**COMP3121 Assignment 2 – Question 1**

**1)** In this question, we have two positive integers – *M* and *n* and we are required to find **Mn** using only **O(log(n))** many multiplications.

We can employ the use of binary to help simplify our method, specifically writing our *n* in binary i.e.

This essentially means that we work out for all *k* where . This can be seen to involve at most multiplications. By computing all of this will require at most multiplications. This can be done by repeated squaring where the number of digits within n is proportional to multiplications at most. Put more simply, as an example if we had , we can convert 17 into binary which would be 10001, then by reading from right to left (or from the least-significant bit to most-significant bit) we would find that it would be:

These final values to be multiplied are referred to as the successive squares (16 and 1) and we needed at most 4 “squarings” (as we know that the left-most digit is in the position and the right-most number is just 5 as hence we do not need to include that as a “squaring”). Intuitively, from above we recognise that the number of multiplications required would be at most *m* bits required to represent our exponent and hence would be a maximum of m – 1 multiplications. Putting it all together, the number of bits required to represent our positive integer *n* would be: . Note, however, that in most cases it is simply enough to find the number of bits from doing and then computing the required multiplications from there. In summation, this would lead to an overall maximum of multiplications, as required.

**End of Solution**