Integer and float numbers

Integer arithmetics

We already know the following operators which may be applied to numbers: `+`, `-`, `*` and `**`. The division operator `/` for integers gives a floating-point real number (an object of type `float`). The exponentiation `**` also returns a float when the power is negative:

```
print(17 / 3)  # gives 5.66666666667
print(2 ** 4)  # gives 16
print(2 ** -2) # gives 0.25
```

There's a special operation for integer division where the remainder is discarded: `//`. The operation that yields a remainder of such a division looks like `%`. Both operation always yield an object of type `int`.

```
print(17 / 3)   # gives 5.66666666667
print(17 // 3)  # gives 5
print(17 % 3)   # gives 2
```

Floating-point numbers

When we read an integer value, we read a line with `input()` and then cast a string to integer using `int()`. When we read a floating-point number, we need to cast the string to float using `float()`:

```
x = float(input())
print(x)
```

Floats with very big or very small absolute value can be written using a scientific notation. Eg., the distance from the Earth to the Sun is $1.496 \cdot 10^{11}$, or `1.496e11` in Python. The mass of one molecule of the water is $2.99 \cdot 10^{-23}$, or `2.99e-23` in Python.

One can cast float objects to int objects by discarding the fraction part using the `int()` function. This function demonstrates so called *rounding towards zero* behavior:

```
print(int(1.3))   # gives 1
print(int(1.7))   # gives 1
print(int(-1.3))  # gives -1
print(int(-1.7))  # gives -1
```

There's also a function `round()` that performs the usual rounding:

```
print(round(1.3))   # gives 1
print(round(1.7))   # gives 2
```

print(round(-1.3)) # gives -1
print(round(-1.7)) # gives -2

Floating-point real numbers can't be represented with exact precision due to hardware limitations. This can lead to cumbersome effects. See the Python docs for the details.

print(0.1 + 0.2) # gives 0.30000000000000004

math module

Python has many auxiliary functions for calculations with floats. They can be found in the `math` module.

To use this module, we need to import it first by writing the following instruction at the beginning of the program:

```
import math
```

For example, if we want to find a ceiling value for `x` - the smallest integer not less than `x` - we call the appropriate function from the math module: `math.ceil(x)`. The syntax for calling functions from modules is always the same:`module_name.function_name(argument_1, argument_2, ...)`

import math
x = math.ceil(4.2)
print(x)
print(math.ceil(1 + 3.8))

There's another way to use functions from modules: to import the certain functions by naming them:

from math import ceil
x = 7 / 2
y = ceil(x)
print(y)

Some of the functions dealing with numbers - `int()`, `round()` and `abs()` (absolute value aka modulus) - are built-in and don't require any imports.

All the functions of any standard Python module are documented on the official Python website. Here's the description for math module. The description of some functions is given:

Function	Description
Rounding	
`floor(x)`	Return the floor of x, the largest integer less than or equal to x.
`ceil(x)`	Return the ceiling of x, the smallest integer greater than or equal to x.
Roots and logarithms	
`sqrt(x)`	Return the square root of x

`log(x)`	With one argument, return the natural logarithm of x (to base e). With two arguments, return the logarithm of x to the given base
`e`	The mathematical constant e = 2,71828...
Trigonometry	
`sin(x)`	Return the sine of x radians
`asin(x)`	Return the arcsine of x, in radians
`pi`	The mathematical constant π = 3.1415...

Problem «Last digit of integer» (Easy)

Statement

Given an integer number, print its last digit.

Your solution

```
# Read an integer:
a = int(input())
# Print a value:
print (a % 10)
```

Problem «Tens digit» (Medium)

Statement

Given an integer. Print its tens digit.

Your solution

```
# Read an integer:
a = int(input())
print((a%100)//10)
```

Problem «Sum of digits» (Medium)

Statement

Given a three-digit number. Find the sum of its digits.

Your solution

```
# Read an integer:
n = int(input())
# Print a value:
```

```
first  =  n % 10
second = (n%100 - n%10)// 10
third  = (n%1000 - n%100)//100
print(first + second + third)
```

Suggested solution

```
n = int(input())
a = n // 100
b = n // 10 % 10
c = n % 10
print(a + b + c)
```

Problem «Fractional part» (Easy)

Statement

Given a positive real number, print its fractional part.

Your solution

```
# Read an integer:
n = float(input())
# Print a value:
print( n - int(n))
```

Problem «First digit after decimal point» (Easy)

Statement

Given a positive real number, print its first digit to the right of the decimal point.

Your solution

```
# Read an integer:
n = float(input())
print(int((n*10))%10)
```

Suggested solution

```
x = float(input())
print(int(x * 10) % 10)
```

Problem «Car route» (Easy)

Statement

A car can cover distance of N kilometers per day. How many days will it take to cover a route of length M kilometers? The program gets two numbers: N and M.

Your solution

```
from math import ceil
n = int(input())
m = int(input())
print(ceil(m / n))
```

Problem «Digital clock» (Hard)

Statement

Given the integer N - the number of minutes that is passed since midnight - how many hours and minutes are displayed on the 24h digital clock?

The program should print two numbers: the number of hours (between 0 and 23) and the number of minutes (between 0 and 59).

For example, if N = 150, then 150 minutes have passed since midnight - i.e. now is 2:30 am. So the program should print `2 30`.

Your solution

```
# Read an integer:
n = int(input())
# format hours based on interger
h = n // 60
# Print the number of hours
print(h)
# Print the number of minutes
print( n % 60 )
```

Suggested solution

```
n = int(input())
hours = n // 60
minutes = n % 60
print(hours, minutes)
```

Problem «Total cost» (Medium)

Statement

A cupcake costs A dollars and B cents. Determine, how many dollars and cents should one pay for N cupcakes. A program gets three numbers: A, B, N. It should print two numbers: total cost in dollars and cents.

Your solution

```
# Read an integer:
A = int(input())
B = int(input())
N = int(input())
# Print a value:
print ( A*N + ((B*N) // 100))
print ( B*N % 100)
```

Suggested solution

```
a = int(input())
b = int(input())
n = int(input())
cost = n * (100 * a + b)
print(cost // 100, cost % 100)
```

Problem «Clock face - 1» (Hard)

Statement

H hours, M minutes and S seconds are passed since the midnight (0 ≤ H < 12, 0 ≤ M < 60, 0 ≤ S < 60). Determine the angle (in degrees) of the hour hand on the clock face right now.

Your solution

```
# Read an integer:
H = int(input())
M = int(input())
S = int(input())
# Print a value:
print( H*360/12 + (M*360/(12*60)) + (S*360/(12*60*60)))
```

Problem «Clock face - 2» (Hard)

Statement

Hour hand turned by α degrees since the midnight. Determine the angle by which minute hand turned since the start of the current hour. Input and output in this problems are floating-point numbers.

Your solution

```
# Read in integers
A = float(input())
# Print value:
H = int((A * 12)/360)
M = ((A - H*30)*(60*12)/360)
print(M*360/60)
```

Suggested solution

```
alpha = float(input())
print(alpha % 30 * 12)
```