



INSTITUTO DE ENGENHARIAS E CIÊNCIAS DO MAR

Ficha de Trabalho #4

▷ **Tópicos:** Funções/Recursão

1. Escreva um programa que determine o maior de três números inteiros dados pelo utilizador usando a função `int maior(int a, int b, int c);`.
2. Escreva um programa que determine o maior e o menor número de N reais dado pelo utilizador. Considere que N é um valor solicitado ao utilizador.
3. Escreva um programa que calcule a média de N números reais dados pelo utilizador usando a função `float media(int N);`.
4. Escreva um programa que pede ao utilizador um valor N que corresponde a um certo período de tempo em segundos. O programa deverá apresentar no output esse período de tempo no formato HH:MM:SS. (Sugestão: utilize o operador resto da divisão (%)).
5. Escreva um programa que pede ao utilizador um valor positivo N . O seu programa deverá voltar a pedir um valor para N caso o utilizador introduza um valor inválido. Como output, deverá mostrar o número de dígitos que compõem o N , assim como a soma dos dígitos de N . Por exemplo, o número 54321 tem 5 dígitos e a soma desses dígitos é 15.
6. O índice de massa corporal (IMC) relaciona a altura com o peso da pessoa através da fórmula $IMC = peso/altura^2$. A tabela de classificação é a seguinte:

IMC	Categoria
$IMC < 19$	Magro
$19 \leq IMC \leq 25$	Normal
$25 < IMC \leq 30$	Excesso Peso
$30 < IMC \leq 40$	Obeso I
$> 40 \quad IMC$	Obeso II

Escreva um programa que calcule os seguintes indicadores: IMC, categoria, peso mínimo ($PMI = 19 \times altura^2$) e peso máximo ($PMA = 25 \times altura^2$) que garantem que o pessoal pertence à categoria normal. Utilize e implemente as seguintes funções:

```
float imc(float peso, float altura);  
void escrevaCategoria(int imc);  
float pesoMinimo(float altura);  
float pesoMaximo(float altura);
```

7. Elabore um programa que desenhe um quadrado como o que se segue utilizando a função `void quadrado(int N);`. O valor de N , dado pelo utilizador, deverá ser obrigatoriamente superior ou igual a 2. O quadrado apresentado é o exemplo para $N = 4$.

```
X 0 X 0
0 X 0 X
X 0 X 0
0 X 0 X
```

8. Elabore um programa que imprima um retângulo de dimensão $L \times C$ e preenchido com os primeiros múltiplos de 3 com cinco algarismos. Exemplo para $L = 3$ e $C = 5$.

```
1000210005100081001110014
1001710020100231002610029
1003210035100381004110044
```

9. Escreva um programa que desenhe um pirâmide de números utilizando a função `void piramide(int N);`, o valor de N , dado pelo utilizador, deverá ser obrigatoriamente superior ou igual a 2. A pirâmide apresentada é o exemplo para $N = 4$.

```
1
1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
```

10. Escreva uma função que calcule o factorial de um número. Usando essa função escreva um programa que calcule o número de combinações de m , n a n sabendo que:

$$C_n^m = \frac{m!}{n!(m-n)!}$$

Implementações Recursivas

11. Escreva um programa que calcule o máximo divisor comum de dois números inteiros positivos, baseando-se nas seguintes propriedades:

$$mdc(m, n) = \begin{cases} m & \text{se } n = 0 \\ mdc(n, m \bmod n) & \text{se } n > 0 \end{cases}$$

12. Implemente uma função recursiva que permita calcular a potência:

$$x^n = \begin{cases} 1 & \text{se } n = 0 \\ x^{n-1} * x & \text{se } n > 0 \\ x^{n+1} / x & \text{se } n < 0 \end{cases}$$

13. Pretende-se implementar uma função recursiva que recebe dois números inteiros positivos calcule o valor da função de Ackermann. A função de Ackermann é definida pela seguinte relação de recorrência:

$$Cn = \begin{cases} n + 1 & \text{se } m = 0 \\ Ackermann(m - 1, 1), & \text{se } n = 0 \\ Ackermann(m - 1, Ackermann(m, n - 1)), & \text{com } n > 0 \text{ e } m > 0 \end{cases}$$

14. Escreva uma função que devolva o número de Fibonacci associado a um determinado inteiro positivo. A sequência de Fibonacci, é definida da seguinte forma:

$$F(n) = \begin{cases} 0 & \text{se } n = 0, \\ 1 & \text{se } n = 1, \\ F(n - 1) + F(n - 2) & \text{se } n > 1 \end{cases}$$

15. O puzzle Torres de Hanoi com 3 colunas resolve-se, para um número n de peças (discos), em $h(n) = 2 \times h(n-1) + 1$ movimentos (para $n > 1$) e $h(1) = 1$. Desenvolva uma função recursiva para determinar $h(n)$ e caso $n < 1$ a função deve devolver -1. O protótipo da função a desenvolver é: `int hanoi(int n)`