# Computer Science I — CSCI141
# Conditionals — Homework

08/29/2016

## 1 Task 1

Write and test a function that determines if the larger of two integers is evenly divisible by the smaller integer.

### 1.1 Requirements

- Name your function `divisible`. Your function must take two parameters. You may assume that the inputs provided to your function are integer values. That is, you do not need to do any error-checking to confirm that they are integer values.
- Your function should print one of the following messages:
  - If either (or both) of the integers is not positive, output the message: `Inputs must be positive integers!`.
  - If the integers are both positive, and are equal, output the message: `Those integers are equal: x`, where `x` is the integer value input to the function.
  - If the integers are both positive, are not equal, and the larger integer is evenly divisible by the smaller integer, output the message: `a is evenly divisible by b`, where `a` is the larger of the input integers, and `b` is the smaller of the input integers.
  - If the integers are both positive, are not equal, and the larger integer is not evenly divisible by the smaller integer, output the message: `a is not evenly divisible by b`, where `a` is the larger of the input integers, and `b` is the smaller of the input integers.

  Your messages need not exactly match the messages above, but they must provide the same information. Note that the input integers are not necessarily provided in a particular order. The larger integer may be either the first or the second value provided to your function.
- Write a test function called `test_divisible` that tests your `divisible` function by calling it several times with different inputs. You should have at least six (and perhaps more) tests. Choose inputs that test different aspects of your function. For example, you may have one test `divisible(5,5)`, that checks that the function works properly when the inputs are both positive and the same value. Once you have this test, you do not need another test `divisible(8,8)`, as this checks the same thing. You should not use any tests involving inputs that are not integers.

### 1.2 Examples

If you were to call your `divisible` function from the interactive console:

```
>>> divisible(6,4)
6 is not evenly divisible by 4
>>> divisible(-4, 9)
Inputs must be positive integers!
```

## 2 Task 2

Write and test a function that determines if three integer inputs could represent the lengths of the sides of a triangle. (Look up the triangle inequality if necessary.)

### 2.1 Requirements

- Name your function `triangle`. Your function must take three parameters. You may assume that the inputs provided to your function are integer values. That is, you do not need to do any error-checking to confirm that they are integer values.
- Your function should print one of the following messages:
  - If any of the integers is not positive, output the message: `Triangles require sides of positive length!`.
  - If the integers are all positive, and can represent the lengths of the sides of a triangle, output the message: `a, b and c can form a triangle`, where `a`, `b` and `c` are the integer values input to the function.
  - If the integers are all positive, and can not represent the lengths of the sides of a triangle, output the message: `a, b and c can not form a triangle`, where `a`, `b` and `c` are the integer values input to the function.

  Your messages need not exactly match the messages above, but they must provide the same information. Note that the input integers are not necessarily provided in a particular order. The largest integer may be any of the three provided values.
- The degenerate case (when the integers are all positive and the sum of two of the integers is exactly equal to the third integer) counts as a case of inputs that can form a triangle.
- Write a test function called `test_triangle` that tests your `triangle` function by calling it several times with different inputs. You should have at least six (and perhaps more) tests. You should not use any tests involving inputs that are not integers.

### 2.2 Examples

If you were to call your `triangle` function from the interactive console:

```
>>> triangle(-3, 6, 5)
Triangles require sides of positive length!
>>> triangle(4,6,5)
4, 6 and 5 can form a triangle
```

# 3 Programming Tips

## 3.1 Modulo

Python provides an operator that computes the remainder when one number is divided by another. It is very useful - we will see it again in this course! It is called the modulo operator. From the interactive console:

```
>>> 8 % 3
2
>>> 4 % 11
4
>>> 60 % 5
0
```

## 3.2 Simple `print` Formatting

Normally, if you execute a print statement like

```
print( 5, "x", 2 )
```

your output looks like

```
5 x 2
```

If you want there to be no spaces between the values printed, add a value for a special, named parameter `sep`:

```
print( 5, "x", 2, sep="" )
```

# 4 Last Lines

At the bottom of your program, include calls to your test functions (perhaps from inside a `main` function) so that when you run your program, it proceeds to call your test functions and output the results of the multiple tests you have.

Note that you are not prompting for input from the user anywhere in this program.

### 4.1 Grading

- 35%: Correct functionality of the `divisible` function.
- 35%: Correct functionality of the `triangle` function.
- 10%: Sufficient testing provided in `test_divisible`.
- 10%: Sufficient testing provided in `test_triangle`.
- 5%: Each function has a *docstring* containing a sentence describing its purpose. This documentation helps others understand how they may reuse the function. An example is provided on the Course Resources webpage:

  `http://www.cs.rit.edu/~csci141/Docs/style-example-py.txt`
- 5% The program is in the correct, standard style, starting with a *docstring* for the whole file. This program file docstring must contain your *full* name.

### 4.2 Submission

Put your program code with the appropriate documentation into a file called `conditionals.py` and submit that file to the MyCourses dropbox for this assignment.