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.

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
```

Submissions | Leaderboard | Discussions | Editorial

Change Theme Language Java 7

```
import java.util.*;
import java.io.*;
class Solution{
    public static void main(String []argh)
    {
        Scanner sc = new Scanner(System.in);
        int t=sc.nextInt();
        for(int i=0;i<t;i++)
        {
            try
            {
                long x=sc.nextLong();
                System.out.println(x+" can be fitted in:");
                if(x>=-128 && x<=127)System.out.println("* byte");
                 if (x>=-32768 && x<=32767)System.out.println("* short");
                 if (x>=-2147483648 && x<=2147483647)System.out.println("* int");
                 if (x>=-9223372036854775808L && x<=9223372036854775807L)System.out.println("* long");
            }
            catch(Exception e)
            {
                System.out.println(sc.next()+" can't be fitted anywhere.");
            }
        }
    }
}
```

Line: 25 Col: 2

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

hackerrank.com/challenges/java-datatypes/problem?isFullScreen=true

HackerRank Prepare > Java > Introduction > Java Datatypes

Exit Full Screen View

Problem | Submissions | Leaderboard | Discussions | 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
```

```
24      }
25  }
```

Line: 25 Col: 2

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

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0

Input (stdin) Download

```
5
-150
150000
1500000000
21333333333333333333333333333333333333
-100000000000000
```

Your Output (stdout)

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

Search | ENG IN | 1:52 PM 2/13/2023

hackerrank.com/challenges/java-datatypes/problem?isFullScreen=true

HackerRank Prepare > Java > Introduction > Java Datatypes

Exit Full Screen View

Problem | Submissions | Leaderboard | Discussions | 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
```

```
24      }
25  }
```

Line: 25 Col: 2

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

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0

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

Expected Output | Download

```
-150 can be fitted in:
* short
* int
* long
150000 can be fitted in:
```

Search | ENG IN | 1:52 PM 2/13/2023

hackerrank.com/challenges/java-datatypes/problem?isFullScreen=true

HackerRank Prepare > Java > Introduction > Java Datatypes

Exit Full Screen View

Problem | Submissions | Leaderboard | Discussions | 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
```

```
24      }
25  }
```

Line: 25 Col: 2

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

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0

```
-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.
-10000000000000 can be fitted in:
* long
```

Search | ENG IN | 1:52 PM 2/13/2023

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
```

Line: 25 Col: 2

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

You have earned 10.00 points!
You are now 22 points away from the 3rd star for your java badge.

27% 58/80

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

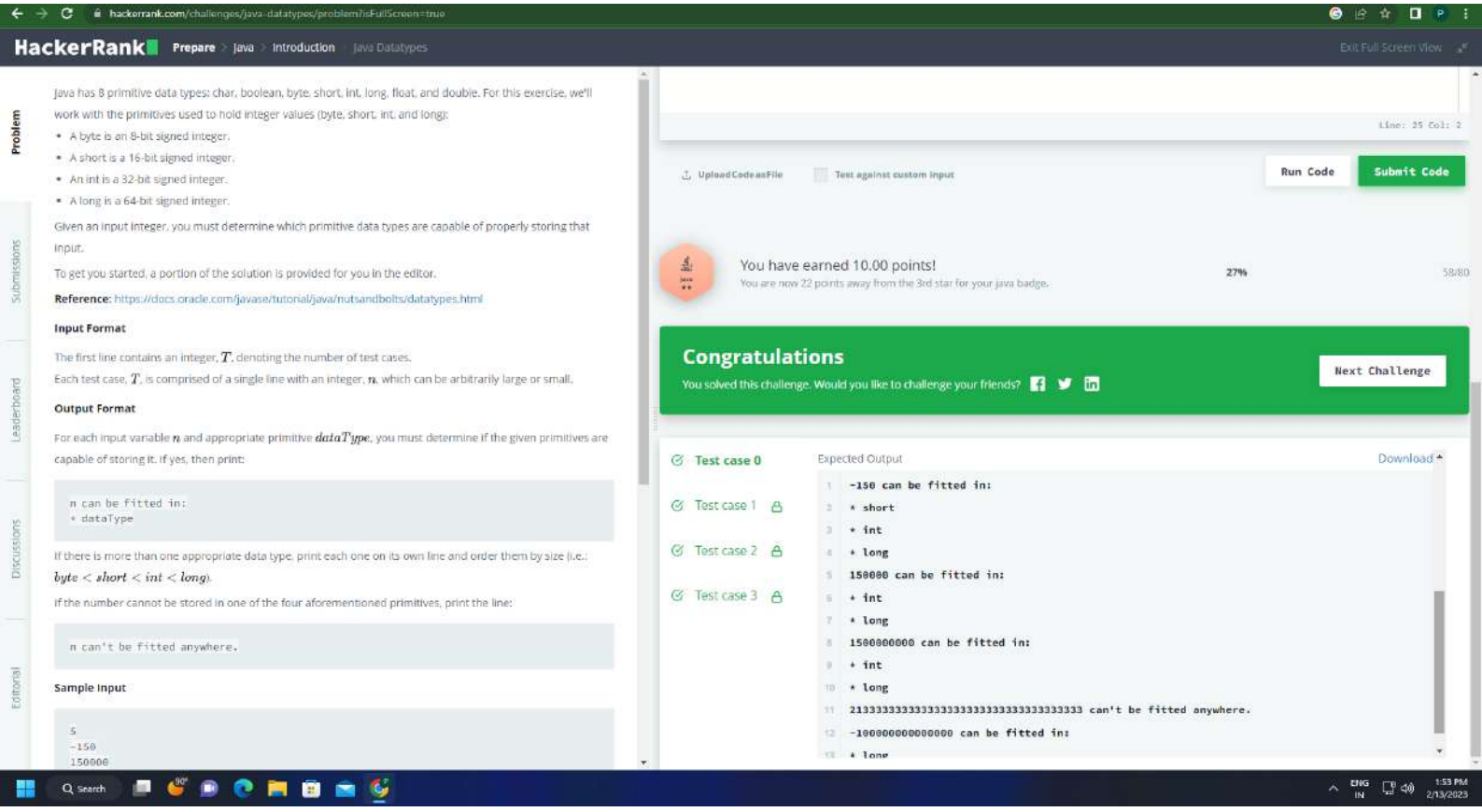
Compiler Message

Success

Input (stdin) Download

```
5
-150
150000
1500000000
21333333333333333333333333333333333
-100000000000000
```

Expected Output Download


hackerrank.com/challenges/java-datatypes/problem?isFullScreen=true
HackerRank
Prepare > Java > Introduction > Java Datatypes
Exit Full Screen View
Problem
Submissions
Leaderboard
Discussions
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
Line: 25 Col: 2
Upload Code as File
Test against custom input
Run Code
Submit Code
You have earned 10.00 points!
You are now 22 points away from the 3rd star for your java badge.
27%
58/80
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
Expected Output
Download
-150 can be fitted in:
* short
* int
* long
150000 can be fitted in:
* int
* long
1500000000 can be fitted in:
* int
* long
21333333333333333333333333333333333 can't be fitted anywhere.
-100000000000000 can be fitted in:
* long
Search
ENG
IN
1:53 PM
2/13/2023