

# Hexaware Coding Challenge Plan Day - 1

## Problem 1 - Python if-else

**Congratulations**

You solved this challenge. Would you like to challenge your friends? [f](#) [t](#) [in](#) [Next Challenge](#)

**Test case 0** **Test case 1** **Test case 2** **Test case 3** **Test case 4** **Test case 5** **Test case 6**

**Compiler Message**

Success

**Input (stdin)** [Download](#)

1 3

**Expected Output** [Download](#)

1 Weird

## Problem - 2 Arithmetic Operators

**HackerRank** Prepare > Python > Introduction > Arithmetic Operators

Check Tutorial tab to know how to solve.

**Task**

The provided code stub reads two integers from STDIN,  $a$  and  $b$ . Add code to print three lines where:

1. The first line contains the sum of the two numbers.
2. The second line contains the difference of the two numbers (first - second).
3. The third line contains the product of the two numbers.

**Example**

$a = 3$   
 $b = 5$

Print the following:

```
8
-2
15
```

**Input Format**

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

**Constraints**

$1 \leq a \leq 10^{10}$   
 $1 \leq b \leq 10^{10}$

**Congratulations**

You solved this challenge. Would you like to challenge your friends? [f](#) [t](#) [in](#) [Next Challenge](#)

**Test case 0** **Test case 1**

**Compiler Message**

Success

**Input (stdin)** [Download](#)

1 3  
2 2

**Expected Output** [Download](#)

1 5  
2 1  
3 6

## Problem - 3 Division

**HackerRank** Prepare > Python > Introduction > Python: Division

Check the Tutorial tab to know learn about division operators.

**Task**  
The provided code stub reads two integers,  $a$  and  $b$ , from STDIN. Add logic to print two lines. The first line should contain the result of integer division,  $a // b$ . The second line should contain the result of float division,  $a / b$ . No rounding or formatting is necessary.

**Example**  
 $a = 3$   
 $b = 5$   
• The result of the integer division  $3 // 5 = 0$ .  
• The result of the float division is  $3 / 5 = 0.6$ .

Print:

```
0
0.6
```

**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.

**Congratulations**  
You solved this challenge. Would you like to challenge your friends? [f](#) [t](#) [in](#) [Next Challenge](#)

Test case 0 ✓  
Test case 1 ✓

Compiler Message  
Success

Input (stdin) [Download](#)  
4  
3

Expected Output [Download](#)  
1  
1.333333333333

## Problem - 4 Loops

**HackerRank** Prepare > Python > Introduction > Loops

Check Tutorial tab to know how to solve.

**Task**  
The provided code stub reads and integer,  $n$ , from STDIN. For all non-negative integers  $i < n$ , print  $i^2$ .

**Example**  
 $n = 3$   
The list of non-negative integers that are less than  $n = 3$  is  $[0, 1, 2]$ . Print the square of each number on a separate line.

```
0
1
4
```

**Input Format**  
The first and only line contains the integer,  $n$ .

**Constraints**  
 $1 \leq n \leq 20$

**Output Format**  
Print  $n$  lines, one corresponding to each  $i$ .

**Sample Input 0**

**Congratulations**  
You solved this challenge. Would you like to challenge your friends? [f](#) [t](#) [in](#) [Next Challenge](#)

Test case 0 ✓  
Test case 1 ✓

Compiler Message  
Success

Input (stdin) [Download](#)  
5

Expected Output [Download](#)  
0  
1  
4  
9  
16

## Problem - 5 Function

# HackerRank

Prepare > Python > Introduction > Write a function

Exit Full Screen View

Sign in with email or GitHub

Test against custom input

Problem

Submissions

Leaderboard

Discussions

An extra day is added to the calendar almost every four years as February 29, and the day is called a leap day. It corrects the calendar for the fact that our planet takes approximately 365.25 days to orbit the sun. A leap year contains a leap day.

In the Gregorian calendar, three conditions are used to identify leap years:

- The year can be evenly divided by 4, is a leap year, unless:
  - The year can be evenly divided by 100, it is NOT a leap year, unless:
    - The year is also evenly divisible by 400. Then it is a leap year.

This means that in the Gregorian calendar, the years 2000 and 2400 are leap years, while 1800, 1900, 2100, 2200, 2300 and 2500 are NOT leap years. [Source](#)

**Task**

Given a year, determine whether it is a leap year. If it is a leap year, return the Boolean `True`, otherwise return `False`.

Note that the code stub provided reads from STDIN and passes arguments to the `is_leap` function. It is only necessary to complete the `is_leap` function.

**Input Format**

Read `year`, the year to test.

**Constraints**

$1900 \leq \text{year} \leq 10^5$

**Output Format**

The function must return a Boolean value (`True/False`). Output is handled by the

Test case 0

Test case 1

Test case 2

Test case 3

Test case 4

Test case 5

Compiler Message

Success

Hidden Test Case

Unlock this testcase for 5 hacks.

Unlock

Next Challenge