

Lista de Exercícios

1. Declare uma struct que possa armazenar um horário do dia (hora, minuto e segundo). Em seguida faça as seguintes funções:

- a. Uma função que receba um horário e retorne se este horário é um horário válido do dia ou não (considerando o intervalo válido de horário de 0:00:00 até 23:59:59).
- b. Uma função que receba dois horários e retorne se o primeiro argumento é um horário posterior ao segundo.
- c. Uma função que receba dois horários e retorne se o primeiro horário é um horário anterior ao segundo.
- d. Uma função que receba dois horários e retorne se são iguais (faça como se a struct não fosse declarada como transparent).
- e. Uma função que receba um valor em segundos (assuma que esse valor estará entre 0 e 86399) e converta para um horário.
- f. Uma função que receba dois horários e retorne a diferença em segundos entre ambos.
- g. Uma função que receba dois horários e retorne a uma struct de horário que seja a diferença de ambos (ou seja, diferença em horas, minutos e segundos).

2. Declare uma struct que armazene o nome, peso, idade de uma pessoa. Em seguida faça funções para:

- a. Verificar se os dados da pessoa são válidos (idade e peso positivos).
- b. Receber duas pessoas e retornar a mais velha (em caso de mesma idade, retorne a pessoa que foi passada como primeiro argumento).
- c. Receber duas pessoas e retornar a mais pesada (em caso de mesmo peso, retorne a que foi passada como primeiro argumento)
- d. Receber uma pessoa e retornar ela com um ano a mais.
- e. Receber uma pessoa, um valor real que represente quantos quilos a pessoa ganhou (ou perdeu) e retornar a pessoa com um novo peso.
- f. Calcular o índice de massa corporal (IMC) da pessoa.

O IMC de uma pessoa é dado pela fórmula a seguir:

$$IMC = \frac{Peso(kg)}{Altura(m)^2}$$

- g. Classificar a pessoa de acordo com seu IMC considerando a tabela a seguir

IMC	Classificação
Menor que 18,5	Magreza
18,5 a 24,9	Normal
25 a 29,9	Sobrepeso
30 a 34,9	Obesidade grau I
35 a 39,9	Obesidade grau II
Maior que 40	Obesidade grau III

3. Declare uma struct que armazene uma data (dia, mes e ano). Em seguida faça funções para:

- a. Receber uma data e verificar se ela é válida (lembre-se que nos anos bissextos o mês de fevereiro tem 29 dias).

Um ano bissexto é um ano que satisfaz uma das duas condições

- É múltiplo de 400; ou

- É múltiplo de 4 mas não é múltiplo de 100.
- b. Receber duas datas e retornar verdadeiro se a primeira for maior que a segunda.
- c. Receber duas datas e retornar verdadeiro se a primeira for menor que a segunda.
- d. Receber duas datas e retornar verdadeiro se elas forem a mesma data (faça como se a struct não fosse declarada como transparent).