



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

Aula 02 – Ponteiros, Testes e Depuração

Atenção

1. Código inicial para resolução dos exercícios encontra-se **disponível no e-Disciplinas**.
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 permite calcular o custo e a duração estimada de atividades dentro de um projeto trabalhe com **pessoas**, as quais podem ser designadas para cumprir determinadas atividades. Suponha que as informações de cada pessoa sejam guardadas por dois vetores de double: valoresPorHora e horasDiarias, que armazenam os valores que elas recebem por hora trabalhada e o número de horas que elas têm disponível por dia, respectivamente.

Por exemplo, se tivéssemos as pessoas Antônio, que recebe 10 reais/hora e possui disponibilidade de 8 horas/dia, e Bruna, que recebe 9 reais/hora e possui disponibilidade de 11 horas/dia, os vetores seriam:

valoresPorHora = {10.0, 9.0} e horasDiarias= {8.0, 11.0}.

Deseja-se implementar a seguinte função:

```
double* encontrarPessoa(double valoresPorHora[], double horasDiarias[],  
                        int quantidade, double valorPorHora, double horaDiaria)
```

Essa função deve retornar o ponteiro para o vetor de **valoresPorHora** equivalente à pessoa cujas informações correspondam aos parâmetros valorPorHora e horaDiaria recebidos pela função. O parâmetro quantidade informa a quantidade de pessoas existentes. Note que os dois vetores têm o mesmo tamanho. Caso existam duas ou mais pessoas com as mesmas informações, a função deve retornar a **primeira** ocorrência. Caso não exista nenhuma pessoa com as especificações passadas, a função deve retornar NULL.

Dica: Por se tratarem de pontos flutuantes, o resultado de se utilizar o operador “==” para comparar dois valores de double pode não ser o esperado. Assim, é melhor trabalhar com um intervalo em que a diferença entre dois valores é considerada suficientemente pequena para que sejam considerados iguais. Para isso, utilize a função ehProximo fornecida, se precisar realizar essa comparação.

```
#include <cmath> // para o abs  
  
bool ehProximo(double a, double b) {  
    return abs(a - b) < 0.001;  
}
```



Usando como exemplo os vetores apresentados anteriormente, se `valorPorHora == 9.0` e `horasDiarias == 11.0`, a função deve retornar o ponteiro equivalente ao índice 1 do vetor `valoresPorHora[]`, referente ao valor por hora trabalhada de Bruna.

Exercício 2

Parâmetros podem ser usados como entrada e/ou saída de funções. Para se ter um parâmetro de saída pode-se usar ponteiros ou passagem por referência. Considerando isso, implemente a seguinte função:

```
void calculaAtividade (double horasDiarias[], double valoresPorHora[],  
                     int horasNecessarias, double* custo, int &duracao,  
                     int quantidade);
```

Essa função recebe vetores com as *horas disponíveis por dia* e os *valores por hora trabalhada* das pessoas adicionadas a uma atividade (como no exercício 1), com número de pessoas, quantidade e tem como objetivo calcular a duração (em dias) e o custo da atividade.

A duração em dias estimada para a atividade deve ser calculada como:

$$duracao = \left\lceil \frac{\text{horas necessárias para terminar a atividade}}{\sum \text{horas por dia de cada pessoa adicionada}} \right\rceil$$

Dica: Use a função `ceil` para calcular o teto (arredondamento para cima). Para isso faça include da biblioteca `cmath` e faça `"using namespace std;"`.

O custo de cada pessoa deve ser calculado como:

$$custo = duracao * \sum (\text{número de horas disponíveis por dia} \\ * \text{valor por hora trabalhada de cada pessoa})$$

Note que o valor do custo é passado como ponteiro e a duração da tarefa é passada por referência.

Assuma que os valores passados pelos vetores sejam sempre positivos. Para o caso de nenhuma pessoa ter sido adicionada à atividade, ou seja, `quantidade == 0`, tanto `custo` como `duração` devem retornar -1.

Por exemplo, considere os vetores que contém as informações de Antônio e de Bruna, do exercício 1. Se fossem adicionados à uma atividade que necessite de 35 horas para ser completada, a duração seria de $\left\lceil \frac{35}{8+11} \right\rceil = 2$ dias, e o custo seria de $2 * 8 * 10 + 2 * 11 * 9 = \text{R\$358,00}$.

Testes do Judge

Exercício 1

- Testa com pessoa no início do vetor;
- Testa com pessoa no meio do vetor;



- Testa com pessoa no fim do vetor;
- Testa sem a pessoa passada como parâmetro;
- Testa com duas pessoas iguais;

Exercício 2

- Testa com quantidade 0;
- Testa com quantidade 1;
- Testa com quantidade maior do que 1;
- Testa com duração arredondada, em dias;