1. A seguir estão comentadas ambas as funções comparando suas implementações.

int fibonacci\_01::fib(int n) {  
 /\*  
 \* Aqui utiliza-se função recursiva, em que são gerados "espaços" dentro de "espaços" análogo a  
 \* uma busca em profundidade. Quando a codição n<=1 é satisfeita, a função chamada dentro do último  
 \* "espaço" retorna um valor e assim as demais chamadas retornam valores sucessivamente obedecendo  
 \* a soma fib(n-1) + fib(n-2). o Valor final retornado é o n-ésimo termo da sequência de fibonacci.  
 \*/  
 if (n<=1){  
 return n;  
 }  
 return fib(n-1) + fib(n-2);  
}  
int fibonacci\_02::fib(int n) {  
 //int f[] = new int[n+2]; declaração original incorreta  
 /\*  
 \* Nesse caso, primero são gerados todos os valores da sequência de fibonacci até o n-ésimo item,  
 \* retornando-o no final  
 \*/  
 int \*f = new int(n+2);  
 int i;  
 f[0] = 0;// o termo zero é atribuído  
 f[1] = 1;// o primeiro termo é atribuído  
 for (i=2;i<=n;i++){  
 f[i] = f[i-1] + f[i-2];// com os primeiros termos inicializados, os demais são gerados até o n-ésimo  
 }  
 return f[n]; //retorna o n-ésimo termo