Java Datatypes ★

25 more points to get your next star!

Rank: **716609** | Points: **55/80**

Your Java Datatypes submission got 10.00 points. Share Tweet

You are now 25 points away from the 3rd star for your java badge.

Try the next challenge | **Try a Random Challenge**

Problem Submissions Leaderboard Editorial

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.

To get you started, a portion of the solution is provided for you in the editor.

Reference: https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html

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 $dataType$, 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
21333333333333333333333333333333333333 can't be fitted anywhere.
-100000000000000 can be fitted in:
* long
```

Explanation

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

$21333333333333333333333333333333333333$ is very large and is outside of the allowable range of values for the primitive data types discussed in this problem.

Change Theme Language Java 7

```
1  import java.util.*;
2  import java.io.*;
3
4  class Solution{
5      public static void main(String []argh)
6      {
7          Scanner sc = new Scanner(System.in);
8          int t=sc.nextInt();
9          for(int i=0;i<t;i++)
10         {
11             try
12             {
13                 long x=sc.nextLong();
14                 System.out.println(x+" can be fitted in:");
15                 if(x>=-128 && x<=127)System.out.println("* byte");
16                  if (x>=-32768 && x<=32767)System.out.println("* short");
17                  if (x>=-2147483648 && x<=2147483647)System.out.println("* int");
18                  if (x>=-9223372036854775808L && x<=9223372036854775807L)
   System.out.println("* long");
19             }
20             catch(Exception e)
21             {
22                 System.out.println(sc.next()+" can't be fitted anywhere.");
23             }
24
25         }
26     }
27 }
28
29
30
31
```

Line: 31 Col: 1

Upload Code as File Test against custom input Run Code Submit Code

You have earned 10.00 points!

You are now 25 points away from the 3rd star for your java badge.

17% 55/80

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0

Test case 1

Test case 2

Test case 3

```
-150 can be fitted in:
* short
* int
* long
150000 can be fitted in:
* int
* long
1500000000 can be fitted in:
* int
* long
21333333333333333333333333333333333333 can't be fitted anywhere.
-100000000000000 can be fitted in:
* long
```