Dynamic Programming to compute $F(n)$

Let $F(n)$ denote the nth Fibonacci number:
Input: n, positive integer
Output: F(n)

Fill in an array, FIB[1...n] as follows:

- $FIB[1] := 1$
- $FIB[2] := 1$
- For $i := 3$ up to n do:
    - $FIB[i] := FIB[i-1] + FIB[i-2]$
- Return $FIB[n]$

Recall we analyzed the running time and showed it was $O(n)$ to compute $FIB[n]$.

Let's prove that $FIB[n]$ is the same as $F(n)$, the nth Fibonacci number.