

Name: Minh Khai Tran
zID: z5168080

Question 1:

- The main idea of the algorithm to solve this question: divide-and-conquer by computing $M^{\frac{n}{2}}$ and storing it.
- When we compute $M^{\frac{n}{2}}$ we mainly have 3 cases:
 - If $n = 0$: we return 1
 - If $\frac{n}{2}$ is an even number: $M^n = M^{\frac{n}{2}} * M^{\frac{n}{2}}$ (count as one multiplication)
 - If $\frac{n}{2}$ is an odd number: $M^n = M^{\lfloor \frac{n}{2} \rfloor} * M^{\lfloor \frac{n}{2} \rfloor} * M$ (count as one multiplication)
- The algorithm is called (recursively) as many times as the original n is divisible by 2. Each time we calling the algorithm, we compute one multiplication. Let define k is the number of times we call the algorithm. So that we have:

$$2^k = y$$

- There for, $k = \log_2 y$. Hence the greatest number of multiplications we have is:
 $O(\log n)$