Passed all tests! ✓

Lab-03-Arrays

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

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

This is explained below:

Example 1:

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

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

Step 1:

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

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

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

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

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

4^{th} 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 %)

```
import java.util.*;
class prog{
    public static void main(String args[]){
        Scanner in=new Scanner(System.in);
        int n=in.nextInt();
        int s=0;
        int arr[]=new int[n];
        int res[]=new int[n];
        for(int i=0;i<n;i++)
        arr[i]=in.nextInt();

        for(int i=0;i<n;i++){
            if(arr[i]<Math.pow(10,i))
            {
                res[i]=0;
                s+=res[i]*res[i];
            }

            else{
                res[i]=(arr[i]/(int)Math.pow(10,i))%10;
                s+=res[i]*res[i];
            }
        }
        System.out.println(s);
    }}
```

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

Passed all tests! ✓

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 %)

```java
import java.util.*;
class prog{
    public static void main(String args[]){
        Scanner in=new Scanner(System.in);
        int n=in.nextInt();
        int arr[]=new int[n];
        for(int i=0;i<n;i++){
            arr[i]=in.nextInt();

        }
        System.out.print(res(n,arr));
    }
    public static int res(int n,int arr[]){
        int ms=0,ml=0,cs=0,cl=0;
        boolean r=false;
        for(int i=0;i<n;i++){
        if(arr[i]>=0)
        {
            cl++;
            cs+=arr[i];
            r=true;
        }
        else
        {
            if(cl>ml){
                ml=cl;
                ms=cs;
            }
            else if(cl==ml){
                ms+=cs;
            }
            cl=0;
            cs=0;
        }
        }

        if(cl>ml){
            ms=cs;
        }
        else if(cl==ml){
            ms+=cs;
        }
        return r ? ms : -1;
    }
}
```

	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! ✓

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

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

After the operations are done, return the resultant array.

Example 1:

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

input2 = {1, 5, 6, 9}

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

Explanation:

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

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

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

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

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

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

Example 2:

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

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

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

Explanation:

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

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

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

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

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

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

Example 3:

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

input2 = {-9, 9}

Expected Output = {-162, 0}

Explanation:

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

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

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

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

{(-18 x 9), (0 x 9)} = {-162, 0}

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

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

Answer: (penalty regime: 0 %)

```
import java.util.*;
class prog{

        private static int findmax(int numb[]){
            int mx=numb[0];
            for(int num : numb){
            if(mx<num)
            {
                mx=num;
            }
        }
        return mx;
        }
        public static void submax(int[] numb,int mx){
            for(int i=0;i<numb.length;i++)
                numb[i]-=mx;
        }
        public static void mulmax(int num[],int mx){
            for(int i=0;i<num.length;i++)
                num[i]*=mx;
        }
        public static void main(String args[]){
            Scanner in=new Scanner(System.in);
            int n=in.nextInt();
            int ar[]=new int[n];
            for(int i=0;i<n;i++){
                ar[i]=in.nextInt();
            }
            int maxnum=findmax(ar);
            submax(ar,maxnum);
            mulmax(ar,maxnum);
            for(int num :ar){
                System.out.print(num+" ");
            }
            }
        }

```

	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-04-Classes and Objects

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 %)

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

```java
    @Override
    public String toString() {
        return "manufacturer = " + manufacturer + "\n" +
                "operating_system = " + operating_system + "\n" +
                "color = " + color + "\n" +
                "cost = " + cost;
    }

    public static void main(String[] args) {

        prog mobile = new prog("Redmi", "Andriod", "Blue", 34000);

        System.out.println(mobile.toString());

    }
}
```

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

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

```java
import java.util.*;
class Circle
{
    private double radius;
    public Circle(double radius){
        this.radius= radius;
    }
    public void setRadius(double radius){
        this.radius=radius;


    }
    public double getRadius()    {
        return radius;


    }
    public double calculateArea()  { // complete the below statement
       return Math.PI*radius*radius;

    }
    public double calculateCircumference()    {
        // complete the statement
       return  2*Math.PI*radius;
    }
}
class prog{
    public static void main(String[] args)  {
        int r;
        Scanner sc= new Scanner(System.in);
        r=sc.nextInt();
        Circle c= new Circle(r);
        System.out.println("Area = "+String.format("%.2f", c.calculateArea()));
        System.out.println("Circumference = "+String.format("%.2f", c.calculateCircumference()));


    }
}
```

	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	✓

Lab-05-Inheritance

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

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

This is explained below:

Example 1:

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

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

Step 1:

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

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

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

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

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

4^{th} 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

```java
import java.util.*;
class prog{
    public static void main(String args[]){
        Scanner in=new Scanner(System.in);
        int n=in.nextInt();
        int s=0;
        int arr[]=new int[n];
        int res[]=new int[n];
        for(int i=0;i<n;i++)
        arr[i]=in.nextInt();

        for(int i=0;i<n;i++){
            if(arr[i]<Math.pow(10,i))
            {
                res[i]=0;
                s+=res[i]*res[i];
            }

            else{
                res[i]=(arr[i]/(int)Math.pow(10,i))%10;
                s+=res[i]*res[i];
            }
        }
        System.out.println(s);
    }}
```

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

Passed all tests! ✓

```java
import java.util.*;
class prog{
    public static void main(String args[]){
        Scanner in=new Scanner(System.in);
        int n=in.nextInt();
        int arr[]=new int[n];
        for(int i=0;i<n;i++){
            arr[i]=in.nextInt();

        }
        System.out.print(res(n,arr));
    }
    public static int res(int n,int arr[]){
        int ms=0,ml=0,cs=0,cl=0;
        boolean r=false;
        for(int i=0;i<n;i++){
        if(arr[i]>=0)
        {
            cl++;
            cs+=arr[i];
            r=true;
        }
        else
        {
            if(cl>ml){
                ml=cl;
                ms=cs;
            }
            else if(cl==ml){
                ms+=cs;
            }
            cl=0;
            cs=0;
        }
        }

            if(cl>ml){
                ms=cs;
            }
            else if(cl==ml){
                ms+=cs;
            }
            return r ? ms : -1;
    }
}
```

	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! ✓

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

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

After the operations are done, return the resultant array.

Example 1:

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

input2 = {1, 5, 6, 9}

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

Explanation:

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

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

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

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

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

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

Example 2:

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

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

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

Explanation:

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

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

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

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

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

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

Example 3:

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

input2 = {-9, 9}

Expected Output = {-162, 0}

Explanation:

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

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

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

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

{(-18 x 9), (0 x 9)} = {-162, 0}

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

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

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

Answer: (penalty regime: 0 %)

```
import java.util.*;
class prog{

        private static int findmax(int numb[]){
            int mx=numb[0];
            for(int num : numb){
            if(mx<num)
            {
                mx=num;
            }
        }
        return mx;
        }
        public static void submax(int[] numb,int mx){
            for(int i=0;i<numb.length;i++)
                numb[i]-=mx;
        }
        public static void mulmax(int num[],int mx){
            for(int i=0;i<num.length;i++)
                num[i]*=mx;
        }
        public static void main(String args[]){
            Scanner in=new Scanner(System.in);
            int n=in.nextInt();
            int ar[]=new int[n];
            for(int i=0;i<n;i++){
                ar[i]=in.nextInt();
            }
            int maxnum=findmax(ar);
            submax(ar,maxnum);
            mulmax(ar,maxnum);
            for(int num :ar){
                System.out.print(num+" ");
            }
            }
        }

```

	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-06-String, StringBuffer

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

```java
import java.util.Scanner;

public class prog {
    public static String processingdaWord(String word) {
        int mid = (word.length()-1)/ 2;
        int mid2=(word.length()/2);
        StringBuilder midToBegin = new StringBuilder();
        StringBuilder midToEnd = new StringBuilder();

        for (int i = mid; i >= 0; i--) {
            midToBegin.append(word.charAt(i));
        }

        for (int i = mid2; i < word.length(); i++) {
            midToEnd.append(word.charAt(i));
        }

        return midToBegin.toString() + midToEnd.toString();
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String input1 = sc.nextLine();
        int input2 = sc.nextInt();
        String[] words = input1.split(" ");
        int firstDigit  = input2 / 10;
        int secondDigit = input2 % 10;
        String word1 = words[firstDigit - 1];
        String word2 = words[secondDigit - 1];

        String processedWord1 = processingdaWord(word1);
        String processedWord2 = processingdaWord(word2);

        System.out.println(processedWord1 + " " + processedWord2);

    }
}
```

	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	✓

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

```java
import java.util.*;

public class prog {

    public static String processStrings(String input1, String input2) {
        StringBuilder ccStr = new StringBuilder();
        ccStr.append(input1).append(input2);
        Set<Character> uniqueChars = new TreeSet<Character>((a, b) -> b - a);

        for (char c : ccStr.toString().toCharArray()) {

            if (c != ' ') {
                uniqueChars.add(c);
            }
        }
        if (uniqueChars.isEmpty()) {
            return "null";
        }

        // StringBuilder to form the final result string
        StringBuilder result = new StringBuilder();
        for (char c : uniqueChars) {
            result.append(c);
        }

        return result.toString();
    }

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        String input1 = scanner.nextLine();

        String input2 = scanner.nextLine();

        String result = processStrings(input1, input2);

        System.out.println(result);

    }
}
```

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

Passed all tests! ✓

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

```java
import java.util.Scanner;

public class WordProcessor {

    public static char getAlphabetAtPosition(int position) {
        return (char) ('a' + (position - 1));
    }

    public static String processInput(String input) {

        StringBuilder output = new StringBuilder();

        String[] words = input.split(":");

        for (String word : words) {
            char firstChar = word.charAt(0);
            char secondChar = word.charAt(1);

            if (firstChar == secondChar) {

                output.append(firstChar);
            } else {

                int position1 = firstChar - 'a' + 1;
                int position2 = secondChar - 'a' + 1;
                int diff = Math.abs(position1 - position2);

                output.append(getAlphabetAtPosition(diff));
            }
        }


        return output.toString().toUpperCase();
    }

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        String input = scanner.nextLine();

        String result = processInput(input);
        System.out.println(result);
    }
}
```

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

Passed all tests! ✓

Lab-07-Interfaces

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.

```
interface RBI {
    String parentBank = "RBI";
    double rateOfInterest();

    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 in 2024.");
    }
}

class SBI implements RBI {

    @Override
    public double rateOfInterest() {
        return 7.6;
    }
}

class Karur implements RBI {

    @Override
    public double rateOfInterest() {
        return 7.4;
    }
}

public class Main {
    public static void main(String[] args) {

        SBI sbi = new SBI();
        Karur karur = new Karur();
        sbi.policyNote();
        RBI.regulations();
        System.out.println("SBI rate of interest: " + sbi.rateOfInterest() + " per annum.");
        System.out.println("Karur rate of interest: " + karur.rateOfInterest() + " per annum.");


    }
}
```

	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! ✓

```java
import java.util.Scanner;

interface Playable {
    void play0();
}

class Football implements Playable {
    String name;

    public Football(String name) {
        this.name = name;
    }

    @Override
    public void play0() {
        System.out.println(name + " is Playing football");
    }
}
class Volleyball implements Playable {
    String name;

    public Volleyball(String name) {
        this.name = name;
    }

    @Override
    public void play0() {
        System.out.println(name + " is Playing volleyball");
    }
}
class Basketball implements Playable {
    String name;

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

    @Override
    public void play0() {
        System.out.println(name + " is Playing basketball");
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String name1 = scanner.nextLine();
        String name2 = scanner.nextLine();
        String name3 = scanner.nextLine();
        Playable player1 = new Football(name1);
        Playable player2 = new Volleyball(name2);
        Playable player3 = new Basketball(name3);
        player1.play0();
        player2.play0();
        player3.play0();
    }
}
```

	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	✓

assed all tests! ✓

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 me

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

```java
import java.util.*;
 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);

}
class College implements Football {
    String homeTeam;
    String visitingTeam;

    public void setHomeTeam(String name){
        this.homeTeam=name;
    }
public void setVisitingTeam(String name){
        this.visitingTeam=name;
}
public void homeTeamScored(int points){
    System.out.println(homeTeam+" "+points+" scored");
}
public void visitingTeamScored(int points){
  System.out.println(visitingTeam+" "+points+" scored");
}
public void winningTeam(int p1, int p2){
    if(p1>p2)
        System.out.println(homeTeam +" is the winner!");
    else if(p1<p2)
        System.out.println(visitingTeam+" is the winner!");
    else
        System.out.println("It's a tie match.");
}
}
class prog{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
```

```
        String hname=sc.next();
        String  vteam=sc.next();
        int htpoints=sc.nextInt();
        int vtpoints=sc.nextInt();
    College s= new College();
     s.setHomeTeam(hname);
     s.setVisitingTeam(vteam);
     s.homeTeamScored(htpoints);
     s.visitingTeamScored(vtpoints);
     s.winningTeam(htpoints,vtpoints);

}
}
```

	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-08 - Polymorphism, Abstract Classes, final Key...

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

```java
import java.util.Scanner;

public class VowelChecker {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Read the number of elements in the array
        int n = Integer.parseInt(scanner.nextLine().trim());

        // Read the entire line of strings
        String input = scanner.nextLine().trim();
        String[] arr = input.split(" ");

        // Ensure the number of elements matches the input count
        if (arr.length != n) {
            System.out.println("Input count does not match the number of elements provided.");
            return;
        }

        // Concatenate strings with vowels as the first and last characters
        StringBuilder result = new StringBuilder();
        for (String str : arr) {
            if (!str.isEmpty() && isVowel(str.charAt(0)) && isVowel(str.charAt(str.length() - 1))) {
                result.append(str);
            }
        }

        // Convert to lowercase and print the result
        if (result.length() > 0) {
            System.out.println(result.toString().toLowerCase());
        } else {
            System.out.println("no matches found");
        }
    }

    private static boolean isVowel(char c) {
        return "AEIOUaeiou".indexOf(c) != -1;
    }
}
```

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

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.

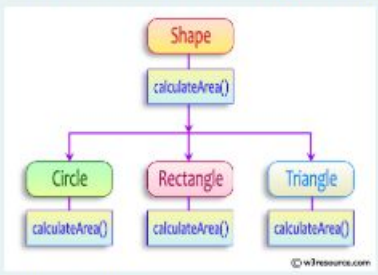
```
class FinalExample {

    // Final variable
            int maxSpeed = 120;

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

class SubClass extends FinalExample {

    public void displayMaxSpeed() {
        System.out.println("Cannot override a final method");
    }

    // You can create new methods here
    public void showDetails() {
        System.out.println("This is a subclass of FinalExample.");
    }
}

class prog {
    public static void main(String[] args) {
        FinalExample obj = new FinalExample();
        obj.displayMaxSpeed();

        SubClass subObj = new SubClass();;
        subObj.showDetails();
    }
}
```

	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! ✓

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

```java
import java.util.Scanner;


abstract class Shape {
    public abstract double calculateArea();
}
class Circle extends Shape {
    private double radius;

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

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

// Rectangle class extending Shape
class Rectangle extends Shape {
    private double length;
    private double breadth;

    // Constructor to initialize length and breadth
    public Rectangle(double length, double breadth) {
        this.length = length;
        this.breadth = breadth;
    }

    // Override calculateArea to compute the area of the rectangle
    @Override
    public double calculateArea() {
        return length * breadth; // Formula: length × breadth
    }
}

// Triangle class extending Shape
class Triangle extends Shape {
    private double base;
    private double height;

    // Constructor to initialize base and height
    public Triangle(double base, double height) {
        this.base = base;
        this.height = height;
    }

    // Override calculateArea to compute the area of the triangle
    @Override
    public double calculateArea() {
        return 0.5 * base * height; // Formula: 0.5 × base × height
    }
}

public class TestShape {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        double radius = scanner.nextDouble();
        Shape circle = new Circle(radius);
        double length = scanner.nextDouble();
        double breadth = scanner.nextDouble();
        Shape rectangle = new Rectangle(length, breadth);
        double base = scanner.nextDouble();
        double height = scanner.nextDouble();
        Shape triangle = new Triangle(base, height);

        System.out.printf("Area of a circle: %.2f%n", circle.calculateArea());
        System.out.printf("Area of a Rectangle: %.2f%n", rectangle.calculateArea());
        System.out.printf("Area of a Triangle: %.2f%n", triangle.calculateArea());

        scanner.close();
    }
}
```

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

Passed all tests! ✓

Lab-09-Exception Handling

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.

```java
import java.util.InputMismatchException;
import java.util.Scanner;

public class SumArray {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int sum = 0;
        try {
            int n = scanner.nextInt();

            int[] arr = new int[n];

            // Take array input from the user
            for (int i = 0; i < n; i++) {
                arr[i] = scanner.nextInt();
                sum += arr[i]; // accumulate the sum of the integers
            }

            System.out.println(sum);

        } catch (InputMismatchException e) {
            System.out.println("You entered bad data.");
        } finally {
            scanner.close();
        }
    }
}
```

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

Passed all tests! ✓

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

```java

class prog {
    public static void main(String[] args) {
        int m = 82;
        trynumber(m);

        int n = 37;
        trynumber(n);
    }

    public static void trynumber(int n) {
        try {

            checkEvenNumber(n);
            System.out.println(n + " is even.");
        } catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        }
    }

    public static void checkEvenNumber(int number) throws Exception {
        if (number % 2 != 0) {
            throw new Exception(number + " is odd.");
        }
    }
}
```

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

Passed all tests! ✓

Write a Java program to handle ArithmeticException and ArrayIndexOutOfBoundsException.

Create an array, read the input from the user, and store it in the array.

Divide the 0th index element by the 1st index element and store it.

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

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

Input:

5

10 0 20 30 40

Output:

java.lang.ArithmeticException: / by zero
I am always executed

Input:

3
10 20 30

Output

java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3
I am always executed

For example:

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

Answer: (penalty regime: 0 %)

```
import java.util.Scanner;

public class ExceptionHandling {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        try {
            // Read the number of elements in the array
            int n = Integer.parseInt(scanner.nextLine().trim());

            // Create an array and populate it with integers
            int[] arr = new int[n];
            for (int i = 0; i < n; i++) {
                arr[i] = scanner.nextInt();
            }

            // Perform the division and handle possible exceptions
            int result = arr[0] / arr[1];
            System.out.println("java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3");
        } catch (ArithmeticException e) {
            System.out.println("java.lang.ArithmeticException: " + e.getMessage());
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("java.lang.ArrayIndexOutOfBoundsException: " + e.getMessage());
        } finally {
            System.out.println("I am always executed");
        }
    }
}

```

	Test	Input	Expected	Got	
✓	1	6 1 0 4 1 2 8	java.lang.ArithmeticException: / by zero I am always executed	java.lang.ArithmeticException: / 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	✓

Lab-10- Collection- List

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 %)

```
import java.util.*;
public class Array
{
    public static void main(String[]args)
    {
        Scanner in=new Scanner(System.in);
        int n=in.nextInt();
        ArrayList<Integer> l=new ArrayList<>(n);
        for(int i=0;i<n;i++)
        l.add(in.nextInt());
        System.out.print("ArrayList: "+l+"\n"+"First : "+l.get(0)+", Last : "+l.get(n-1));
    }
}
```

	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! ✓

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 %)

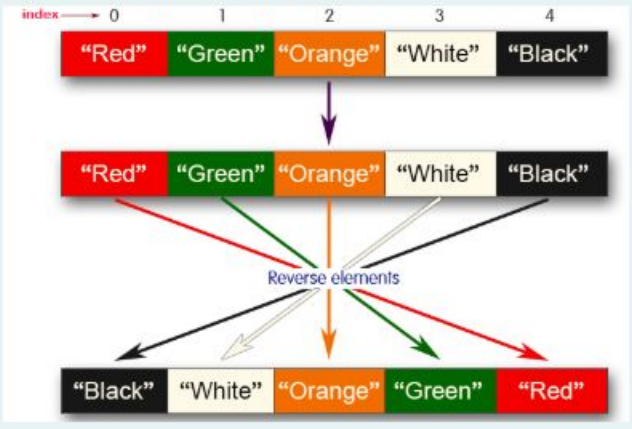
Reset answer

```java
import java.util.ArrayList;
import java.util.Scanner;
public class Prog {
    public static void main(String[] args) {
        Scanner sc= new Scanner(System.in);
        int n = sc.nextInt();
        ArrayList<Integer> list = new ArrayList<Integer>();
        for(int i = 0; i<n;i++)
        list.add(sc.nextInt());
        System.out.println("ArrayList: " + list);
        list.set(1, 100);
        System.out.println("Index of 100 = "+ list.indexOf(100));
        System.out.println("LastIndex of 100 = "+ list.lastIndexOf(100));
        System.out.println(list.contains(200));
        System.out.println("Size Of ArrayList = "+ list.size());
        list.add(1,500);
        list.remove(3);
        System.out.print("ArrayList: " + list);
    }
}
```

	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! ✓

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 %)

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

	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]	✓

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

```java
import java.util.*;
 class prog {
  public static void main(String[] args) {
    Scanner sc= new Scanner(System.in);
    int n = sc.nextInt();
    // Create a HashSet object called numbers
    Set<Integer> num = new HashSet<>();

    // Add values to the set
    for(int i=0;i<n;i++)
        num.add(sc.nextInt());

    int skey=sc.nextInt();

    if (num.contains(skey)) {
        System.out.println(skey + " was found in the set.");
    } else {
        System.out.println(skey + " was not found in the set.");
    }
  }
}
```

	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! ✓

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

```java
import java.util.*;

public class prog {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int n1 = scanner.nextInt();
        scanner.nextLine();

        HashSet<String> set1 = new HashSet<>();
        for (int i = 0; i < n1; i++) {
            set1.add(scanner.next());
        }

        int n2 = scanner.nextInt();
        scanner.nextLine();

        HashSet<String> set2 = new HashSet<>();
        for (int i = 0; i < n2; i++) {
            set2.add(scanner.next());
        }

        set1.retainAll(set2);

        for (String element : set1) {
            System.out.println(element);
        }
    }
}
```

	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! ✓

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

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

        String name;
        int num;
        Scanner sc= new Scanner(System.in);
        int n=sc.nextInt();
         for(int i =0;i<n;i++)
         {
             name=sc.next();
             num= sc.nextInt();
             map.put(name,num);
         }

        //Printing key-value pairs

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

        for (Entry<String, Integer> entry : entrySet)
        {
            System.out.println(entry.getKey()+" : "+entry.getValue());
        }
        System.out.println("----------");
        //Creating another HashMap

        HashMap<String, Integer> anotherMap = new HashMap<String, Integer>();

        //Inserting key-value pairs to anotherMap using put() method

        anotherMap.put("SIX", 6);

        anotherMap.put("SEVEN", 7);

        //Inserting key-value pairs of map to anotherMap using putAll() method

        anotherMap.putAll(map);  // code here

        //Printing key-value pairs of anotherMap

        entrySet = anotherMap.entrySet();

        for (Entry<String, Integer> entry : entrySet)
        {
            System.out.println(entry.getKey()+" : "+entry.getValue());
```

```
        }

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

        map.putIfAbsent("FIVE", 5);

        //Retrieving a value associated with key 'TWO'

        int value = map.get("TWO");
        System.out.println(value);

         //Checking whether key 'ONE' exist in map

        System.out.println(map.containsKey("ONE"));
        //Checking whether value '3' exist in map

        System.out.println(map.containsValue(3));

        //Retrieving the number of key-value pairs present in map

        System.out.println(map.size());
    }
}
```

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

Passed all tests! ✓

Lab-12-Introduction to I/O, I/O Operations, Object...

Answer: (penalty regime: 0 %)

```
import java.util.Scanner;

public class prog {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        String input = scanner.nextLine(); ;
        int caseOption = scanner.nextInt();

        String result = reverseWords(input, caseOption);
        System.out.println(result);
    }

    public static String reverseWords(String input, int caseOption) {

        String[] words = input.split(" ");
        StringBuilder result = new StringBuilder();

        for (String word : words) {
            String reversedWord = reverseWordWithCase(word, caseOption);
            result.append(reversedWord).append(" ");
        }

        return result.toString().trim();
    }

    public static String reverseWordWithCase(String word, int caseOption) {
        char[] reversed = new char[word.length()];

        for (int i = 0; i < word.length(); i++) {
            reversed[i] = word.charAt(word.length() - 1 - i);
        }

        if (caseOption == 1) {
            for (int i = 0; i < word.length(); i++) {
                if (Character.isUpperCase(word.charAt(i))) {
                    reversed[i] = Character.toUpperCase(reversed[i]);
                } else if (Character.isLowerCase(word.charAt(i))) {
                    reversed[i] = Character.toLowerCase(reversed[i]);
                }
            }
        }

        return new String(reversed);
    }
}
```

	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! ✓

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

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

Each alphabet is represented by a sequence of 0s.

This is as mentioned below:

Z : 0

Y : 00

X : 000

W : 0000

V : 00000

U : 000000

T : 0000000

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

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

Example 1:

input1: 010010001

The decoded string (original word) will be: ZYX

Example 2:

input1: 000010000000000000000010000000000010000000001000000000001

The decoded string (original word) will be: WIPRO

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

For example:

Input	Result
010010001	ZYX
000010000000000000000010000000000010000000001000000000001	WIPRO

Answer: (penalty regime: 0 %)

```
import java.util.Scanner;

public class prog {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        String encoded = scanner.nextLine();

        StringBuilder decoded = new StringBuilder();

        String[] encodedLetters = encoded.split("1");

        for (String letter : encodedLetters) {
            if (!letter.isEmpty()) {
                int zeroCount = letter.length();

                char decodedChar = (char) ('Z' - (zeroCount - 1));

                decoded.append(decodedChar);
            }
        }

        System.out.println(decoded.toString());

    }
}
```

	Input	Expected	Got	
✓	010010001	ZYX	ZYX	✓
✓	000010000000000000000010000000000010000000001000000000001	WIPRO	WIPRO	✓

Passed all tests! ✓

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 b c	8

```java
import java.util.*;

public class prog {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        String input1 = scanner.nextLine().replaceAll("\\s+", "");

        String input2 = scanner.nextLine().replaceAll("\\s+", "");

        Set<Character> set1 = new HashSet<>();
        for (char c : input1.toCharArray()) {
            set1.add(c);
        }

        Set<Character> common = new HashSet<>();
        for (char c : input2.toCharArray()) {
            if (set1.contains(c)) {
                common.add(c);
            }
        }
        int sum1 = 0;
        for (char c : common) {
            sum1 += (int) c;
        }

        int singleDigitSum = getSingleDigitSum(sum1);

        System.out.println(singleDigitSum);

    }
    private static int getSingleDigitSum(int num) {
        while (num >= 10) {
            int tempSum = 0;
            while (num > 0) {
                tempSum += num % 10;
                num /= 10;
            }
            num = tempSum;
        }
        return num;
    }
}
```

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