Java Loops II

Problem:-

Java has 8 primitive data types; *char, boolean, byte, short, int, long, float, and double*. For this exercise, we'll work with the primitives used to hold integer values (*byte, short, int,* and *long*):

- A *byte* is an 8-bit signed integer.
- A *short* is a 16-bit signed integer.
- An *int* is a 32-bit signed integer.
- A *long* is a 64-bit signed integer.

Given an input integer, you must determine which primitive data types are capable of properly storing that input.

Input Format

The first line contains an integer, T, denoting the number of test cases.
Each test case, T, is comprised of a single line with an integer, n, which can be arbitrarily large or small.

Output Format

For each input variable n and appropriate primitive data type, you must determine if the given primitives are capable of storing it. If yes, then print:

```
n can be fitted in:
* dataType
```

```
If there is more than one appropriate data type, print each one on its own line
and order them by size (i.e.: byte<short<int<long).
```

```
If the number cannot be stored in one of the four aforementioned primitives,
print the line:
```

```
n can't be fitted anywhere.
```

Sample Input

```
5
-150
150000
1500000000
213333333333333333333333333333333333
-100000000000000
```

Sample Output

```
-150 can be fitted in:
* short
* int
* long
```

```
150000 can be fitted in:
* int
* long
1500000000 can be fitted in:
* int
* long
213333333333333333333333333333333333 can't be fitted anywhere.
-100000000000000 can be fitted in:
* long
```

Explanation

-150 can be stored in a *short*, an *int*, or a *long*.

213333333333333333333333 is very large and is outside of the allowable range of values for the primitive data types discussed in this problem.

Solution:-

```
 import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;
public class JavaDatatypes {
   static String whoCanFitTheNumber(String numString)
   {
      String answer = "";
      try{
         long num = Long.parseLong(numString);
         answer = numString + " can be fitted in:\n";
         if((num<=Byte.MAX_VALUE) && (num>=Byte.MIN_VALUE)){
            answer = answer.concat("* byte\n* short\n* int\n* long");
            }else if((num <= Short.MAX_VALUE) && (num >= Short.MIN_VALUE)){
              answer = answer.concat("* short\n* int\n* long");
            }else if((num <= Integer.MAX_VALUE) && (num >= Integer.MIN_VALUE)){
                answer = answer.concat("* int\n* long");
            }else{
                answer = answer.concat("* long");
            }
        }catch (NumberFormatException e){
            answer = numString+" can't be fitted anywhere.";
        }
        return answer;
    }
    public static void main(String[] args) {

            Scanner scanner = new Scanner(System.in);
        int numTestCases = scanner.nextInt() ;
        scanner.nextLine();
        for(int i=0; i<numTestCases;i++){
            String numString = scanner.nextLine();
            System.out.println(whoCanFitTheNumber(numString));
        }
    }
}
```