



Programação Imperativa e Funcional

Exercício 01

Aula 01

1. Leia três números e retorne o maior deles.
2. Converta a temperatura de Fahrenheit para Celsius e vice-versa
3. Determine se um ano é bissexto.

Aula 02

4. Calcule o determinante de uma matriz 3x3
5. Dada uma matriz de inteiros 4x4 quais os índices dos 3 números adjacentes em qualquer direção (horizontal, vertical ou diagonal) que correspondem ao maior produto de 3 números adjacentes. A seguir, um exemplo:

Dada a matriz

```
1 2 3 4
5 6 7 8
9 0 1 2
3 4 5 6
```

O retorno será

(1,1) (1,2) (1,3)

Aula 03

6. Escreva um programa que receba dois `struct` do tipo `data`, cada um representando uma data válida com dia mês e ano, e calcule o número de dias que decorreram entre as duas datas.

7. A partir da estrutura abaixo

```
struct Point{
    int x;
    int y;
};
```

, que representa um ponto num sistema 2D, defina estruturas geométricas como: triângulo, retângulo, quadrado, círculo e trapézio.

8. Escreva uma função que calcule a média de números inteiros positivos, similar a apresentada em sala de aula, mas que não passe o número total de números como parâmetro.

Aula o4

9. Qual a saída do programa abaixo (sem usar computador)

```
#include <stdio.h>
int main() {
    int i=3, j=5;
    int *p, *q;
    p = &i;
    q = &j;
    printf ("%d %d %d\n", *p - *q , **&p, 3* - *p/(*q)+7);
    return 0;
}
```

10. Usando ponteiros, escreva um programa em C que associe um array ao reverso de um array de entrada.

Aula o5

11. Usando ponteiros, escreva um programa em C que dada uma *string* retorne todas as permutações distintas possíveis entre os caracteres de entrada.

12. Crie 5 funções que imprimem diferentes mensagens e associe cada um a um inteiro. A partir da entrada do usuário, um inteiro, execute a função associada. Faça duas implementações: uma por *switch* e outra utilizando um *array* de funções, expandindo o conceito de *callback* apresentado em sala de aula, utilizando uma função que recebe um array de ponteiros de funções.

Aula o6

13. Usando ponteiros, escreva um programa em C que adicione duas matrizes.

14. Usando ponteiro, crie uma função similar à *swap* apresentada na aula 5 utilizando ponteiros, mas agora a troca será entre *strings*. Receba essas *strings* como entrada do usuário.

Aula o7

15. Crie um programa que chama uma função que inicialize os valores de uma matriz de duas dimensões de acordo com a entrada do usuário, sem indicar o tamanho das dimensões no argumento referente a matriz, mas sim como outros argumentos independente.

16. A partir da estrutura

```
typedef struct node{struct node * next;} node;
```

Escreva um programa que receba um inteiro *n* do usuário e retorne um estrutura encadeada formada por *n* *node*.

17. Crie um programa que receba as dimensões de um array de ponto flutuantes, chame uma função que aloca espaço para tal array (*allocateArray*), em que as entradas sejam a dimensão do array e uma referência para o array, e nada retorne (*void*).

Aula o8

18. Escreva um programa para determinar o número de linhas de um arquivo.

19. Converter todas as maiúsculas de um arquivo em minúsculas.

20. Refaça a questão 6, mas agora a matriz está armazenada num arquivo e suas dimensões não se restringe a 4x4.