

Universidade Federal Fluminense
Escola de Engenharia
Linguagens de Programação para Engenharia Elétrica TEE00140
2017 (1º período)

Exercício para entrega – Ponteiros, tipos de dados definidos pelo usuário e classes

1) Escreva uma subrotina HM que converte minutos em horas e minutos. A subrotina recebe um inteiro *mnts* e os endereços de duas variáveis inteiras, *h* e *m*, e atribui valores a essas variáveis sendo *m* os minutos (menor que 60) e *h* horas, tal que $mnts = 60 \cdot h + m$. Escreva um programa que use a subrotina HM.

2) Crie uma função que faça a alocação dinâmica de memória. Ela deve receber como argumento o tamanho do vetor e devolver um ponteiro indicando a área de memória armazenada. Teste a função num programa que recebe dois vetores de tamanhos distintos.

3) Uma empresa precisa armazenar os seguintes dados de um cliente: nome (máximo 50 caracteres), CPF, renda mensal e ano de nascimento. Crie a estrutura de dados CLIENTE que realize essa função e um programa que leia o número de clientes e os dados de cada um deles. Use um vetor de objetos de CLIENTE. Refaça o programa incluindo a alocação dinâmica de memória desse vetor de objetos.

4) Um programa deve calcular o vetor resultante das forças num determinado ponto. Isso é implementado como uma coleção (vetor computacional) de objetos que armazenam as coordenadas de cada vetor força. Defina uma classe VETOR com as seguintes características: pode aceitar tantos vetores bidimensionais (coordenadas *x* e *y*) quanto tridimensionais (coordenadas *x*, *y* e *z*); as coordenadas podem ser introduzidas na declaração do objeto ou através da função *setCoords*; deve-se sobrecarregar o operador “+” para que opere com objetos de VETOR, fazendo a soma das coordenadas correspondentes. Teste a classe num programa que receba como dados de entrada o número de forças aplicadas num ponto e as coordenadas de cada vetor força. Utilize alocação dinâmica de memória nesse programa.

Sugestão: Convém fazer a implementação do que se pede por partes. Pode-se por exemplo, produzir um código básico para vetores bidimensionais, ao qual são adicionadas melhorias.

Entrega **no dia da P2**