

Exercício Ponteiros e Estrutura

1. Crie uma função que recebe 2 struct Aluno e troque os nomes dos mesmos;

```
void swapName(Aluno* a1, Aluno* a2);
```

2. Cria uma função que copie um vetor **v** de tamanho **n** passado para uma variável externa.

```
int* copy(int* v, int n);
```

3. Crie uma estrutura **Album** que guarde um vetor de **Musicas** e uma função que atualiza o preço do album como a soma dos preços das música multiplicado por uma taxa **x** (%);

```
struct Music{
    string nome;
    string autor;
    float preco;
    int ano;
};
struct Album{
    Music* musics; // vetor
    float preco;
};
//função
void updatePrice(Album* a, int x);

// Album.preco = sum(musics.preço)*x
/* exemplo:
sum(musics.preco) = 46.78
x = 5 // aumento de 5%
*/
Album.preco += 46.78*0.05 // ou = 46.78*1.05
```

4. [extra] Faça uma estrutura de Musica, junto com as funções de criar, imprimir, copiar e busca por nome. Agora faça as seguintes operações. * Recebe do usuário quantas musicas vão ser criadas; * Crie estas musicas; * Salve-as em um vetor; * crie flags em cada

```
struct Music{
    string nome;
    int ano;
    string autor;
    float preco
}
```

Com base na estrutura a baixo faça o que se pede:

a. Printe o vetor de musica na ordem de inserção;

```
void printMusic(Music** m, int n);
```

```
//porque ponteiro duplo? **
```

```
// Porque m é um vetor que armazena ponteiros para musicas.
```

b. Crie uma função que busque por nome da musica e edite o nome do autor;

```
void searchMusic(Music** m, int n, string musicName, string setAutor);
```

c. Crie uma função que ordene as musica por nome (*pode ser qualquer método que conhecer*);

```
void sortName(Music** m, int n);
```

d. Utilize a função que copia uma musica para uma variável fora do vetor;

```
Music copyMusic(Music* music);
```

5. Faça um algoritmo que recebe 2 vetores, **v** e **w**, de tamanhos, **n** e **m**, e retorna um vetor que intercale os elementos dos dois vetores.

```
int* intercal(int v*, int n, int* w, int m);
```

```
//exemplos
```

```
int* v; // 1 3 5 7 9 11
```

```
int *w; // 2 4 6 8 10
```

```
int* z = intercal(v,6, w, 5); //- > [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
```