



2 RECURSIVIDADE

9. Utilizando recursividade, calcule a soma de todos os valores em um vetor.
10. Utilizando recursividade, encontre o maior elemento em um vetor. **Dica:** encontre o maior elemento no subconjunto que contém todos elementos, exceto o último elemento. Então, compare o maior elemento desse subconjunto com o valor do último elemento.
11. Escreva uma função recursiva que retorne o comprimento de uma determinada string.
12. Crie uma função recursiva que inverta a ordem dos caracteres de uma string. Por exemplo, se a string é “**Hello!**”, a função transformará a string em “**!olleH**”.
13. Faça uma função recursiva que verifica se uma string t é uma substring da string s . Por exemplo, a string “**lele**” é uma substring da string “**Paralelepípedo**”. **Dica:** se o texto começa com a string que você quer encontrar, então está feito. Se não, analise a frase que voce obtém removendo o primeiro caractere.
14. Crie uma função recursiva que retorna a posição inicial do primeiro substring da string s que seja igual a t .
15. De acordo com a seguinte definição

$$MDC(n, m) = \begin{cases} m & \text{se } m \leq n \text{ e } n \bmod m = 0 \\ MDC(m, n) & \text{se } n < m \\ MDC(m, n \bmod m) & \text{caso contrário} \end{cases}$$

Dê o que se pede:

- (a) apresente uma função recursiva para retornar o maior divisor comum de dois inteiros n e m .
 - (b) apresente a árvore de recursão para $n = 12$ e $m = 20$ e apresente o valor final.
16. Seja a função definida conforme descrita a seguir

```
float divisaoRec(int num, int den){  
    if (num < den)  
        return 0;  
    else  
        return divisaoRec(num - den, den) + 1;  
}
```

Dê o que se pede:

- (a) Apresente o modelo matemático para essa função.
- (b) Apresente a árvore de recursão para essa função com $num = 10$ e $den = 2$.