

# **FUNÇÕES - ATIVIDADE**

**Prof. Humberto Razente**

**Sala 1B144**

## EXERCÍCIO 1

- Escreva um programa que tenha uma função que receba a massa (kg) e a altura (metros) de uma pessoa. Calcule e retorne o IMC (índice de massa corporal) definido como:

$$\text{IMC} = \text{peso} / (\text{altura} * \text{altura})$$

## EXERCÍCIO 2

- Escreva um programa que tenha uma função que calcule e retorne a soma dos primeiros  $n$  números inteiros elevados a si próprios:

$$S = 1^1 + 2^2 + 3^3 + 4^4 + \dots + n^n$$

## EXERCÍCIO 3

- Escreva 2 funções **recursivas** denominadas crescente e decrescente que recebam um número inteiro N e imprimam todos os números naturais de 0 até N em ordem crescente e decrescente:

```
int main( ) {  
    printf("\n crescente: "); crescente(15);  
    printf("\n decrescente: "); decrescente(15);  
    return 0;  
}
```

Saída:

crescente: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

decrescente: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

## EXERCÍCIO 4

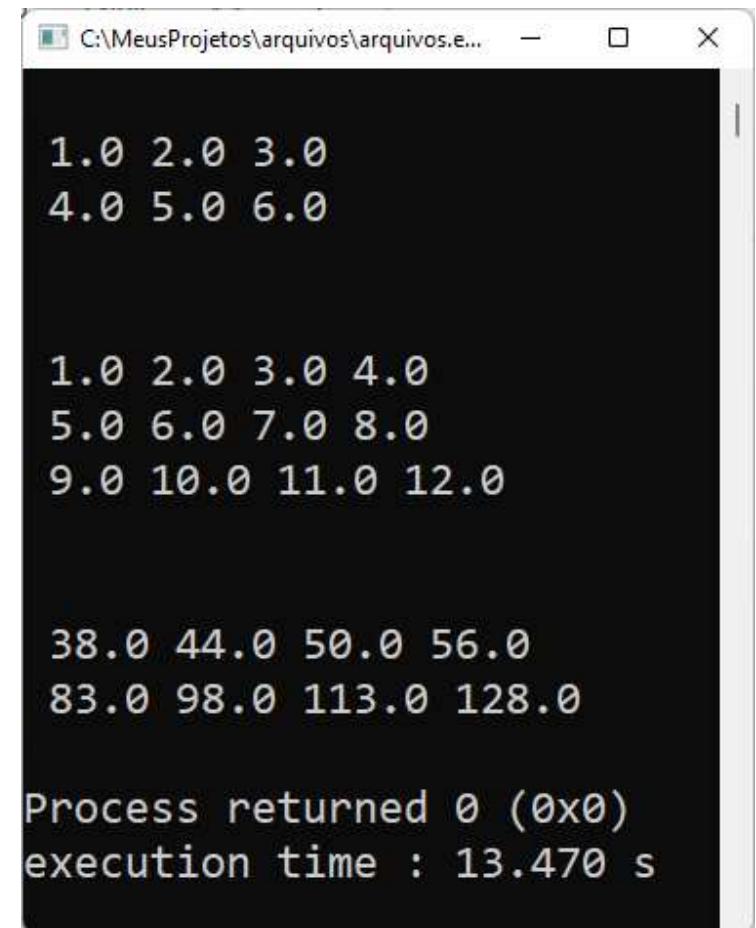
- Escreva um programa que faça a leitura de 2 matrizes do tipo *float* denominadas A e B e calcule a matriz resultante da multiplicação de A e B. As matrizes devem ser alocadas dinamicamente. O programa deve fazer uso das seguintes funções:
  - a) função ALOCA: recebe o número de linhas e o número de colunas passados por valor, e retorna um ponteiro para uma matriz alocada dinamicamente pela função
  - b) função LEITURA: recebe um ponteiro para uma matriz, o número de linhas e o número de colunas, e faz a leitura da matriz (*scanf*). O tipo do retorno deve ser *void*.
  - c) função MULTIPLICA: recebe um ponteiro para a matriz A, o número de linhas e o número de colunas de A, um ponteiro para a matriz B, o número de linhas e o número de colunas de B. A função deve executar a função ALOCA para alocar a matriz resultado, calcular a multiplicação, e retorná-la.
  - d) função IMPRIME: recebe um ponteiro para uma matriz, o número de linhas e o número de colunas, e faz a impressão (*printf*)
  - e) função DESALOCA: recebe um ponteiro para uma matriz e o número de linhas, e desaloca a matriz (*free*)

## EXERCÍCIO 4 (CONTINUAÇÃO)

- A função main deve conter:

```
int main() {  
    float **m1 = aloca(2,3);  
    leitura(m1,2,3);  
  
    float **m2 = aloca(3,4);  
    leitura(m2,3,4);  
  
    float **m3 = multiplica(m1,2,3,m2,3,4);  
  
    imprime(m1,2,3);  
    imprime(m2,3,4);  
    imprime(m3,2,4);  
  
    desaloca(m1,2);  
    desaloca(m2,3);  
    desaloca(m3,2);  
  
    return 0;  
}
```

- Exemplo de saída:



```
C:\MeusProjetos\arquivos\arquivos.e...  
  
1.0 2.0 3.0  
4.0 5.0 6.0  
  
1.0 2.0 3.0 4.0  
5.0 6.0 7.0 8.0  
9.0 10.0 11.0 12.0  
  
38.0 44.0 50.0 56.0  
83.0 98.0 113.0 128.0  
  
Process returned 0 (0x0)  
execution time : 13.470 s
```

## EXERCÍCIO 5

- Escreva uma função iterativa e uma função recursiva que receba um número inteiro positivo e retorne o enésimo termo da sequência de Fibonacci. Essa sequência começa com os termos 0 e 1, e a partir do segundo termo seu valor é dado pela soma dos dois termos anteriores. Os primeiros termos da sequência são: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34.
- Utilize o código a seguir para imprimir os 100 primeiros termos. Escreva um comentário no programa sobre o desempenho das duas funções (tempo de execução).

```
int main( ) {  
    int i;  
    for (i = 1; i < 100; i++)  
        printf("\n termo %d: %d", i, FibonacciIterativo(i));  
    for (i = 1; i < 100; i++)  
        printf("\n termo %d: %d", i, FibonacciRecursivo (i));  
    return 0;  
}
```