



Lista de exercícios 4 – Recursão – **Prazo: combinado em aula**

INSTRUÇÕES:

- A – Todos os arquivos compactados em formato ZIP (.zip). Inclua apenas os códigos-fonte (.c e .h), ou seja, não me envie os executáveis (.exe). O zip deve seguir o padrão: *[SEU_NOME].zip*.
B – Todas as funções solicitadas na lista de exercícios devem estar dentro de sua biblioteca. A biblioteca deve seguir o padrão: *[SEU_NOME].h* para os protótipos e *[SEU_NOME].c* para a descrição das funções.
C – Os códigos-fonte dos exercícios devem conter APENAS a função `main()` e devem seguir o padrão: *ex[NUMERO].c*. Lembre-se que são apenas 6 arquivos.
D – A lista de exercícios é entregue pelo Google Classroom.

PARTE 1 – RECURSÃO e MATEMÁTICA

1 – Escreva a função recursiva `void printNaturals(int N)` para imprimir os números naturais de 1 `a N.

Saída esperada para N=10:

Numeros naturais sao: 10 9 8 7 6 5 4 3 2 1

2 – Escreva a função recursiva `int sumNaturals(int N)` para imprimir a soma dos números naturais de 1 `a N.

Saída esperada para N=5:

A soma dos numeros de 1 a 5 eh: 15

3 – Escreva a função recursiva `int nofDigits(int N)` para contar os dígitos do numero N.

Saída esperada para N=50:

O numero de digitos de 50 eh: 2

4 – Escreva a função recursiva `int DigitSum(int N)` para calcular a soma dos dígitos do numero N.

Saída esperada para N=25:

A soma dos digitos de 25 eh: 7

5 – Escreva a função recursiva `long convertBinary(int decN)` para calcular o numero binário a partir do numero decimal decN.

Saída esperada para decN=66:

O binario correspondente ao decimal 66 eh: 1000010

6 – Escreva a função recursiva `void EvenAndOdd(int valorPadrao, int N)` para imprimir todos os números pares OU impares de 1 ate N.

Saída esperada para N=10 e valorPadrao=2:

Todos os pares de 1 `a 10 sao : 2 4 6 8 10

Saída esperada para N=10 e valorPadrao=1:

Todos os impares de 1 `a 10 sao: 1 3 5 7 9

7 – Escreva a função recursiva `int reverse(int N)` para inverter os dígitos do número N.

Saída esperada para N=123:

O inverso de 123 eh: 321

8 – Escreva a função recursiva `int findMCD(int a, int b)` para calcular o Máximo Divisor Comum (MCD) entre os números a e b.

Saída esperada para a=10 e b=50:

O MCD entre 10 e 50 eh: 10

PARTE 2 – RECURSÃO e STRING

9 – Escreva a função `int findUpper(char*)` para encontrar a primeira letra maiuscula na string usando recursao. Saída esperada:

Primeira maiuscula de "pROGRAMACAO eSTRUTURADA" eh 'R'

10 – Escreva a função `void copyString(char *str1, char *str2)` para copiar o o conteudo de str1 em str2 usando recursao. Saída esperada:

String de entrada: Programacao Estruturada

Str1: Programacao Estruturada, Str2: Programacao Estruturada

11 – Escreva a função `int isPalindrome(char *str, int index)` para verificar se uma string é palindromo utilizando recursao. Saída esperada:

"Programacao" eh palindromo? 0

"SERES" eh palindromo? 1