# COMP108 Data Structures and Algorithms

## Week 02 Tutorial Exercises

Due: 07 February 2025, 5:00pm

(Late submission accepted until Monday 9:00am)

**Information**

- Handwrite/typeset your answers and make it into a single pdf file. See this guide on how to scan documents to pdf.
- Submission: Submit a file named **COMP108W02.pdf** on Canvas
  **Late submission is only accepted until Monday 9:00am.**
- Submission of lab/tutorial exercises contributes to 10% of the overall module mark. Submission is marked on a pass/fail basis - you will get full marks for submitting a *reasonable attempt*.
- Individual feedback will not be given, but solutions will be posted promptly after the deadline has passed.
- These exercises aim to give you practices on the materials taught during lectures and provide guidance towards preparation of examination.
- Relevant lectures: **Lectures 1-3**
- Turn to next page for the questions.

1. Consider the following algorithm.

```
// Assume n is a given integer being power of 2
count ← 0
x ← n
while x > 1 do
begin
    x ← x/2
    count ← count + 1
end
output count
```

   (a) Give the **trace table** and the **output** of the above algorithm when $n = 32$.

   (b) In general, how many times the while loop is executed for input $n$ being a positive power of 2 (e.g., when $n = 2, 4, 8, 16, 32, 64, \ldots$)?

2. Write a pseudo code of a while-loop to find the sum of all multiples of 3 between $x$ and $y$ inclusively. You can assume that $0 < x \leq y$. For example, if $x = 4$ and $y = 12$, then your pseudo code should output 27 (which equals to $6 + 9 + 12$).

3. A *prime number* is a number that can be divisible by 1 and itself only. Write a pseudo code of an algorithm to determine if a positive integer $x > 1$ is a prime number or not.

   Hints: (1) We can use a loop to check for each integer $i$ smaller than $x$ whether $x$ is divisible by $i$. (2) If we want to make it quicker, we can stop earlier, the question is when should we stop the loop. (3) You can use the % operator to find remainder, e.g., $a\%b$ gives the remainder of $a$ divided by $b$.