



PCS 3111 - Laboratório de Programação Orientada a Objetos para Engenharia Elétrica

2021

Aula 02 – Ponteiros, Testes e Depuração

Atenção

1. Código inicial para resolução dos exercícios encontra-se **disponível no Discord**.
2. Os tipos, os nomes e os parâmetros das funções **devem seguir o especificado** em cada exercício para fins de correção automática.
3. A função `main` **não deve ser submetida**. Caso contrário, a correção automática retornará um *Compilation Error*.

Exercício 1

Considere um software que tem como função gerenciar as informações de uma *playlist* de músicas. Suponha que essas músicas têm três informações importantes: o nome, a duração e o estilo musical. Essas informações são armazenadas em três vetores diferentes. Por exemplo, se tivéssemos os seguintes vetores:

```
nomes = {"Redemption Song", "Alien", "Octavarium"}, duracoes = {4, 3, 24},  
estilos = {"Reggae", "Alternativa", "Metal"}
```

teríamos três músicas:

- "Redemption Song", do estilo "Reggae" e com duração de 4 minutos;
- "Alien", do estilo "Alternativa" e com duração de 3 minutos; e
- "Octavarium", do estilo "Metal" e com duração de 24 minutos.

Deseja-se implementar a seguinte função:

```
string* encontrarMusica(int duracoes[], string estilos[], string nomes[],  
                        int quantidade, string estilo, int duracao);
```

Essa função deve procurar o nome de uma música cujo estilo e duração são passados nos parâmetros `estilo` e `duracao`, respectivamente. O parâmetro `quantidade` se refere ao tamanho dos três vetores que especificam a playlist de músicas.

A função deve retornar um ponteiro para a **posição de memória do vetor `nomes`** onde está guardado o nome da música encontrada. Caso haja mais de uma música com o mesmo estilo e duração desejados, a música escolhida deve ser a **primeira** que satisfaz os critérios. Caso a música não seja encontrada, retorne o valor `NULL`. A função não deve alterar os vetores que são passados como argumento.

Usando como exemplo os vetores apresentados anteriormente, se `estilo == "Metal"` e `duracao == 24`, a função deve retornar o **ponteiro** equivalente ao índice 2 do vetor `nomes`, referente ao nome "Octavarium" (ou seja, será uma posição de memória como `0x6efea8`).



Obs: não se esqueça de fazer o include de `<string>` no arquivo e adicionar o comando `using namespace std;`

Exercício 2

Parâmetros podem ser usados como entrada e/ou saída de funções. Para criar um parâmetro de saída pode-se usar ponteiros ou passagem por referência. Trabalharemos agora com três tipos de saída: através do retorno da função, através de ponteiros e através de referência. Implemente a função a seguir:

```
int calcularEstatisticas(int duracoes[], int quantidade, int& minimo, int* maximo)
```

Essa função recebe o vetor `duracoes` cujo tamanho é definido em `quantidade`. Os valores da duração da música mais curta e da música mais longa são retornados pelos parâmetros `minimo` e `maximo`, respectivamente. Atribua o valor da duração mínima ao parâmetro mínimo; atribua a duração máxima ao valor apontado por máximo. O valor da duração da *playlist* (que é a soma da duração de todas as músicas nela contida) é retornada por meio da função com o comando `return`.

Note que o valor do mínimo é passado como referência e o máximo é passado por ponteiro.

Assuma que as durações passadas são sempre positivas. Para o caso de uma *playlist* vazia, ou seja, `quantidade == 0`, **todos** os valores retornados devem valer 0.

Por exemplo, considere os vetores do exercício 1. Teríamos que o valor mínimo valeria 3, o valor máximo valeria 24 e o retorno da função daria $4 + 3 + 24 = 31$.

Cuidado: as variáveis `minimo` e `maximo` não necessariamente são inicializadas antes de serem passadas à função `calcularEstatisticas`. Por exemplo, a função pode ser chamada da seguinte forma:

```
int main() {  
    int duracoes[] = {1, 2, 3};  
    int quantidade = 3;  
  
    int minimo; // valor inicial não definido  
    int maximo; // valor inicial não definido  
    int duracao = calcularEstatisticas(duracoes, quantidade, minimo, &maximo);  
  
    // fim  
}
```

No exemplo acima, espera-se que em “fim” tenha-se `minimo=1`, `maximo=3` e `duracao=6`, apesar dos valores iniciais de `minimo` e `maximo` não serem inicializados (e poderem conter “lixo”).

Testes do Judge

Exercício 1

- Teste com música no início do vetor;



- Teste com música no meio do vetor;
- Teste com música no fim do vetor;
- Teste com a música repetida;
- Teste sem a música desejada;
- Teste se os vetores não são alterados pela função.

Exercício 2

- Teste com vetor vazio;
- Teste com um elemento no vetor;
- Teste com quantidade maior do que 1;