Input, print and numbers

How to read and write in Python

Every program is eventually a data processor, so we should know how to input and output data within it. There exists a function, `print()`, to output data from any Python program. To use it, pass a comma separated list of arguments that you want to print to the `print()` function. Let's see an example. Press "run" and then "next" to see how the program is being executed line by line:

```
print(5 + 10)
print(3 * 7, (17 - 2) * 8)
print(2 ** 16)  # two stars are used for exponentiation (2 to the power of 16)
print(37 / 3)  # single forward slash is a division
print(37 // 3)  # double forward slash is an integer division
        # it returns only the quotient of the division (i.e. no remainder)
print(37 % 3)  # percent sign is a modulus operator
        # it gives the remainder of the left value divided by the right value
```

To input data into a program, we use `input()`. This function reads a single line of text, as a String.
Here's a program that reads the user's name and greets them:

```
print('What is your name?')
name = input()  # read a single line and store it in the variable "name"
print('Hi ' + name + '!')
```

Sum of numbers and strings

Let's try to write a program that inputs two numbers and prints their sum. We read the two numbers and store them in the variables `a` and `b` using the assignment operator `=`. On the left side of an assignment operator we put the name of the variable. The name could be a string of latin characters (`A-Z`, `a-z`, `0-9`, `_`) but **must** start with a letter in the range `A-Z` or `a-z`. On the right side of an assignment operator we put any expression that Python can evaluate. The name starts pointing to the result of the evaluation. Read this example, run it and look at the output:

```
a = input()
b = input()
```

```
s = a + b
print(s)
```

After running the example we can see that it prints `57`. As we were taught in school, `5 + 7` gives `12`. So, the program is wrong, and it's important to understand why. The thing is, in the third line `s = a + b` Python has "summed" two strings, rather than two numbers. The sum of two strings in Python works as follows: they are just glued one after another. It's also sometimes called "string concatenation".

Do you see in the variable inspector, on the right hand side, that the values bound to variables `a` and `b` are wrapped in quotes? That means that the values there are string, not numbers. Strings and numbers are represented in Python differently.

All the values in Python are called "objects". Every object has a certain type. The number 2 corresponds to an object "number 2" of type "int" (i.e., an integer number). The string `'hello'` corresponds to an object "string 'hello'" of type "str". Every floating-point number is represented as an object of type "float". The type of an object specifies what kind of operations may be applied to it. For instance, if the two variables `"first"` and `"second"` are pointing to the objects of type `int`, Python can multiply them. However, if they are pointing to the objects of type `str`, Python can't do that:

```
first = 5
second = 7
print(first * second)
# you can use single or double quotes to define a string
first = '5'
second = "7"
print(first * second)
```

To cast (convert) the string of digits into an integer number, we can use the function `int()`. For example, `int('23')`gives an int object with value `23`.

Given the information above, we can now fix the incorrect output and output the sum of the two numbers correctly:

```
a = int(input())
b = int(input())
s = a + b
print(s)
```

Problem «Sum of three numbers» (Easy)

Statement

Write a program that takes three numbers and prints their sum. Every number is given on a separate line.

Your solution

```
# This program reads thre numbers and prints their sum:
a = int(input())
b = int(input())
c = int(input())
print(a + b +c)
```

Problem «Area of right-angled triangle» (Easy)

Statement

Write a program that reads the length of the base and the height of a right-angled triangle and prints the area. Every number is given on a separate line.

Your solution

```
# Read the numbers b and h like this:
b = int(input())
h = int(input())
# Print the area of triangle
print(b*h/2)
```

Problem «Hello, Harry!» (Easy)

Statement

Write a program that greets the user by printing the word "Hello", a comma, the name of the user and an exclamation mark after it. See the examples below.

Warning. Your program's output should strictly match the desired one, character by character. There shouldn't be any space between the name and the exclamation mark. You can use `+` operator to concatenate two strings. See the lesson for details.

Your solution

```
# Read in value:
a = input()
# Print the value in greeting:
```

```
print( "Hello, " + a + "!")
```

Problem «Apple sharing» (Medium)

Statement

`N` students take `K` apples and distribute them among each other evenly. The remaining (the undivisible) part remains in the basket. How many apples will each single student get? How many apples will remain in the basket?

The program reads the numbers `N` and `K`. It should print the two answers for the questions above.

Your solution

```
# Read the numbers like this:
n = int(input())
k = int(input())
# Print the number of apples per student
print(k // n)
# Print the number of apples left over
print(k % n )
```

Problem «Previous and next» (Medium)

Statement

Write a program that reads an integer number and prints its previous and next numbers. See the examples below for the exact format your answers should take. There shouldn't be a space before the period.

Remember that you can convert the numbers to strings using the function `str`.

Your solution

```
# Read an integer:
a = int(input())
# print next
print("The next number for the number " + str(a) + " is "
        + str(a + 1))
# print previous
print("The previous number for the number " + str(a) + " is "
        + str(a - 1))
```

Problem «School desks» (Hard)

Statement

A school decided to replace the desks in three classrooms. Each desk sits two students. Given the number of students in each class, print the smallest possible number of desks that can be purchased.

The program should read three integers: the number of students in each of the three classes, `a`, `b` and `c`respectively.

In the first test there are three groups. The first group has 20 students and thus needs 10 desks. The second group has 21 students, so they can get by with no fewer than 11 desks. 11 desks is also enough for the third group of 22 students. So we need 32 desks in total.

Your solution

```
# Read in integers:
a = int(input())
b = int(input())
c = int(input())
# Print minimum number of desks needed
print ( a // 2 + a % 2 + b // 2 + b % 2 + c // 2 + c % 2)
```