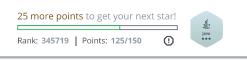# Java Datatypes ★

Problem   Submissions   Leaderboard   Editorial

RATE THIS CHALLENGE
★ ★ ★ ★ ★

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
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.

213333333333333333333333333333333333333 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 |

```java
1    import java.util.*;
2    import java.io.*;
3
4
5
6    class Solution{
7        public static void main(String []argh)
8        {
9
10           Scanner sc = new Scanner(System.in);
11       int t=sc.nextInt();
12
13       for(int i=0;i<t;i++)
14       {
15
16           try
17           {
18               long x=sc.nextLong();
19               System.out.println(x+" can be fitted in:");
20               if(x>=-128 && x<=127)System.out.println("* byte");
21               //Complete the code
22               if(x >= -Math.pow(2, 15) && x <= Math.pow(2, 15) - 1)
23                   System.out.println("* short");
24               if(x >= -Math.pow(2, 31) && x <= Math.pow(2, 31) - 1)
25                   System.out.println("* int");
26               if(x >= -Math.pow(2, 63) && x <= Math.pow(2, 63) - 1)
27                   System.out.println("* long");
28           }
29           catch(Exception e)
30           {
31               System.out.println(sc.next()+" can't be fitted anywhere.");
32           }
33
34       }
35   }
36   }
37
```

Line: 37 Col: 1

⬆ Upload Code as File   ☐ Test against custom input

Run Code   **Submit Code**

⬆ Upload Code as File   ☐ Test against custom input

Run Code   **Submit Code**