

UNIVERSIDADE T - 1 -ECNOLÓGICA FEDERAL DO PARANÁ
Campus de Curitiba (Brasil) - **Departamento Acadêmico de Eletrônica (DAELN).**

Prof.: Jean Marcelo Simão.

Exercício Complementar (Struct) – Programação em Linguagem C.

Fazer um programa que permita ao usuário escolher entre um conjunto de funcionalidades relativas ao tratamento de uma folha de pagamento de uma empresa. Estas funcionalidades são:

- Entrar com: os salários, os nomes dos funcionários, sexos dos funcionários e os índices de produtividade (de 0,1 à 1,0).
- Calcular a média salarial.
- Calcular o maior salário.
- Calcular o menor salário.

- Calcular quantos salários são maiores que a média salarial.
- Calcular quantos salários são menores que a média salarial.
- Calcular quantos salários são iguais a média salarial.

- Especificar quais são os funcionários que têm salário maior que a média salarial
- Especificar quais são os funcionários que têm salário menor que a média salarial.
- Especificar quais são os funcionários que têm salário igual a média salarial.

- Calcular gratificação salarial baseado na seguinte fórmula:

$$\text{gratificação} = \text{índice de produtividade} * \text{percentual do salário},$$

onde *percentual do salário* é igual à:

- 100% para salários abaixo da média salarial.
 - 80% para os salários menores que $(\text{média_salarial} + \text{maior_salário}) / 2$.
 - 50% para os demais salários.
- Calcular o maior e o menor salário levando em conta as gratificações.

Observações:

- Os dados de cada funcionário devem ser tratados por uma “variável” de um tipo *Funcionario* que será uma *struct* definida pelo ‘programador’. Na verdade os dados dos funcionários estarão num vetor de *Funcionario*.
- Todas estas funcionalidades podem ser aplicadas a ambos os sexos ou somente a um deles conforme desejo do usuário.
- A quantia de funcionários pode variar entre 10 à 50 e não deve haver desperdício nem falta de memória relativo a esta quantia (i.e. utilizar alocação dinâmica de memória).
- As funcionalidades devem ser tratadas por funções.
- Funcionalidades efetivamente parecidas devem ser tratadas por uma única função cuja diferenciação será feita por meio de parâmetros da função e estruturas de decisão. Isto significa que se deve evitar a repetição de código/funções.
- Pelo menos a função que calculará o maior e o menor salário (considerando a gratificação) terá estes resultados devolvidos por meio de parâmetros por referência.

- Procurar utilizar diferentes estruturas de decisão e repetição, diferentes formas de aplicar funções e diferentes formas de percorrer vetores (i.e. tradicional e aritmética de ponteiros), visando contemplar um maior aprendizado.