WEEK 9 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.

```
import java.util.Scanner;

import java.util.InputMismatchException;

 class prog {

  public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);

    int length = sc.nextInt();

    // create an array to save user input
```

```
 int[] name = new int[length];
 int sum=0;//save 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. */
  try
   {for (int i=0;i<length;i++){
      name[i]=sc.nextInt();
      sum+=name[i];
   }
   System.out.print(sum);

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

  }
//System.out.println(sum);
```

}

	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 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

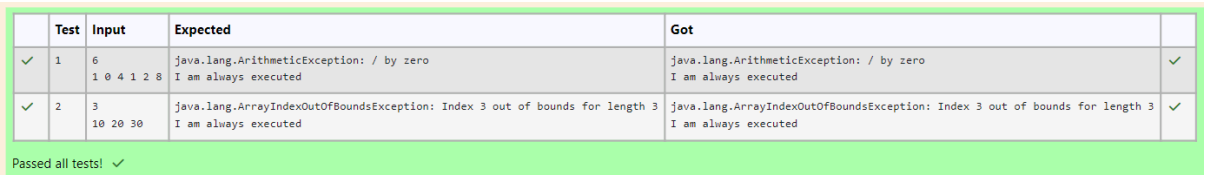
import java.util.Scanner ;

```
public class Main {
   public static void main (String [] args){
      Scanner sc=new Scanner (System.in);
      int a=sc.nextInt();

      int [] arr= new int [a];
      for(int i=0;i<a;i++){
         arr[i]=sc.nextInt();
      }
      int b;
      try {
       b=arr[0]/arr[1];
      }
      catch(ArithmeticException e){
         System.out.println(e);
         System.out.println("I am always executed");

      }
      try{
         arr[3]=15;
      }
      catch(ArrayIndexOutOfBoundsException e){
         System.out.println(e);
         System.out.print("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	✓

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.

```
class prog {
  public static void main(String[] args) {
    int n = 82;
    trynumber(n);
    n = 37;
    // call the trynumber(n);
    trynumber(n);


  }


  public static void trynumber(int n) {
    try {
       //call the checkEvenNumber()
         checkEvenNumber(n);
     System.out.println(n + " is even.");
    } catch (Exception e) {
```

```
    System.out.println("" + e.getMessage());
   }
  }

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

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

Passed all tests! ✓