



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

### Aula 01 – Introdução

#### Cuidados

1. Os tipos, os nomes, e os parâmetros das funções **devem seguir o especificado** em cada exercício para fins de correção automática.
2. A função main **não deve ser submetida**. Caso contrário, a correção automática retornará um *Compilation Error*.

#### Exercício 1

Implemente a função:

```
int calculaSalarioMensal(int tipo, int horasDiarias)
```

que retorna o salário mensal de um funcionário de uma determinada empresa. Existem três tipos de funcionários, cujos salários são determinados da seguinte forma:

- Tipo 1  $\Rightarrow$  **15 reais por hora**
- Tipo 2  $\Rightarrow$  **(600 + 20 \* horas diárias) por semana**
- Tipo 3  $\Rightarrow$   $\frac{24000 * \text{horas diárias}}{24}$  **por mês**

Considere que todos os empregados trabalham cinco dias por semana e que um mês possui quatro semanas.

*Exemplo:* calculaSalarioMensal(1, 4) deve retornar 1200.  
calculaSalarioMensal(2, 5) deve retornar 2800.  
calculaSalarioMensal(3, 7) deve retornar 7000.

Suponha que a função não será chamada com valores não especificados de tipo.

#### Exercício 2

Um empregado deseja verificar se concluirá uma atividade dentro do prazo estipulado. Para isso, implemente a função:

```
bool estaNoPrazo(int prazo, int horasAtividade, int horasDiarias)
```

que devolve true caso o prazo seja cumprido e false, caso contrário. O parâmetro horasAtividade representa as horas necessárias para finalizar a atividade e o prazo é dado em dias. Como esse funcionário faz um treinamento interno, após 3 dias de trabalho no projeto esse funcionário trabalha um dia por meio período no projeto. Ou seja, por 3 dias ele trabalha horasDiarias e no 4º dia ele trabalha horasDiarias/2 (despreze o resto da divisão).



Por exemplo, se `horasDiarias = 2`, para uma atividade que demanda 16 horas, a função deve retornar `true` para o parâmetro `prazo ≥ 9`. Nessa situação, o funcionário levará nove dias para finalizá-la, como mostrado abaixo:

- 1º dia: 2 horas
- 2º dia: 2 horas
- 3º dia: 2 horas
- 4º dia: 1 hora (`horasDiarias/2`)
- 5º dia: 2 horas
- 6º dia: 2 horas
- 7º dia: 2 horas
- 8º dia: 1 hora (`horasDiarias/2`)
- 9º dia: 2 horas

**Dica:** use um laço para fazer o cálculo.

### Exercício 3

Suponha que uma empresa deseja estimar as despesas com funcionários na realização de uma determinada atividade. Implemente a função:

```
double calculaCusto(int horasGastas[], double custoHora[], int quantidade)
```

que retorna o custo de uma certa atividade. O parâmetro `quantidade` representa o número de funcionários disponíveis para executá-la. São passados dois vetores: um contém as horas que serão gastas por cada funcionário e o outro, o valor cobrado por hora. Ambos os vetores possuem o mesmo tamanho.

Caso não existam funcionários disponíveis no momento (ou seja, a quantidade for 0), a função deve retornar -1.

Por exemplo, suponha uma atividade com 3 funcionários disponíveis.

- Funcionário A: 4 horas e R\$25,50/hora
- Funcionário B: 2 horas e R\$50,00/hora
- Funcionário C: 3 horas e R\$30,00/hora

Nesse caso, os vetores seriam {4, 2, 3} e {25.5, 50, 30} e o custo equivaleria a 292.

### Testes do Judge

#### Exercício 1

- Teste `calculaSalarioMensal` Funcionário tipo 1
- Teste `calculaSalarioMensal` Funcionário tipo 2
- Teste `calculaSalarioMensal` Funcionário tipo 3

#### Exercício 2

- Teste `estaNoPrazo` Funcionário dentro do prazo e `horasAtividade < horasDiarias`
- Teste `estaNoPrazo` Funcionário dentro do prazo e `horasAtividade > horasDiarias`
- Teste `estaNoPrazo` Funcionário fora do prazo

#### Exercício 3

- Teste `calculaCusto` Sem funcionários disponíveis



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO  
Departamento de Engenharia de Computação e Sistemas Digitais

---

- Teste calculaCusto Com um funcionário disponível
- Teste calculaCusto Com vários funcionários disponíveis