Python: Division

Problem Statement

In Python, there are two kinds of division: integer division and float division.

During the time of Python 2, when you divided one integer by another integer, no matter what, the result would always be an integer.

For example:

```
>>> 4/3
1
```

In order to make this a float division, you would need to convert one of the arguments into a float.

For example:

```
>>> 4/3.0
1.3333333333333333
```

Since Python doesn't declare data types in advance, you never know when you want to use integers and when you want to use a float. Since floats lose precision, it's not advised to use them in integral calculations.

To solve this problem, future Python modules included a new type of division called integer division given by the operator `//`.

Now, `/` performs float division, and `//` performs integer division.

In **Python 2**, we will import a feature from the module **__future__** called `division`.

```
>>> from __future__ import division
>>> print 4/3
1.3333333333333333
>>> print 4//3
1
```

Note: The `__` in `__future__` is a double underscore.

Task

Read two integers and print two lines. The first line should contain integer division, $a//b$. The second line should contain float division, a/b.

You don't need to perform any rounding or formatting operations.

Input Format

The first line contains the first integer, a. The second line contains the second integer, b.

Output Format

Print the two lines as described above.

Sample Input

```
4
3
```

sample Output

```
1
1.3333333333333333
```