1) (10pts) ANL (Algorithm Analysis)

Consider the recursive function `diminish` shown below:

```
double diminish(int m, int n){
    if (n == 0)
        return m;
    return  1.0/2*diminish(m,n-1)
}
```

(a) (3 pts) Let T(n) represent the run time of the function `diminish`. Write a recurrence relation that T(n) satisfies.

$$T(n) = T(n-1) + O(1)$$

Grading: 1 pt for each component. Note that the last component may be any positive integer constant and still receive full credit.

(b) (6 pts) Using the iteration method, determine a closed-form solution (Big-Oh bound) for T(n).

$$T(n) = T(n-1) + O(1)$$
$$T(n) = T(n-2) + O(1) + O(1)$$
$$T(n) = T(n-3) + O(1) + O(1) + O(1)$$
$$T(n) = T(n-k) + kO(1)$$

Plugging in k = (n-1), we find:

$$T(n) = T(n-(n-1)) + (n-1)O(1)$$
$$T(n) = T(1) + (n-1)O(1)$$
$$T(n) = 1 + (n-1)O(1)$$
$$T(n) = O(n)$$

Grading: 3 pts for couple iterations and generalization, 3 pts for rest - allow with or without Big-Oh notation. Give full credit to any function that is O(n), most likely ones are n, 2n, 3n, with a plus or minus 1, potentially.

(c) (1 pt) In terms of the values of `m` and `n`, respectively, what does the function call `diminish(m,n)` return? (You may assume that `m` and `n` are both positive.)

$$diminish(m,n) = \frac{m}{2^n}$$

Grading: 1 pt for correct answer, 0 otherwise