



Lista de exercícios – Struct

Desenvolva todos os exercícios considerando os novos tipos de dados a seguir:

Data - dia, mês e ano

Horário - horas, minutos e segundos

Pessoa - altura (metros), idade, sexo, peso

PARTE 1

1. Crie novos tipos abstratos (TAD) de dados por meio de struct;
2. Na função `main()`, declare uma variável de seu novo tipo de dado;
3. Usuário deve preencher todos os dados do TAD por meio de `scanf`;
4. Valide os dados do TAD quando necessário (exemplo: `1<mes<12`);
5. Imprime todos os itens do TAD;

PARTE 2

1. Crie funções para ler os dados de cada um dos TADs
2. Crie funções para inicializar os dados de cada um dos TADs
3. Crie funções para imprimir os dados de cada um dos TADs já preenchidos
4. Desenvolva as seguintes funções
 - `void escrevaDataExtenso(Data * data)` para escrever por extenso a data passado por parâmetro seguinte padrão: `<dia> de <mes_por_extenso> de <ano>`
 - `void escrevaHoraPadrao12h(Horario * hora)` para escrever a hora passada como parâmetro no padrão americano: as horas são números entre 0 e 11 e ao final do horário, o código “AM” indica horário antes do meio dia e o código “PM” indica horário após do meio dia. Exemplo do padrão: `<horas> : <minutos> : <segundos> <AM ou PM>`
 - `float calculaIMC(Pessoa * alguem)` para calcular o IMC de alguém passado como parâmetro. Fórmula do IMC é $IMC = peso / (altura * altura)$

PARTE 3

1. Declare vetores de cada TAD. Exemplos:
 - `aniversarios[]` pode ser um vetor de Data
 - `alarme[]` pode ser um vetor de Horario
 - `gurizada[]` pode ser um vetor de Pessoa
2. Preencha ou leia cada um dos vetores de TADs com as funções da parte 2
3. Utilize as funções de imprimir para visualizar todos os elementos do vetor de TAD.

PARTE 4

1. Crie as seguintes funções de contagem com nomes auto-explicativos:
 - `int contaDatasNoMes(Data * vetData, int tamanho, int mes)`
 - `int contaCompromissosAposHora(Horario * vetHorario, int tamanho, int hora)`
 - `float mediaDePessoaIMC(Pessoa * vetPessoa, int tamanho)`
 - `Pessoa * consultaPessoaMaisNova(Pessoa * vetPessoa, int tamanho)`
 - `Pessoa * consultaPessoaMaisBaixa(Pessoa * vetPessoa, int tamanho)`