Python: Division

Problem Statement

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

A long time ago during the era of Python 2, when you divided an integer with another integer, no matter the result, it was always an integer.

Example

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

And in order to make this a float division, you'd need to convert one of the arguments into a float.

Example

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

Since Python doesn't have data types declared in advance, you never know when you want integers and when you want a float. And since floats lose precision, it's unadvisable 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 module from `__future__` called `division`.

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

Task

Read two integers and print two lines, such that the first line contains integer division and the second line contains float division.

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

Input Format

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

Output Format

Print two lines as described above.

Sample Input

```
4
3
```

sample Output

```
1
1.3333333333333333
```