



Questão 1.

Crie uma estrutura representando um aluno de uma disciplina. Essa estrutura deve guardar as seguintes informações: o número de matrícula do aluno (um inteiro de 4 dígitos), sexo (masculino ou feminino), seu índice de rendimento (um inteiro de 0 a 100) e se é ou não a primeira vez que o aluno faz a disciplina.

Escreva um programa que mostre o tamanho em bytes dessa estrutura e realize os ajustes necessários para que ela não possua mais que 4 bytes.

Questão 2.

Crie um programa de gerenciamento de estoque. O programa deve exibir um menu com todas as opções do usuário.

- Cadastrar um produto
- Listar os produtos (em ordem alfabética)
- Excluir um produto

Um produto tem os seguintes atributos:

- Código
- Nome
- Marca
- Preço

Crie uma estrutura para representar os produtos.

Questão 3.

Elabore um programa que

- Crie um tipo *Aluno* baseado em uma estrutura com três notas (números reais de 0 a 10) como membros.
- Inicialize aleatoriamente um vetor com 20 posições do tipo *Aluno* as notas precisam ser números reais de 0 a 10 com uma casa decimal.
- Em seguida, imprima o vetor mostrando as notas de cada aluno, seguidas da média do aluno e da palavra *Aprovado* se a média for maior ou igual a 7,0, ou *Reprovado*, caso contrário.

Questão 4.

- Crie um tipo *Métricas* baseado em uma estrutura com os campos: *precisão* e *acurácia*.
- Implemente a função `Métricas calcularMetricas(float m[2][2])` que recebe como parâmetros uma matriz 2×2 e calcule os valores de precisão e acurácia da seguinte maneira:

a	b
c	d

$$\text{acurácia} = \frac{(a+d)}{(a+b+c+d)}$$

$$\text{precisão} = \frac{d}{(c+d)}$$

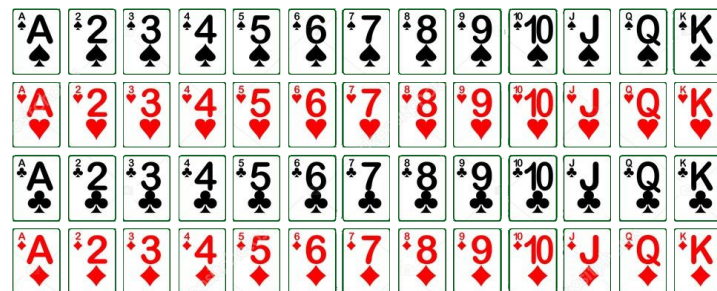
- A função retorna uma estrutura com os valores de precisão e acurácia preenchidos.
- Para demonstrar o cálculo, na função `main`, inicialize uma matriz e use a função para preencher os membros de uma estrutura *Métricas*, em seguida imprima no console os valores a partir da estrutura.

Questão 5.

- Crie um tipo *Candidato* baseado em uma estrutura com os membros: *notaProvaEscrita* (real), *notaProvaDidatica* (real), *notaProvaTitulos* (real) e *idade* (inteiro).
- Crie uma função que recebe como parâmetro dois candidatos e retorna o melhor colocado segundo os seguintes critérios:
 1. Maior soma das notas da prova escrita, prova didática e de títulos
 2. Em caso de empate: maior idade
- Crie uma função que recebe um vetor do tipo *Candidato* e seu tamanho, a função deve retornar um ponteiro para a posição do candidato mais bem colocado.
- Inicialize um vetor do tipo *Candidato* aleatoriamente com 10 candidatos, notas (real de 0 a 10) e idades (inteiro de 18 a 60).
- Na função `main` imprima as notas de todos os candidatos junto com sua soma e a idade do candidato. Em seguida demonstre o uso da função que seleciona o melhor candidato e imprima os dados do candidato selecionado.

Questão 6.

Crie um tipo *Carta* baseado em uma estrutura com os membros: *face* e *naipe*. Inicialize um vetor com uma sequência de cartas como na imagem abaixo.



Crie uma função que recebe um vetor da estrutura *Carta* e aleatoriza as posições das cartas no vetor.