**Exceptions - String to Integer**

**Objective**   
Today, we're getting started with *Exceptions* by learning how to parse an integer from a string and print a custom error message. Check out the [Tutorial](https://www.hackerrank.com/challenges/30-exceptions-string-to-integer/tutorial) tab for learning materials and an instructional video!

**Task**   
Read a string, , and print its integer value; if  cannot be converted to an integer, print Bad String.

**Note:** You *must* use the String-to-Integer and exception handling constructs built into your submission language. If you attempt to use loops/conditional statements, you will get a  score.

**Input Format**

A single string, .

**Constraints**

* 1<|S|<6, where |S| is the length of string .
* |S| is composed of *either* lowercase letters (a-z) *or* decimal digits (0-9).

**Output Format**

Print the parsed integer value of , or Bad String if  cannot be converted to an integer.

**Sample Input 0**

3

**Sample Output 0**

3

**Sample Input 1**

za

**Sample Output 1**

Bad String

**Explanation**

*Sample Case* contains an integer, so it should not raise an exception when we attempt to convert it to an integer. Thus, we print the .   
*Sample Case* does not contain any integers, so an attempt to convert it to an integer will raise an exception. Thus, our exception handler prints Bad String.

import java.io.\*;

import java.util.\*;

import java.text.\*;

import java.math.\*;

import java.util.regex.\*;

​

public class Solution {

​

   public static void main(String[] args) {

       Scanner in = new Scanner(System.in);

       String S = in.next();

       try

      {

           int i = Integer.parseInt(S);

           System.out.println(i);

      }

       catch(Exception e)

      {

           System.out.println("Bad String");

      }

  }

}

Objective

Today, we're getting started with Exceptions by learning how to parse an integer from a string and print a custom error message. Check out the Tutorial tab for learning materials and an instructional video!

Task

Read a string, , and print its integer value; if cannot be converted to an integer, print Bad String.

Note: You must use the String-to-Integer and exception handling constructs built into your submission language. If you attempt to use loops/conditional statements, you will get a score.

Input Format

A single string, .

Constraints

, where is the length of string .

is composed of either lowercase letters () or decimal digits ().

Output Format

Print the parsed integer value of , or Bad String if cannot be converted to an integer.

Sample Input 0

3

Sample Output 0

3

Sample Input 1

za

Sample Output 1

Bad String

Explanation

Sample Case contains an integer, so it should not raise an exception when we attempt to convert it to an integer. Thus, we print the .

Sample Case does not contain any integers, so an attempt to convert it to an integer will raise an exception. Thus, our exception handler prints Bad String.

Solved score: 15.00pts

1

import java.io.\*;

2

import java.util.\*;

3

import java.text.\*;

4

import java.math.\*;

5

import java.util.regex.\*;

6

​

7

public class Solution {

8

​

9

public static void main(String[] args) {

10

Scanner in = new Scanner(System.in);

11

String S = in.next();

12

try

13

{

14

int i = Integer.parseInt(S);

15

System.out.println(i);

16

}

17

catch(Exception e)

18

{

19

System.out.println("Bad String");

20

}

21

}

22

}