### Worksheet for Lab #3 Ex2: Square-free Integer

http://www.comp.nus.edu.sg/~cs1010/labs/2017s1/lab3/controlstructures.html

#### **Task Statement**

Read 4 positive integers *lower1*, *upper1*, *lower2*, *upper2*, determine which of the 2 ranges [*lower1*, *upper1*] and [*lower2*, *upper2*] contains more square-free integers, and report the number of square-free integers in the range that has more square-free integers.

A square-free integer is a positive integer not divisible by any square number except 1.

For this exercise, let's try a bottom-up design.

### Step 1

We should have a function that takes in an integer and determines if it is square-free or not. Let's call this function **is\_square\_free()**. Complete the table about **is\_square\_free()** below.

Dotum tuno	Parameter		Precondition	
Return type	Туре	Name	Precondition	

## **Step 2**Write out your algorithm for **is\_square\_free()**. What control structures does it use?

Algorithm:	
is_square_free(	) {
}	

This algorithm uses (circle all appropriate answers): Sequence / Selection / Repetition

St	e	p	3

You may practise **Incremental Coding** (which Aaron has demonstrated in his week 4 lecture) here. The correctness of your final program depends on the correctness of **is\_square\_free()** function. So it is worthwhile checking that the function works before you proceed.

How do you think you can test the is\_square\_free() function?

### Step 4

After you have ensured that **is\_square\_free()** is perfect, how do you use it to solve the task? Since you are going to do the same thing (count the number of square-free integers) for TWO ranges, you should write a function for it. Let's call it **count\_square\_free()**.

Complete the table about **count\_square\_free()** below.

Return type	Parameters (types and names)	Precondition	

# Step 5 Write out your algorithm for count\_square\_free().

Algorithm:	
count_square_free(	_) {
}	

### Step 6

Complete your main() function and test your program thoroughly before you submit it to CodeCrunch.