

Listas de Exercícios 3 – Recursão

1. Faça um algoritmo de uma função *string* recursiva que receba como parâmetro uma *string* e devolva uma *string* na qual todas as letras sejam minúsculas.
2. Faça um algoritmo de uma função inteira recursiva que receba um parâmetro inteiro e retorne um inteiro que seja o reverso (de trás para frente) do inteiro passado como parâmetro.
3. A sequência de Fibonacci é dada pela relação de recorrência $F_0 = 1, F_1 = 1, F_n = F_{n-1} + F_{n-2}$, para $n \geq 2$. Então, encontre a relação de recorrência que dá o número de chamadas da função algorítmica que calcula a citada função. Em seguida implemente o algoritmo da função inteira recursiva que implementa a relação de recorrência encontrada por você.
4. Faça o algoritmo de uma função inteira recursiva que calcule o *máximo divisor comum* (MDC) de dois números inteiros estritamente positivos transmitidos a ela como parâmetros. Para o cálculo do MDC utilize o algoritmo de Euclides: $\text{MDC}(a, a) = a$, e sendo $a > b$, positivos, e r o resto da divisão inteira de a por b , temos que $\text{MDC}(a, b) = \text{MDC}(b, r)$. Retorne 0 como *flag* em caso de erro.
5. Faça o algoritmo de uma função inteira recursiva que receba dois parâmetros inteiros positivos a e b (sendo que b deve ter apenas 1 dígito) e retorne a quantidade de vezes que o dígito b aparece dentro do número a . Caso seja necessário, utilize o valor -1 como *flag* indicador de erro dos valores de entrada.
6. Faça um algoritmo de uma função *string* recursiva que receba como parâmetros um vetor de *strings* e o seu tamanho, e retorne a *menor string* (lexicograficamente, alfabeticamente falando). Considere que letras maiúsculas e minúsculas são diferentes.

Exemplo:

parâmetro = [“José” , “Eliana” , “Mariana” , “Lucas” , “Thiago”] , 5
retorno = “Eliana”

7. Faça o algoritmo de uma função inteira recursiva que receba um valor inteiro n como argumento e escreva, na saída, os valores dos somatórios $\left\{ \sum_{i=1}^k | 0 \leq k \leq n \right\}$ para cada valor de k no intervalo, do menor para o maior. O valor inteiro de retorno da função é o do próprio somatório descrito. Se possível, faça com que as vírgulas sejam corretamente escritas pela função apenas entre os valores inteiros.

Exemplo: Se for passado o valor 5 à função, serão escritos os valores $\{0, 1, 3, 6, 10, 15\}$, pois esses valores são obtidos das somas $\{0, 0+1, 0+1+2, 0+1+2+3, 0+1+2+3+4, 0+1+2+3+4+5\}$.

8. Faça o algoritmo de uma função inteira recursiva que receba um número inteiro representando um valor numérico na base decimal e devolva o inteiro que contenha o mesmo valor escrito na base octal.
9. Faça o algoritmo de uma função *string* recursiva que receba um número inteiro representando um valor numérico na base decimal e devolva a *string* que contenha o mesmo valor escrito na base hexadecimal.

10. Faça o algoritmo de uma função inteira recursiva que receba uma *string* representando um valor numérico na base decimal e devolva o inteiro que contenha o mesmo valor escrito na base binária.
11. Faça o algoritmo de uma função inteira recursiva que receba um inteiro representando um valor numérico na base binária e devolva o inteiro que contenha o mesmo valor escrito na base decimal. Em caso de erro, retorne -999999.
12. Faça o algoritmo de uma função inteira recursiva que receba uma *string* representando um valor numérico na base octal e devolva o inteiro que contenha o mesmo valor escrito na base decimal. Em caso de erro, retorne -999999.
13. Faça o algoritmo de uma função *string* recursiva que receba uma *string* representando um valor numérico na base hexadecimal e devolva a *string* que contenha o mesmo valor escrito na base decimal. Em caso de erro, retorne a *string* vazia.
14. Faça o algoritmo de uma função real recursiva que retorne o maior valor entre vários valores lidos do usuário. A função deve receber um parâmetro real que funcione como *flag* de parada da leitura, ou seja, esse valor passado como parâmetro serve para testar se o valor digitado pelo usuário é um valor válido ou um valor que funcione como parada e que não deva ser utilizado no cômputo do maior valor digitado. Note que se o usuário não digitar nenhum valor diferente do *flag*, é o próprio *flag* que deve ser retornado, caso contrário, o maior valor entre os digitados deve ser retornado.
15. Faça o algoritmo de uma função inteira recursiva que receba dois parâmetros: uma *string* e um caracter, e retorne a última posição na qual o caracter (segundo parâmetro) aparece dentro da *string* (primeiro parâmetro). A primeira posição da *string* é 1 e caso o caracter não apareça na *string* a função deve retornar o valor 0.
16. Analise a função abaixo e diga que valor é impresso na saída quando for executada a chamada $F(a, 6)$ sendo $a = \{0, 1, 2, 3, 4, 5\}$.

tipos

```
Vet = vetor[0..5] de inteiro
função F(ref a: Vet; n: inteiro): inteiro
início
  se n ≤ 0 então
    retorno 1
  fim se
  retorno a[n - 1] × F(a, n - 2) + 1
fim função
```

17. Qual o valor de $X(5)$? Demonstre sua solução.

```
função X(n: inteiro): inteiro
início
  se n=1 ou n=2 então
    retorno n
  senão
    retorno X(n - 1) + n * X(n - 2)
  fim se
fim
```