Quiz 1

Deadline	Friday, 12 June 2020 at 11:59PM
Latest Submission	*no submission yet*
Maximum Mark	4

Question 1 (1 mark)

The following function aims to compute x^n :

```
long pow(int x, uint n) {
{
   long raised = x;
   for (int i = 0; i <= n; i++) raised *= i;
   return raised;
}
```

What does it actually return?

(a) ○	Always 0
(b) ○	Always 1
(c) ○	x^{n-1}
(d) ○	x^n
(e) ○	x^{n+1}
(f) ○	A random number

Question 2 (1 mark)

How many times will the for-loop condition (`i <= n`) be checked in this function?

```
long factorial(int n)
{
   long product = 1;
   for (int i = 1; i <= n; i++)
      product *= i;
   return product;
}
```

(a) ○	Once
(b) ○	*n-1* times
(c) ○	*n* times
(d) ○	*n+1* times
(e) ○	*n!* times
(f) ○	None of the above

Question 3 (1 mark)

Consider the following function:

```
int myFunction(int n)
{
   int i = 0;
   while (n > 0) {
      i++;
      n = n/2;
   }
   return i;
}
```

What value is returned from the function call myFunction(100) ?

Enter number

Question 4 (1 mark)

Arrange these algorithm complexity values in order from *best* to *worst*.

Best = the one you'd most like to achieve. Worst = the one you'd most like to avoid.

Note: (x^y) means x^y.

✥ O(n^2)

- ✥ O(1)
- ✥ O(log n)
- ✥ O(n)
- ✥ O(2^n)

✔ Submit