Q1:

Given： positive integers M and n

Output： a methods with O(logn) multiplications

Solution:

According to exponential product rules, $a^n * a^m = a^{m+n}$, then we can rewrite our n in binary as n = $\frac{n}{2}+\frac{n}{2}$, then M^n is equivalent to $M^n = M^{\frac{n}{2}} * M^{\frac{n}{2}}$, repeat this process until n = 1 and stop. The Pseudocode shown as

```
Input: integer m and n;
Output: the n power of m
Function Rescurssion(m,n):
    If n=0 then
        Return 1;
    End if
    If n%2=0 then
        Return square(square(m,n/2))
    End if
    Else
        Return m*square(square(m,n/2))
    End if
End function
```

The height of this Recurrance tree with in the range 0 to $\lfloor \log_2 n \rfloor$, So that, Totally there will be at most 2*$\lfloor \log_2 n \rfloor$ multiplications. The square multiplication can be solved by bits shift in one time each. Then there will have O(logn) times multiplications.