

## Aula 10 – Exercícios

(deve ser entregue via Moodle)

---

- 1) Faça um programa que leia o nome, a idade e o endereço (rua e número) de uma pessoa e armazene os dados em uma estrutura. Mostre na tela as informações armazenadas.
- 2) Crie uma estrutura para armazenar os dados de um aluno: nome, número de matrícula e curso. Leia as informações de 5 alunos e armazene em um vetor dessa estrutura. A seguir, imprima os dados de todos os alunos na tela. Siga o formato de saída mostrado no Moodle.
- 3) Crie uma estrutura representando os dados de um aluno: matrícula, nome, nota da primeira prova, nota da segunda prova e nota da terceira prova. A seguir: **a)** entre com os dados de 5 alunos. **b)** encontre o aluno com maior nota na primeira prova, imprimindo seu nome completo e a nota, com 2 casas decimais. **c)** encontre o aluno com maior média geral, imprimindo seu nome completo, seguido da média, com 2 casas decimais. **d)** imprima uma listagem de matrícula e nome, dizendo também se cada aluno foi aprovado ou reprovado, considerando ao menos 60% de média para a aprovação.
- 4) Crie uma estrutura para representar pontos em coordenadas polares: raio  $r$  e argumento  $a$  (em graus). Crie uma estrutura para representar pontos em coordenadas cartesianas:  $x$  e  $y$ . Leia 6 pontos em coordenadas polares e armazene em um vetor. A seguir, converta esses pontos para coordenadas cartesianas e armazene o resultado em outro vetor. A conversão é dada por:  $x = r * \cos(a)$  e  $y = r * \sin(a)$ . Por fim, mostre o resultado de cada ponto em coordenadas cartesianas ( $x, y$ ), com 1 casa decimal. Lembre-se que o valor do argumento deve estar em radianos para usar as funções  $\cos$  e  $\sin$ .
- 5) Utilizando um vetor de estrutura, faça um programa que leia o nome e o telefone de 5 pessoas e os imprima em ordem alfabética pelo nome. Dica: pesquise os detalhes de uso da função `strcmp()`.
- 6) Faça um programa que leia um vetor de estrutura com os dados de 8 carros: marca, preço e ano. A seguir, leia um valor  $P$  e mostre as informações de todos os carros com preço menor que  $P$ . Repita esse processo até que seja lido um valor  $P \leq 0$ .
- 7) Em um vetor de estrutura, armazene as seguintes informações sobre 8 livros: título, autor e ano. A seguir, leia o nome de um autor e mostre os títulos e anos dos livros escritos por ele. Repita o processo até que o usuário digite “sair”.
- 8) Declare uma estrutura para representar números complexos, formados por parte real e parte imaginária. A seguir, leia dois números complexos (parte real seguida de parte imaginária de cada um). Apresente como resposta a soma, a subtração e o produto dos números complexos, sendo um valor por linha.

---

Siga os formatos de entrada e saída mostrados no Moodle.

## Desafios

9) Escreva um programa que simule o jogo *Blackjack* (também chamado de 21). Inicialmente, crie a struct para representar as cartas e um vetor para armazenar todo o baralho. Faça a atribuição dos valores automaticamente. O objetivo do jogo é fazer mais pontos que o adversário (computador). Basicamente, o jogador pode retirar uma carta do baralho e o valor de cada carta (independente do naipe) é somado ao seu total (considere que A vale 1, e Q, J, K sempre valem 10). Cada vez que o jogador escolher retirar uma carta, o programa sorteia uma carta aleatória (posição do vetor) e a mostra na tela, até que o jogador escolha não mais retirar cartas. Então, o programa sorteia as cartas para o computador, *trapaceando para que o computador nunca estoure 21 pontos* (caso em que se perde o jogo). Além de implementar a “trapaça”, você não deve permitir que a mesma carta apareça mais de uma vez.

Resultados de 3 execuções do programa do professor:

*Execução 1: jogador ganha*

```
Sua primeira carta eh:
J (copas)                Total de Pontos do Jogador = 10

Deseja retirar outra carta (s/n)?
s

Sua proxima carta eh:
9 (paus)                 Total de Pontos do Jogador = 19

Deseja retirar outra carta (s/n)?
n

Computador retirou uma carta: J (paus)    Total de Pontos do Computador = 10
Computador retirou uma carta: 4 (ouro)    Total de Pontos do Computador = 14

*** Voce venceu!!! ***
```

O computador só para de retirar cartas quando estoura. Nota-se que isto aconteceu na terceira carta. O computador tinha 14 pontos até a segunda carta e “estourou” na terceira, porém, devido à implementação da “trapaça”, ele parou de retirar cartas.

*Execução 2: computador ganha*

```
Sua primeira carta eh:
7 (copas)                Total de Pontos do Jogador = 7

Deseja retirar outra carta (s/n)?
s

Sua proxima carta eh:
4 (espada)               Total de Pontos do Jogador = 11

Deseja retirar outra carta (s/n)?
n

Computador retirou uma carta: J (copas)    Total de Pontos do Computador = 10
Computador retirou uma carta: Q (espada)   Total de Pontos do Computador = 20

*** Computador venceu!!! ***
```

### Execução 3: jogador estoura

```
Sua primeira carta eh:  
J (ouro)          Total de Pontos do Jogador = 10  
  
Deseja retirar outra carta (s/n)?  
S  
Sua proxima carta eh:  
9 (espada)       Total de Pontos do Jogador = 19  
  
Deseja retirar outra carta (s/n)?  
S  
Sua proxima carta eh:  
J (paus)         Total de Pontos do Jogador = 29  
  
*** Game over: voce estourou os 21 pontos!!! ***
```

---

**10)** Escreva, em linguagem C, um programa que permita armazenar os dados de disciplinas e professores. Uma disciplina possui um código único, um nome e uma carga horária. Um professor possui um número funcional único, um nome e uma titulação (mestrado/doutorado). Disciplinas e professores devem ser armazenados em vetores de estrutura com no máximo 10 posições. O programa deve permitir realizar as seguintes operações:

- Cadastrar disciplinas e professores.
- Listar os dados de disciplinas e professores pelo código ou número funcional.
- Vincular uma disciplina a um professor.
- Dado um professor, listar as disciplinas que ele leciona.