Problem 4

Algorithm iterativeFibonacci (n)	*Count of operations*
Input: non negative number *n*	
Output: array of Fibonacci numbers [0….n]	
if (n = 0 | n = 1) then return n	*4*
initialize array fib [n + 1]	*2 (n + 1)*
fib [0] = 0	*2*
fib [1] = 1	*2*
for (i = 2 ; i ≤ n ; i++) do	*1 + n + 1 + 2*
fib [i] = fib [i – 1] + fib [i – 2]	*n(2 + 2 + 1 + 2)*
return fib	*1*

$f(n) = 4 + 2(n + 1) + 4 + 4 + n + 7n + 1 = 10n + 15$

To know the asymptotic running time:

$$\lim_{n\to\infty} \frac{10n + 15}{n} = \lim_{n\to\infty} \frac{10 + \frac{15}{n}}{1} = 10 \to f(n)\ is\ O(n).$$

Proof that the algorithm is correct:

1. It has two base cases: $fib[0]$ and $fib[1]$, and they yield the correct Fibonacci's values. Also, the loop in the algorithm starts with $fib[2]$. According to the algorithm, it is calculated using the previous two numbers, which are $fib[1]$ and $fib[0]$, the base cases.
2. The loop invariant is $fib[i]$. assuming the call to $fib[i]$ is true, we try to show that $fib[i + 1]$ also holds. From the pseudocode above: $fib[i + 1] = fib[i] + fib[i - 1]$. Since $fib[i]$ and $fib[i - 1]$ are the two numbers preceding $fib[i + 1]$, then the code is correct according to the definition of Fibonacci's series.