

# FIBONACCI

## Explicação do problema:

A sequência de Fibonacci pode ser definido como uma série de números em que são encontrados na primeira e na segunda posição, respectivamente, o 0 e o 1. Os demais números desta sequência se dão pela soma dos dois números anteriores. Por exemplo, o décimo número dessa sequência é composto pela soma do nono com o oitavo número da série, que por sua vez, também são formados pela soma dos dois anteriores.

Desta forma, chega-se a uma sequência: 0,1,1,2,3,5,8,13,21,34,55... Em que para se chegar a “3”, precisa-se somar “2” e “1” e assim por diante mantendo a regra da soma das duas posições anteriores.

$$\text{Fib}(i) = \text{Fib}(i-1) + \text{Fib}(i-2);$$

## Explicação da solução:

A solução utilizada para o problema de Fibonacci foi a divisão deste em subproblemas, sendo o primeiro deles a questão de não ser conhecido os dois primeiros valores da sequência, de forma a impedir o desenvolvimento do restante da sequência. Dessa forma, foram criadas duas variáveis em que guardamos os dois valores que serão somados para a confecção do valor da terceira posição da sequência.

Chamadas de fib0 e fib1 essas variáveis são **inicialmente** iguais aos dois primeiros valores de Fibonacci, 0 e 1, para que dessa forma, um outro subproblema possa ser solucionado: a soma que gera o próximo número da sequência. Dessa forma, cria-se mais uma variável, chamada de fib, em que soma-se os valores encontrados em fib0 e fib1, gerando o valor da terceira posição.

Ao unir a solução dos dois subproblemas apresentados, podemos encontrar o restante dos valores presentes em Fibonacci, considerando que, após o encontro do valor da terceira posição, movimentamos fib0, fib1 e fib uma “casa” a frente, para que fib0 esteja na posição 2, fib1 na posição 3 (a qual era fib até que esse valor tivesse sido encontrado pela soma), fib na próxima posição a ter seu valor descoberto (posição 4 no exemplo) e assim por diante.

## Implementação:

```
1  #include <stdio.h>
2
3  int fibonacci(int n){
4      int fib;
5      int fib0 = 0;
6      int fib1 = 1;
7
8      if(n<=1)
9          return n;
10
11     for(int i=2; i<=n; i++){
12         fib = fib0 + fib1;
13         fib0 = fib1;
14         fib1 = fib;
15     }
16     return fib;
17 }
18
19 int main(void) {
20     int pos = 0, resultado, v;
21
22     printf("coloque o numero da posicao desejada: ");
23     scanf("%d", &pos);
24
25     resultado = fibonacci(pos);
26
27     printf("valor da posicao: %d", resultado);
28
29     return 0;
30 }
```

```
➤ ./main
coloque o numero da posição desejada: 7
valor da posicao: 13
```

### Análise assintótica:

<code>int fibonacci(int n){</code>	VEZES
<code>int fib;</code>	1
<code>int fib0 = 0;</code>	1
<code>int fib1 = 1;</code>	1
<code>if(n&lt;=1)</code>	1
<code>return n;</code>	
<code>for(int i=2; i&lt;=n; i++){</code>	n
<code>fib = fib0 + fib1;</code>	(n-1)
<code>fib0 = fib1;</code>	(n-1)
<code>fib1 = fib;</code>	(n-1)
<code>}</code>	
<code>return fib;</code>	1
<code>}</code>	

---

$$T(n) = 5 + n + 3(n-1)$$

$$T(n) = 5 + n + 3n - 3$$

$$T(n) = 4n + 2$$

$$\text{Logo, } O(n)$$