

Lista 4 de Exercícios de Introdução à Programação – INF 1025

Esta lista refere-se a exercícios de recursividade. Para cada exercício você deve fazer a função indicada juntamente com o programa que lê os parâmetros, chama a função e exibe o resultado.

- 1) Crie uma função recursiva que conte quantas vezes os algarismos 3 e 4 aparecem no número.

ex: nº recebido: 2394284 retorno da função: 3

- 2) Crie uma função recursiva que receba um número e retorne-o como *string* (mesma tarefa da função *str()*).

- 3) Crie uma função recursiva que receba dois números, com mesma quantidade de algarismos, e retorne TRUE se todos os algarismos posicionalmente do 1º número são menores que os do 2º ou FALSE, caso contrário.

ex: algMenores(1234 , 2456) --> True

algMenores(1234 , 2436) --> False

- 4) Crie uma função recursiva que retorne um número com os algarismos invertidos.

- 5) Crie uma função recursiva que receba um número e exiba-o conforme exemplo.

Ex Nº1234

Saída:

1

12

123

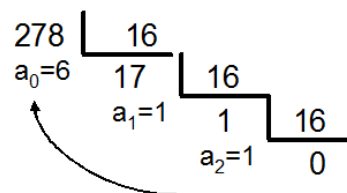
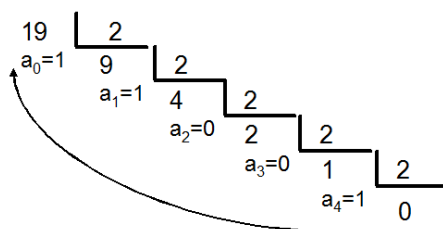
1234

- 6) Crie uma função recursiva que receba um número N, na base 10, uma base B e retorne o número N na base B.

Dica: O número decimal deve ser dividido sucessivas vezes pela base, o resto de cada divisão ocupará sucessivamente as posições de ordem 0, 1, 2 e assim por diante, até que o resto da última divisão (que resulta em quociente 0) ocupe a posição de mais alta ordem.

Ex: $19_{10} == 10011_2$

$278_{10} == 116_{16}$



- 7) Crie uma função recursiva que receba um número N, na base B, a base B e retorne o número N na base 10.

Ex: 116 na base 16 = $6 \times 16^0 + 1 \times 16^1 + 1 \times 16^2 = 278$

10011 na base 2 = $1 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + 0 \times 2^3 + 1 \times 2^4 = 19$

- 8) Crie uma função recursiva que “elimine” um determinado algarismo de um número. Esta função recebe o algarismo a ser eliminado.
- 9) Crie uma função recursiva que gere um número cujos algarismos são os sucessores dos algarismos de um número recebido. (Obs: o 0 é o sucessor de 9).
Por exemplo: nº recebido: 2984 nº gerado: 3095
- 10) Crie uma função recursiva que receba uma string e exiba cada palavra em uma nova linha. Considere que as palavras estão separadas por um espaço (brancos) ou vírgula.
- 11) Crie uma função recursiva que recebe uma string e retorna o número de palavras desta string. Considere que as palavras estão separadas por um espaço (brancos) ou vírgula.
- 12) Crie uma função recursiva que recebe uma string e um caractere e retorne True, se o caractere ocorre na string, False , caso contrário.
- 13) Crie uma função recursiva que receba um caractere e uma string e retorne quantas vezes o caractere aparece na string.
- 14) Crie uma função recursiva que implemente o int(), recebe uma string e, caso todos os caracteres sejam algarismos, retorne o número equivalente, caso contrário, retorna 'Erro'.
- 15) Crie uma função recursiva que receba um caractere e uma string e retorne quantas vezes o caractere aparece na string.
- 16) Crie uma função recursiva que receba uma string e retorne outra string com todos os caracteres duplicados.
Ex: manobra => mmaannoobbrraa
- 17) Crie uma função recursiva que receba uma string e retorne outra string repetindo apenas os algarismos.
Ex: prog1 => prog11
- 18) Construir uma função recursiva que indique se uma palavra (string) digitada pelo usuário é um palíndromo (retorna True) ou não (retorna False).
- 19) Crie uma função recursiva que recebe uma string e um caractere e retorne o índice do caractere.
Se o caractere não for encontrado a função deve retornar -1.
Ex: string: 'Ana Maria Martins' , caractere: 'M' --> 4
 string: 'Ana Maria Martins' , caractere: 'm' --> -1
- 20) Implemente uma função recursiva que retorne o MDC (algoritmo de Euclides) de dois números inteiros e positivos x e y, que são passados como parâmetros.