HW4 Dakota Krogmeier

1. Key Idea: The key idea will be to count backwards starting at “power” or “n” until we reach zero, multiplying “m” by its base value until n == 0.

Algorithm Exponentiate(m, n)

Input: Any natural number greater than 0 for M and N.

Output: A positive natural number where M is risen to the power of N.

Process:

Exponentiate(m,n):

If y == 1:

Return x

If y !=1:

Return x \* exponentiate(m, n-1)

B. Key Idea: Idea is to recursively subtract the divisor from the dividend until we reach the point where x < y. We then return the number of loops, reflected by adding 1.

Algorithm Quotient(dividend,divisor):

Input: Any whole number X and any natural number Y.

Output: The quotient of dividend + divisor.

Process:

If (x<y):

Return 0;

Else:

Return 1 + Quotient(x-y, x);

C. Key Idea:

Algorithm Mod(A,B):

Input:

Output:

Process: