

Lista de Exercícios 3 – Estruturas – Linguagem de Programação II

1. Escreva um programa que leia os dados de um aluno e os imprima na tela. Considere os dados (nome, sobrenome, idade, telefone, sexo, email). Utilize o operador de ponto para manipular a estrutura.
2. Reescreva o programa anterior utilizando um ponteiro para manipular a estrutura.
3. Reescreva o programa anterior alocando memória dinamicamente para a estrutura.
4. Escreva um programa para ler e imprimir os resultados de uma turma de n alunos em uma disciplina (matrícula, nome, sobrenome, nota). O programa deverá informar o melhor estudante, o pior estudante e a média da turma. Considere n um valor informado pelo usuário.
5. Utilize uma estrutura para representar um ponto do plano ($x; y$), escreva um programa que leia um ponto do teclado e informe a que quadrante pertence (I, II, III ou IV). O programa ficará em execução até que o ponto $(0; 0)$ seja digitado. Desconsidere os pontos sobre os eixos coordenados. Utilize a menor quantidade de comparações possível. Sugestão: O problema pode ser resolvido verificando apenas duas condições para cada ponto digitado.
6. Modifique o programa anterior para considerar os pontos sobre os eixos coordenadas com a mensagem “Ponto sobre os eixos”. Utilize a menor quantidade de comparações possível.
7. Crie uma estrutura para representar um ponto no plano ($x; y$). Utilize a estrutura para fazer um programa que leia três pontos e informe se eles formam ou não um triângulo retângulo.
8. Crie uma estrutura para representar um ponto do espaço ($x; y; z$). Escreva um programa que utilize a estrutura para verificar se o ponto pertence ao plano $Ax + By + Cz = D$ onde os coeficientes $A; B; C$; e D são digitados pelo usuário.
9. Crie uma estrutura para representar um ponto TPonto do espaço. Utilize a estrutura TPonto para criar uma estrutura Tesfera (centro e radio). Escreva um programa usando as estruturas anteriores que receba uma esfera e um ponto, seu programa deve informar se o ponto é interno ou externo a esfera.

10. A seguinte estrutura representa um número complexo

```
typedef {
    float Re;
    float Im;
}TNumComp;
```

Escreva um programa que recebe dois números complexos e informa os resultados da soma e subtração desses dois números.

11. A seguinte estrutura representa uma sequência de números inteiros

```
TypeDef{
    int *val;
    int n;
    int max;
    int min;
    float media;
}TSeq;
```

Escreva um programa que declare uma variável de tipo Tseq e realize as seguintes operações:

- a) Recebe a quantidade de elementos da sequência n .
- b) Aloca dinamicamente um vetor para armazenar os valores da sequência (val) e lê os elementos via teclado.
- c) Calcula os atributos da sequência máximo (max), mínimo (min) e média ($media$).
- d) Mostra todos os atributos da sequência na tela.

12. Faça um programa para processar os resultados de um torneio de futebol de areia onde participaram n equipes. Para cada equipe considere: nome da equipe, jogos ganhos, jogos empatados, jogos perdidos, número de cartões amarelos, número de cartões vermelhos, goles a favor, goles em contra e saldo. Seu programa deverá informar:

- a) Nome da equipe que ganhou o torneio (jogos ganhos = 3 pontos, jogos empatados = 1 ponto). Desconsidere o caso que duas equipes alcançaram o mesmo número de pontos.
- b) Nome da equipe que teve o melhor ataque.

- c) Nome da equipe que teve a melhor defesa.
d) Nome da equipe *Fair Play*, os cartões vermelhos têm peso 2 e os amarelos peso 1.
13. Modifique o programa anterior de forma que a tabela de resultados do torneio seja impressa considerando a posição alcançada por cada equipe. Exemplo ($n = 5$)

Nome	JG	JE	JP	CA	CV	GF	GC	SG	Pontos
Copacabana	4	0	0	2	1	16	2	14	12
Botafogo	3	1	0	4	0	12	7	5	9
Leme	2	1	1	0	0	6	5	1	7
Flamingo	1	2	1	8	3	5	8	-3	5
Bangu	0	0	4	5	4	3	9	-6	0

14. Considere a seguinte estrutura para representar uma entrada de uma agenda de telefone

```
typedef struct item{
    char nome[255];
    char telefone[13];
    struct item *ptr;
}Titem;
```

Escreva um programa que receba um número arbitrário de entradas da agenda e ao terminar a leitura de dados imprima elas na mesma ordem que foram digitadas. Sugestão: Construa uma lista enlaçada segundo a seguinte figura.



15. Programa bônus Modifique o programa anterior para que a agenda de telefone seja impressa na ordem alfabética. Sugestão: Construa a lista de forma que cada vez que seja lida uma nova entrada seja colocada na posição correta (ver figura).

