## EXERCÍCIOS ESTRUTURAS, UNIÕES E ENUMERAÇÕES

1. Construa uma declaração de struct em C que possa acomodar dados provenientes de uma tabela conforme o exemplo a seguir:

| | Endereço | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Nome** | **Av., Rua, etc.** | **Núm.** | **Bairro** | **Compl.** | **CEP** | **Cidade** | **UF** | **Telefone 1** | **Telefone 2** | **Telefone 3** |
| Ruy Cerqueira | R. Bonifácio | 44 | Guarany | Apto. 101 | 21332-370 | Rio de Janeiro | RJ | 21 4527 2327 | 21 5523 0013 | |
| Gabriel Knight | Av. Castelo Rubro | 12 | Monte | | 54332-237 | Salgado | ES | 37 3232 1010 | | |
| Pedro Malaquao | R. 14 Bis | 5 | Centro | Ed. Bastos | 12003-010 | Manauara | AM | 89 5477 8700 | 89 5477 8703 | 89 5477 8710 |

2. Utilizando estrutura, fazer um programa em C que permita a entrada de nome, endereço e telefone de 5 pessoas e os imprima em ordem alfabética.
3. Fazer um programa para simular uma agenda de telefones. Para cada pessoa devem-se ter os seguintes dados:

   Nome
   E-mail

   Endereço (contendo campos para Rua, numero, complemento, bairro, cep, cidade, estado, pais).

   Telefone (contendo campo para DDD e numero)

   Data de aniversario (contendo campo para dia, mês, ano).

   **Observações**: Uma linha (string) para alguma observação especial.

(a) Definir a estrutura acima.
(b) Declarar a variável agenda (vetor) com capacidade de agendar ate 100 nomes.
(c) Definir um bloco de instruções busca por primeiro nome: Imprime os dados da pessoa com esse nome (se tiver mais de uma pessoa, imprime para todas).
(d) Definir um bloco de instruções busca por mês de aniversario: Imprime os dados de todas as pessoas que fazem aniversario nesse mês.
(e) Definir um bloco de instruções busca por dia e mês de aniversario: Imprime os dados de todas as pessoas que fazem aniversario nesse dia e mês.
(f) Definir um bloco de instruções insere pessoa: Insere por ordem alfabética de nome.
(g) Definir um bloco de instruções retira pessoa: retira todos os dados dessa pessoa e desloca todos os elementos seguintes do vetor para a posição anterior.
(h) Definir um bloco de instruções imprime agenda com as opções:

   imprime nome, telefone e e-mail.
   imprime todos os dados.

(i) O programa deve ter um menu principal oferecendo as opções acima.

3. Considerando a estrutura:

```
struct Ponto{
    int x;

    int y;
};
```

para representar um ponto em uma grade 2D, implemente um programa que indique se um ponto p esta localizado dentro ou fora de um retângulo. O retângulo e definido por seus vertices inferior esquerdo v1 e superior direito v2. A função deve imprimir caso o ponto esteja localizado dentro do retângulo ou não.

4. Considerando a estrutura

```
struct Vetor{
   float x;
   float y;
   float z;
};
```

   para representar um vetor no $R^3$, implemente um programa que calcule a soma de dois vetores.

5. Construa uma estrutura aluno com nome, numero de matrıcula e curso. Leia do usuário a informação de 5 alunos, armazene em vetor dessa estrutura e imprima os dados na tela.

6. Crie uma estrutura representando os alunos do curso de Introdução a Programação de Computadores. A estrutura deve conter a matrıcula do aluno, nome, nota da primeira prova, nota da segunda prova e nota da terceira prova.

   (a) Permita ao usuário entrar com os dados de 5 alunos.

   (b) Encontre o aluno com maior nota da primeira prova.

   (c) Encontre o aluno com maior media geral.

   (d) Encontre o aluno com menor media geral

   (e) Para cada aluno diga se ele foi aprovado ou reprovado, considerando o valor 6 para aprovação.

7. Escreva um trecho de código em “C” para fazer a criação dos novos tipos de dados conforme solicitado abaixo:

   Horário: composto de hora, minutos e segundos.
   Data: composto de dia, mês e ano.

   Compromisso: composto de uma data, horário e texto que descreve o compromisso. Byte: usado para armazenar 8 bits (definido em função dos tipos básicos do ”C”)

8. Faça um programa que armazene em um registro de dados (estrutura composta) os dados de um funcionário de uma empresa, compostos de: Nome, Idade, Sexo (M/F), CPF, Data de Nascimento, Código do Setor onde trabalha (0-99), Cargo que ocupa (string de ate 30 caracteres) e Salario. Os dados devem ser digitados pelo usuário, armazenados na estrutura e exibidos na tela.

9. Faça um programa que leia os dados de 10 alunos (Nome, matricula, Media Final), armazenando em um vetor. Uma vez lidos os dados, divida estes dados em 2 novos vetores, o vetor dos aprovados e o vetor dos reprovados, considerando a media mínima para a aprovação como sendo 5.0. Exibir na tela os dados do vetor de aprovados, seguido dos dados do vetor de reprovados.

10. Escolha um jogo de cartas, baseado em um ”baralho tradicional”(cada carta tem seu naipe e seu valor) ou tipo ”super trunfo”(cada carta possui um conjunto de atributos). Implemente a parte de distribuição (sorteio) de cartas para 2 jogadores, considerando que cada jogador ira receber 5 cartas. Exiba na tela as cartas que cada um dos jogadores recebeu.

11. Peça ao usuário para digitar seus dados pessoais (Nome, Endereço, Data de Nascimento, Cidade, CEP, email), verifique se as informações de Data de Nascimento, CEP e email

fazem sentido, e mostre ao usuário as informações, se estão todas corretas, ou mostre que alguma informação estava errada.

12. Faça um programa que leia um inteiro n e:

    Crie e leia um vetor com os dados de n carros: marca (máximo 15 letras), ano e preço.

    Leia um valor p e mostre as informações de todos os carros com preço menor que p. Repita este processo ate que seja lido um valor p = 0.

13. Faça um programa que leia um inteiro n e:

    Crie e leia um vetor com dados de n livros: tıtulo (máximo 30 letras), autor (máximo 15 letras) e ano.

    Procure um livro por tıtulo, perguntando ao usuário qual tıtulo deseja buscar. Mostre os dados de todos os livros encontrados.

14. Faça um programa com N=5 e:

    Crie e leia um vetor de alunos, sendo que cada aluno contem os dados: nome (máximo 15 letras), notas de 3 provas, media final e nıvel (inteiro). Este ultimo campo não deve ser lido agora.

    Preencha o campo nível. Seu valor deve ser igual a` parte inteira de (5*media final/media da sala).

    Na sua funçao main(), mostre o nome e o nível de cada aluno.

15. Faça um programa que seja uma agenda de compromissos e:

    Crie e leia um vetor de 5 estruturas de dados com: compromisso (máximo 60 letras) e data. A data deve ser outra estrutura de dados contendo dia, mes e ano.

    Leia dois inteiros m e a e mostre todos os compromissos do mes m do ano a. Repita o procedimento ate ler m = 0.

    Dica: use fgets(string, tamanho, stdin) para ler uma string, precedido imediatamente por fflush(stdin).

16. Faça um programa que controla o consumo de energia dos eletrodomésticos de uma casa e:

    Crie e leia 5 eletrodomésticos que contem nome (máximo 15 letras), potencia (real, em kW) e tempo ativo por dia (real, em horas).

    Leia um tempo t (em dias), calcule e mostre o consumo total na casa e o consumo relativo de cada eletrodoméstico (consumo/consumo total) nesse período de tempo. Apresente este ultimo dado em porcentagem.

17. Faça um programa que gerencie o estoque de um mercado e:

    Crie e leia um vetor de 5 produtos, com os dados: código (inteiro), nome (máximo 15 letras), preço e quantidade.

    Leia um pedido, composto por um código de produto e a quantidade. Localize este código no vetor e, se houver quantidade suficiente para atender ao pedido integralmente, atualize o estoque e informe o usuário. Repita este processo ate ler um código igual a zero.

Se por algum motivo não for possível atender ao pedido, mostre uma mensagem informando qual erro ocorreu.

18. Faça um programa que controle o fluxo de voos nos aeroportos de um país. Com v=5 (voos) e a=5 (aeroportos) e:

    Crie e leia um vetor de voos, sendo que cada voo contem um código de aeroporto de origem e um de destino.

    Crie um vetor de aeroportos, sendo que cada aeroporto contem seu código, quantidade de voos que saem e quantidade de voos que chegam.

    **Nota**: Cada aeroporto e identificado por um código inteiro entre 0 e (a-1). Não aceite aeroportos de código inexistente.

19. Faça um programa para armazenar um livro de receitas e:

    Crie um vetor de 5 receitas, que deve ter nome (máximo 25 letras), quantidade de ingredientes e ingredientes.

    Para cada receita, leia seu nome e a quantidade de ingredientes. Então crie e leia o vetor de ingredientes, sendo que cada ingrediente contem nome e quantidade.

    Procure receita por nome, mostrando seus ingredientes se encontrar. Se não encontrar, informe ao usuário. Repita o processo ate digitar uma string vazia.

20. Faça um programa que armazena filmes produzidos por vários diretores e:

    Crie e leia um vetor de 5 diretores, cada um contendo nome (máximo 20 letras), quantidade de filmes e filmes. O membro filmes e um vetor, que deve ser criado apos ter lido quantidade de filmes. Cada filme e composto por nome, ano e duração.

    Procure um diretor por nome, mostrando todos os filmes que ele já produziu. Repita o processo ate digitar uma string vazia.

21. Crie uma estrutura representando os alunos do curso de Métodos e Técnicas de Programação. A estrutura deve conter a matrícula do aluno, nome, nota da primeira prova, nota da segunda prova e nota da terceira prova.

    (a) Permita ao usuário entrar com os dados de 5 alunos
    (b) Encontre o aluno com maior nota da primeira prova
    (c) Encontre o aluno com maior media geral
    (d) Encontre o aluno com menor media geral
    (e) Para cada aluno diga se ele foi aprovado ou reprovado, considerando o valor 6 para aprovação.

22. Definir a estrutura cuja representação gráfica e dada a seguir, definir os campos com os tipos básicos necessários.

Cadastro

| NOME | ENDEREÇO | SALÁRIO |
|---|---|---|
| IDENTIDADE | CPF | ESTADO CIVIL |
| TELEFONE | IDADE | SEXO |

O ENDEREÇO é composto de:

| RUA | BAIRRO | CIDADE |
|---|---|---|
| ESTADO | CEP | |

(a) Crie um vetor Cadastro com 5 elementos.

(b) Permita ao usuário entrar com dados para preencher esse 5 cadastros.

(c) Encontre a pessoa com maior idade entre os cadastrados

(d) Encontre as pessoas do sexo masculino

(e) Encontre as pessoas com salario maior que 1000.

(f) Imprima os dados da pessoa cuja identidade seja igual a um valor fornecido pelo usuário

23. Implemente um programa em C que leia o nome, a idade e o endereço de uma pessoa e armazene os dados em uma estrutura.

24. Escreva um programa que receba dois structs do tipo dma, cada um representando uma data valida, e calcule o numero de dias que decorreram entre as duas datas.

```
struct dma {
   int dia;
   int mes;
   int ano;

};
```

25. Defina uma estrutura que irá representar bandas de música. Essa estrutura deve ter o nome da banda, que tipo de música ela toca, o número de integrantes e em que posição do ranking essa banda está dentre as suas 5 bandas favoritas.
26. Crie um looping para preencher as 5 estruturas de bandas criadas no exemplo passado. Após criar e preencher, exiba todas as informações das bandas/estruturas.
27. Crie uma função que peça ao usuário um número de 1 até 5. Em seguida, seu programa deve exibir informações da banda cuja posição no seu ranking é a que foi solicitada pelo usuário.
28. Crie uma função em C que peça ao usuário um tipo de música e exiba as bandas com esse tipo de música no seu ranking. Que função da string.h você usaria para comparar as strings que representam o tipo de banda?
29. Crie uma função que peça o nome de uma banda ao usuário e diga se ela está entre suas bandas favoritas ou não.
30. Agora junte tudo e crie uma aplicação em que exibe um menu com as opções de preencher as estruturas e todas as opções das questões passadas.
31. Seja a seguinte struct:

```
struct Sensor {
      int referencia;
      float valor_medido:
      struct Sistema *sist;
}
```

32. Complete a seguinte função com os comandos scanf apropriados.

```
void entra_dados(struct Sensor *s) {
      printf("Digite o número de referência do sensor:");
      scanf(
      printf("\nDigite o valor medido pelo sensor:");
      scanf(
}
```

33. Faça um programa que leia, do teclado, um conjunto de dados de sócios de um clube contendo para cada sócio ( matricula, nome, idade, estado_civil, grau_instrucao, turno, atividade ), considerando:
    Estado Civil pode ser:
    1 - Casado
    2 - Solteiro
    3 - Divorciado
    4 - Viúvo

    Grau de Instrução pode ser:

1 - Fundamental
2 - Secundário
3 - Superior

Turno pode ser:
1 - Manha
2 - Tarde
3 - Noite

Atividade pode ser:
1 - Natação
2 - Hidroginastica
3 - Dança
4 - Kung Fu
5 - Violao
6 - Futebol
7 - Volei

O programa deve ler o conjunto de dados e, ao final, exibir na tela:

- Total de sócios do Turno da Manha
- Total de sócios do Turno da Tarde
- Total de sócios do Turno da Noite
- Média de idade dos sócios com Grau de Instrução igual a Ensino Superior
- Média de Renda dos sócios Casados
- Total de sócios que praticam Natação
- Total de sócios que praticam Hidroginástica
- Sócio mais idoso que pratica Futebol
- Sócio mais jovem que pratica Kung Fu

Use tipos enumerados para os campos: estado_civil, grau_instrução, turno e atividade;

34. Faça um programa que leia um conjunto de dados de contratos de uma administradora de planos de saúde. Os contratos podem ser de um dos dois tipos:

    Contrato Individual:
    - Numero do Contrato
    - CPF
    - Nome
    - Idade
    - Renda

    Contrato Coletivo:
    - Numero do Contrato
    - CNPJ
    - Razao Social
    - Quantidade de Empregados Masculinos
    - Quantidade de Empregados Femininos
    - Quantidade de Dependentes
    - Idade Média
    - Renda Média

    O programa deve declarar estrutura para o Contrato Individual e para o Contrato Coletivo.

    O programa deve declarar uma union. Como um contrato só pode ser individual ou coletivo, o programa deve declarar uma union contendo ou um Contrato Individual ou um Contrato Coletivo.

    O programa deve ler o tipo de contrato que pode ser 'I' para Contrato Individual ou 'C' para Contrato Coletivo ou 'F' para Fim. O ultimo Contrato tem tipo de contrato igual a 'F'.

    Para cada contrato o programa deve exibir:
    - Numero do Contrato

- Tipo
- Renda
- Idade

Ao final do conjunto de contratos, o programa deve exibir:

- Idade média dos contratos individuais
- Maior e Menor Renda dos Contratos Individuais
- Numero do Contrato Coletivo com maior Quantidade de Empregados Masculinos
- Numero do Contrato Coletivo com menor Quantidade de Empregados Femininos

35. Considere um cadastro de produtos de um estoque, com as seguintes informações para cada produto:
    • Codigo de identificação do produto: representado por um valor inteiro
    • Nome do produto: com ate 50 caracteres
    • Quantidade disponível no estoque: representado por um número inteiro
    • Preço de venda: representado por um valor real

    (a) Defina uma estrutura em C, denominada produto, que tenha os campos apropriados para guardar as informações de um produto, conforme descrito acima.

    (b) Escreva uma função que receba os dados de um produto (código, nome, quantidade e preço) e retome o endereço de um struct produto criado dinamicamente e inicializado com os valores recebidos como parâmetros pela função. Essa função deve ter o seguinte protótipo:

    ```
    struct produto* cria (mt cod, char* nome, mt quant, float preco);
    ```

36. Um médico deseja organizar as fichas de seus pacientes. Ele precisa que você crie um programa que cadastre pacientes, com as seguintes informações:
    1 – Nome
    2 – Idade
    3 – Sexo
    4 – Peso
    5 – Altura
    6 – Dias desde a última consulta
    Após o fim dos cadastros, imprima uma lista de pacientes que estão há mais de 120 dias sem se consultar. Faça um menu que permita escolher entre cadastrar e imprimir a lista.

37. O médico deseja agora avaliar o Fator de Risco Cardiovascular de seus pacientes (FRC), o FRC é dado por:
    - IMC/(idade²)
    - Se o paciente for homem, multiplique o FRC por 0.85
    - Se o paciente for mulher, multiplique o FRC por 1.1
      Crie uma função, que dada uma estrutura do tipo Paciente (e somente esse argumento), retorne o FRC do paciente, seu nome e os dias desde a última consulta.
    - O IMC é dado por: peso/(altura)²

38. O médico, ainda preocupado com seus pacientes, pediu que você adicionasse ao programa uma função que permita marcar automaticamente consultas para o paciente, se o FRC dele for maior que 0.105 e ele não tiver se consultado nos últimos 60 dias.
    ● Uma consulta tem as seguintes informações: Paciente, Data e Número de exames a serem realizados.
    ● Percorra os pacientes do médico, e crie dinamicamente consultas, perguntando ao médico apenas a data que ele deseja marcar.
    ● **Nota**: Duas consultas não podem ser marcadas no mesmo dia.

39. A partir do trecho de código abaixo, complete o programa criando as funções que faltam.

    ```
    #include <iostream>

    typedef struct{
       int matricula;
       float notas[3];
    ```

```
}Aluno;

int main(){
    Aluno Turma[5];

    for(i=0; i< 5; i++)  {
        LeAluno(.......);  // função que lê de teclado
                           // os dados do aluno

    }
    ImprimeTurma(Turma); Função que imprime os dados de todos os alunos


    for(i=0; i< 5; i++){
        media = CalculaMediaAluno(.......);// função que calcula a média das
notas de um aluno

        printf("Aluno %d  - Media = %f\n"); // imprime os dados do aluno

    }
    return 0;
}
```

40. Crie um programa que permita armazenar o nome, a altura e da data de nascimento de até 10 pessoas.
    Cada pessoa deve ser representada por uma **struct** dentro de um vetor.
    A data de nascimento também deve ser uma **struct**.
    O nome e a altura de cada pessoa devem ser informados pelo teclado.
    A geração da data de nascimento deve ser feita aleatoriamente através da função abaixo

```
void CriaData(Data &D){
   D.Mes = 1 + (rand() % 12);
   D.Ano = 1950 + (rand() % 49);
   D.Dia = 1 + (rand() % 30);
}
```

    O programa deve, na tela de abertura, apresentar opções para:
    - inserir um nome;
    - listar todos os nomes e respectivas alturas;
    - listar os nomes das pessoas que nasceram antes de uma certa data fornecida.

    **Cada uma destas opções deve ser implementada em uma função separada.**
    Para realizar o exercício, utilize como base o programa apresentado abaixo.

```
#include <stdio.h>
#include <stdlib.h>

typedef struct {
    int dia, mes, ano;
} Data;


typedef struct {
    char nome[30];
    float altura ;
    Data nascimento;
} Pessoa;

void criaData(Data &D)
{
   D.mes = 1 + (rand() % 12);
   D.ano = 1950 + (rand() % 49);
   D.dia = 1 + (rand() % 30);
}

int main(){
    Pessoa povo[10];
    Pessoa p;
    int opcao, qtdPessoas = 0 ;
```

```
        Data dia;

        ImprimeTelaDeOpcoes();
        do {
          opcao = EscolheOpcao();
         if (opcao == 1){
            LePessoaDeTeclado(p);
            AdicionaPessoa(povo, p, qtdPessoas);
         }
         if (opcao == 2){
            ImprimeTodasAsPessoas(povo, qtdPessoas);
         }
         if (opcao == 3){
            LeDataDeTeclado(dia);
            ImprimeMaisVelhos(povo, qtdPessoas, dia);
         }
        } while (opcao != 4);
    }
```

41. Definir uma struct que represente uma lista de inteiros.

Esta struct deve armazenar os dados em um vetor de inteiros de tamanho 10.

Deve existir um **vetor auxiliar** de que indique, na posição **Vaux[i]**, qual o próximo elemento da lista após o elemento que está na posição **'i'**.

Na figura abaixo, considere que a lista inicia na posição 3.

No **vetor auxiliar**, o valor -1 indica que a posição correspondente no vetor 'dados' está vazia. Por sua vez, o valor -2 indica que a posição correspondente no **vetor Dados** é o último elemento da lista.

| **Índice** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Dados** | 10 | 1 | VAZIO | 9 | VAZIO | 11 | 34 | VAZIO | VAZIO | VAZIO | VAZIO |
| **Vetor Auxiliar(próximo)** | 5 | 6 | -1 | 0 | -1 | 1 | -2 | -1 | -1 | -1 | -1 |

Ao ser impressa, esta lista deve apresentar os seguintes valores: 9, 10, 11, 1, 34.

A classe de deve ter métodos para:

- **Inserção:** este método deve inserir um dado (int) na lista, após o último dado inserido. Este método deve retornar 1 quando conseguir inserir e 0 quando não houver mais espaço na lista. Internamente, a inserção deve ser feita no primeiro espaço livre do vetor de dados;
- **Remoção:** Remover um elemento da lista dado seu valor. Este método deve retornar 1 quando conseguir remover e 0 quando o dado não existir na lista;
- **Impressão:** Imprime os dados na lista na ordem que foram inseridos.

```
void imprimeLista(ListaEncadeadaComVetor L){
    int pos = L.inicio;
    printf("Lista:\n");
    printf("Posicao Inicial: %d\n", pos);
    for(int i=0; i< TAM; i++){
        printf("%d  (pos:%d)", L.Dados[pos], pos);
        pos = L.Aux[pos];
        if (pos == -2)
            break;
        else printf(", ");
    }
    printf("\n");

}
```

Sugestão para definição da struct:

```
#include <stdio.h>

const int TAM = 10;

typedef struct ListaEncadeadaComVetor{
    int inicio;
    int Dados[TAM];
    int Aux[TAM];

};

int achaPosUltimo(ListaEncadeadaComVetor L);// Este método faz a busca do
último a partir dos
                    // dados da lista e não a partir do vetor auxiliar
int achaPosLivre(ListaEncadeadaComVetor L); // Este método faz a busca de um
espaço lvre
                    // a partir do vetor auxiliar
int insereDado(ListaEncadeadaComVetor, int D);
int removeDado(ListaEncadeadaComVetor, int D);
void imprimeLista(ListaEncadeadaComVetor);
```

42. Considere que uma empresa precisa armazenar os seguintes dados de um cliente:
- Nome completo com no máximo 50 caracteres;
- renda mensado do cliente;
- ano de nascimento;
- possui ou não carro.

Defina um tipo e uma estrutura para armazenarem estes dados e escreva um programa que leia estes dados armazene-os em uma variável e em seguida os imprima.

43. Considerando a mesma estrutura do exercício anterior, escreva um programa que leia os dados de 100 clientes e imprima:
1. quantos clientes têm renda mensal acima da média;
2. quantos clientes têm carro;
3. quantos clientes nasceram entre 1960 (inclusive) e 1980 (exclusive).

44. Reescreva o programa abaixo (**prog 01**) empregando funções para implementar as diversas tarefas do programa. A função main deve ficar da seguinte maneira

```
 void main () {

  struct aluno turma[MAX];

   le(turma);
   puts("Imprimindo dados lidos da turma.");
   puts("Digite qualquer coisa para continuar.");
   getchar();
   imprime(turma);

   ordena_medias(turma);
   puts("Imprimindo dados ordenados da turma.");
   puts("Digite qualquer coisa para continuar.");
   getchar();
   imprime(turma);
   getchar();
}
```

**Prog 01**

```
#define MAX 4
#include <stdio.h>
#include <string.h>


void main () {
```

```
   struct aluno{
    char nome[40];
   float n1, n2, media;
   } turma[MAX], turma2[MAX];
   int i, j, pos;

   puts("Lendo dados da turma");
   for (i=0; i<MAX; i++) {
      printf("Dados do aluno %d\n", i);
      puts("Nome?"); gets(turma[i].nome);
      puts("Primeira nota?"); scanf("%f", &turma[i].n1);
      puts("Segunda nota?"); scanf("%f", &turma[i].n2);
      getchar();
      turma[i].media=(turma[i].n1+turma[i].n2)/2.0;
   }

   puts("Imprimindo dados lidos da turma.");
   puts("Digite qualquer coisa para continuar."); getchar();

   for (i=0; i<MAX; i++) {
      printf("\nDados do aluno %d\n", i);
      printf("Nome: %s\n",turma[i].nome);
      printf("Primeira nota: %.1f\n", turma[i].n1);
      printf("Segunda nota: %.1f\n", turma[i].n2);
      printf("Media: %.1f\n", turma[i].media);
   }

   for (i=0; i<MAX; i++) {
      pos = 0;
      for (j=0; j<MAX; j++)
         if (turma[i].media > turma[j].media) pos++;
      turma2[pos] = turma[i];
   }

  for (i=0; i<MAX; i++) turma[i]=turma2[i];

   puts("Imprimindo dados ordenados da turma.");
   puts("Digite qualquer coisa para continuar."); getchar();

   for (i=0; i<MAX; i++) {
      printf("\nDados do aluno %d\n", i);
      printf("Nome: %s\n",turma[i].nome);
      printf("Primeira nota: %.1f\n", turma[i].n1);
      printf("Segunda nota: %.1f\n", turma[i].n2);
      printf("Media: %.1f\n", turma[i].media);
   }

   getchar();
}
```

45. Escrever um programa que utilize structs e ponteiro para struct e imprima o conteúdo das variáveis da struct.
46. Escrever um programa que utilize enumeradores com as matérias do seu período. Inicialize cada matéria com um numero. Depois imprime os valores das variáveis enumeradas.
47. Escrever um programa que utilize union. Inicialize as variáveis com valores diferentes e imprima o conteúdo delas.
48. Fazer um programa que simule uma pilha push pop, usando structs. Um exemplo de entrada poderia ser o seguinte:

empilha C
empilha B
empilha A

desempilha A
desempilha B
desempilha C

49. Escreva um programa que solicite o nome e telefone de uma pessoa e grave essas informações num vetor de uma estrutura que contem esses dados (nome e telefone). O programa deve ter três opções apenas: uma que adiciona um novo dado, outra que lista todos os dados atualmente armazenados na memória e outra que sai do programa. Esse vetor de estrutura deve ter apenas 10 elementos e fornecer uma mensagem de erro caso o usuário tente adicionar mais pessoas que este máximo permitido.
50. Escreva a estrutura das strings (possuem um campo armazenando o tamanho da string e um ponteiro para o primeiro caracter da string) e crie as funcões strcpy e strcat para strings nesse formato.
51. Escreva um programa fazendo o uso de estruturas. Você deverá criar uma estrutura chamada Ponto, contendo apenas a posição x e y (inteiros) do ponto. Declare 2 pontos, leia a posição (coordenadas x e y) de cada um e calcule a distância entre eles. Apresente no final a distância entre os dois pontos.
52. Crie uma estrutura chamada retângulo, que possua duas estruturas ponto (o ponto superior esquerdo e o ponto inferior direito). Faça um programa que receba as informações acerca de um retângulo (as coordenadas dos dois pontos), e informe a área, o comprimento da diagonal e o comprimento de cada aresta.
53. Escreva um programa que use as mesmas estruturas do exercício anterior para descobrir se um ponto está dentro de um retângulo.
54. Considere que foi definida a seguinte estrutura:

```
typedef struct _frac{
        int numerador, denominador;
} FRACAO;
```

Escreva um programa em C que calcule as quatro operações usando frações definidas com estruturas do tipo FRACAO. O programa deve ler duas frações e imprimir o resultado de cada uma das quatro operações.