Question 1. Comparing Algorithms

(a) In this lab, for all three algorithms you will
(b) write the pseudo code. (Must follow the notations and conventions used in today's Lecture)
(c) determine the worst case time complexity.

Problem statement
Find the largest distance between any two even integers in an integer array.

Algorithm 1.

Create a new array consisting of even numbers only. Then use nested loops to solve the problem using
the newly created array of even numbers only.

```
Algorithm findLargestDistance(A, n)
  Input Array A of n integers
  Output largestDistance between 2 even elements of A

  numsEven <- new Array
  for i <- 0 to n - 1 do
    if(A[i]%2 == 0)
      numsEven[x] <- A[i]

  largestDistance <- 0
  for i <- 0 to n - 2 do
    for j <- 1 to n - 1 do
      distance <- numsEven[i] - numsEvent[j]
      if(numsEven[j] > numsEvent[i])
        distance <- numsEven[j] - numsEvent[i]
      if(largestDistance < distance)
        largestDistance <- distance

  return largestDistance
```

Time complexity: $O(n) + O(n^2) \rightarrow O(n^2)$

Algorithm 2.

Use a nested loop to solve the problem without creating an extra array.

```
Algorithm findLargestDistance(A, n)
  Input Array A of n integers
  Output largestDistance between 2 even elements of A

  largestDistance <- 0
  for i <- 0 to n - 2 do
    if(A[i]%2 != 0)
       continue
    for j <- 1 to n - 1 do
      if(A[j]%2 != 0)
       continue
      distance <- A[i] - A[j]
      if(A[j] > A[i])
        distance <- A[j] - A[i]
      if(largestDistance < distance)
        largestDistance <- distance

  return largestDistance
```

Time complexity: $O(n^2)$

Algorithm 3.

Use one loop. Find max and min of even integers. Compute max – min.

```
Algorithm findLargestDistance(A, n)
  Input Array A of n integers
  Output largestDistance between 2 even elements of A

  max <- Integer.MIN_VALUE
  min <- Integer.MAX_VALUE
  for i <- 0 to n - 1 do
    if(A[i]%2 == 0)
       if(max < A[i])
         max <- A[i]
```

```
    if(min > A[i])
      min <- A[i]

largestDistance <- max - min

return largestDistance
```

Time complexity: O(n)

Question 2.

Consider the following functions to determine the relationships that exist among the complexity classes they belong: 10, 1, n^3 , $n^{1/3}$, log(log(n)), n^2, $n^{1/2}$, logn , $\log n^n$, n^k (k > 3) , $n^{1/k}$ (k > 3), nlogn, ln(n), 2^n , 3^n , n^n , $n^{1/2}$ logn, $n^{1/3}$ logn, n!.

10, 1	***Θ***(1)
log(log(n))	***Θ***(log(log(n)))
log(n), ln(n)	***Θ***(log(n))
$n^{1/2}$, $n^{1/3}$, $n^{1/k}$ (k > 3)	***Θ***($n^{1/k}$) (0 < k < 1)
$n^{1/2}$ logn, $n^{1/3}$ logn	***Θ***($n^{1/k}$logn) (0 < k < 1)
nlogn, $\log n^n$	***Θ***(nlog(n))
n^2	***Θ***(n^2)
n^3	***Θ***(n^3)
n^k (k > 3), n^n	***Θ***(n^k)
2^n	***Θ***(2^n)
3^n	***Θ***(3^n)
n!	***Θ***(n!)