COMP108 Data Structures and Algorithms

Week 02 Tutorial Exercises

Due: 10 February 2023, 5:00pm

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 $n = 32$.

Suppose n = 32

Iteration	x before	x after	count
before loop		32	0
1	32	16	1
2	16	8	2
3	8	4	3
4	4	2	4
5	2	1	5

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

$\log_2(n)$ times.

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 $6 + 9 + 12$).

```
sum <- 0
i <- x
while (i <= y)
begin
    if (i % 3 == 0) then
        sum <- sum + i
    i <- i + 1
end
output sum
```

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.

```
isPrime <- True
i <- 2
while (i <= √x) && isPrime
begin
        if (x % i == 0) then
                isPrime <- False
        i <- i + 1
end
output isPrime
```