



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Pato Branco



Professora: Rúbia Eliza de Oliveira Schultz Ascari
Departamento Acadêmico de Informática (Dainf)
Tecnologia em Análise e Desenvolvimento de Sistemas
Estruturas de Dados 1

Exercícios sobre Structs

Exercícios adaptados do material do prof. André Backes
(<http://www.facom.ufu.br/~backes/gsi002.html>)

1. Implemente um programa para registrar pedidos de um restaurante. O programa deve ler o nome do cliente, descrição do pedido, e valor total e armazenar em uma struct. Depois, imprima os dados armazenados.

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

typedef struct pedido Pedido;

struct pedido {
    char nomeCliente[30];
    char descricao[100];
    float total;
};

int main(void) {
    Pedido exemploPedido;

    printf("Informe o nome do cliente: ");
    scanf("%[^\\n]s", exemploPedido.nomeCliente);

    printf("Informe a descricao do pedido: ");
    scanf("%[^\\n]s", exemploPedido.descricao);

    printf("Informe o valor total do pedido: ");
    scanf("%f", &exemploPedido.total);

    printf("\\nOs dados informados sao: \\n");

    printf("Nome Cliente: %s \\n", exemploPedido.nomeCliente);
    printf("Pedido: %s \\n", exemploPedido.descricao);
    printf("Valor: %f \\n", exemploPedido.total);

    return 0;
}
```

2. Faça um programa que leia um vetor com os dados de 5 instrumentos musicais: modelo, marca e preço. Leia um valor p e mostre as informações de todos os instrumentos com preço menor que p. Repita este processo até que seja lido um valor p = 0.

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

typedef struct instrumento Instrumento;

struct instrumento {
    char modelo[30];
    char marca[20];
    float preco;
}
```

```

};

int main(void) {
    Instrumento vetInstrumentos[5];

    for (int i = 0; i < 5; i++) {
        printf("\nInforme o modelo do instrumento musical: ");
        scanf(" %[^\\n]s", vetInstrumentos[i].modelo);

        printf("Informe a marca do instrumento musical: ");
        scanf(" %[^\\n]s", vetInstrumentos[i].marca);

        printf("Informe o preco do instrumento musical: ");
        scanf(" %f", &vetInstrumentos[i].preco);
    }

    float p = -1;
    while (p != 0) {
        printf("\nInforme um valor minimo para preco de instrumento musical ou 0 para encerrar: ");
        scanf("%f", &p);
        printf("\nInstrumentos musicais com preco menor que o valor informado: ");
        for (int i = 0; i < 5; i++) {
            if (vetInstrumentos[i].preco < p) {
                printf("\n\t%s - %s", vetInstrumentos[i].modelo, vetInstrumentos[i].marca);
            }
        }
    }

    return 0;
}

```

3. Construa uma estrutura DosesVacina para armazenar a previsão de aplicação de doses de vacina a pacientes de acordo com sua idade. Registre o nome do paciente, cidade e idade. Leia do usuário a informação de 5 pacientes, armazene em vetor dessa estrutura e imprima os dados na tela.

```

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

typedef struct paciente Paciente;

struct paciente {
    char nome[30];
    char cidade[30];
    int idade;
};

int main(void) {
    int numPacientes = 5;
    Paciente vetPacientes[numPacientes];
    int i;
    for (i = 0; i < numPacientes; i++) {
        printf("\nInforme o nome do paciente: ");
        scanf(" %[^\\n]s", vetPacientes[i].nome);

        printf("Informe a idade do paciente: ");
        scanf("%d", &vetPacientes[i].idade);
    }
}

```

```

        printf("Informe a cidade em que o paciente reside: ");
        scanf(" %[^\\n]s", vetPacientes[i].cidade);
    }

    for (i = 0; i < numPacientes; i++) {
        printf("\\nDados do paciente %d: %s, %d, %s ", i+1,
vetPacientes[i].nome, vetPacientes[i].idade, vetPacientes[i].cidade);
    }
    return 0;
}

```

4. Crie uma estrutura representando os alunos de um determinado curso. A estrutura deve conter a matrícula do aluno, nome, nota da primeira prova, nota da segunda prova e nota da terceira prova.

- Permita ao usuário entrar com os dados de 5 alunos;
- Encontre o aluno com maior nota da primeira prova;
- Encontre o aluno com maior média geral;
- Encontre o aluno com menor média geral;
- Percorra o vetor de alunos e para cada aluno diga se ele foi aprovado ou reprovado, considerando o valor 6 para aprovação.

```

#include <math.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

typedef struct aluno Aluno;

struct aluno {
    int matricula;
    char nome[30];
    float notal;
    float nota2;
    float nota3;
};

int main(void) {
    Aluno vetAlunos[5];
    int i;
    for (i = 0; i < 5; i++) {
        printf("\\nInforme o nome do aluno: ");
        scanf(" %[^\\n]s", vetAlunos[i].nome);

        printf("Informe o código de matrícula do aluno: ");
        scanf("%d", &vetAlunos[i].matricula);

        printf("Informe a nota 1 do aluno: ");
        scanf("%f", &vetAlunos[i].notal);

        printf("Informe a nota 2 do aluno: ");
        scanf("%f", &vetAlunos[i].nota2);

        printf("Informe a nota 3 do aluno: ");
        scanf("%f", &vetAlunos[i].nota3);
    }

    char alunoMaiorNotal[30];
    float maiorNotal = 0;

    char alunoMaiorMediaGeral[30];
    float maiorMediaGeral = 0;

```

```

        char alunoMenorMediaGeral[30];
        float menorMediaGeral = (vetAlunos[0].nota1 + vetAlunos[0].nota2 +
vetAlunos[0].nota3)/3.0;

        float media;
        for (i = 0; i < 5; i++) {
            media = (vetAlunos[i].nota1 + vetAlunos[i].nota2 +
vetAlunos[i].nota3)/3.0;

            if (vetAlunos[i].nota1 > maiorNota1) {
                strcpy(alunoMaiorNota1, vetAlunos[i].nome);
                maiorNota1 = vetAlunos[i].nota1;
            }

            if (media >= maiorMediaGeral) {
                strcpy(alunoMaiorMediaGeral, vetAlunos[i].nome);
                maiorMediaGeral = media;
            }

            if (media < menorMediaGeral) {
                strcpy(alunoMenorMediaGeral, vetAlunos[i].nome);
                menorMediaGeral = media;
            }

            printf("\nAluno %d: ", i+1);
            if (media >= 6.0)
                printf("Aprovado - Media: %.2f ", media);
            else
                printf("Reprovado - Media: %.2f ", media);
        }

        printf("\n\nAluno com maior Nota 1: Aluno %s - %.2f", alunoMaiorNota1,
maiorNota1);
        printf("\nAluno com maior media geral: Aluno %s - %.2f",
alunoMaiorMediaGeral, maiorMediaGeral);
        printf("\nAluno com menor media geral: Aluno %s - %.2f \n",
alunoMenorMediaGeral, menorMediaGeral);

        return 0;
    }

```

5. Faça um programa que armazene em uma estrutura os dados de um atleta profissional, compostos de: Nome, CPF, Data de Nascimento, Modalidade esportiva e Salário. Os dados devem ser digitados pelo usuário, armazenados na estrutura e exibidos na tela. Obs.: crie uma estrutura para armazenar a data de nascimento e use esta estrutura dentro da estrutura do cadastro.

```

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

typedef struct atleta Atleta;
typedef struct data Data;

struct data{
    int dia;
    int mes;
    int ano;
};

struct atleta {
    char nome[30];

```

```

    int cpf;
    Data nascimento;
    char modalidade[20];
    float salario;
};

int main(void) {
    Atleta exemploAtleta;
    printf("Informe o nome do atleta: ");
    scanf(" %[^\\n]s", exemploAtleta.nome);
    printf("\\nCPF do atleta: ");
    scanf(" %d", &exemploAtleta.cpf);

    printf("\\nInforme a data de nascimento do atleta (dd/mm/aaaa): ");
    scanf("          %d/%d/%d",          &exemploAtleta.nascimento.dia,
    &exemploAtleta.nascimento.mes, &exemploAtleta.nascimento.ano);

    printf("\\nInforme a modalidade esportiva em que o atleta atua: ");
    scanf(" %[^\\n]s", exemploAtleta.modalidade);
    printf("\\nInforme o salario do atleta: ");
    scanf(" %f", &exemploAtleta.salario);

    printf("\\n\\nDados do atleta: ");
    printf("\\n%s \\t%d \\t%d/%d/%d", exemploAtleta.nome, exemploAtleta.cpf,
    exemploAtleta.nascimento.dia,          exemploAtleta.nascimento.mes,
    exemploAtleta.nascimento.ano);
    printf("\\n%s          \\tR$          %.2f",          exemploAtleta.modalidade,
    exemploAtleta.salario);

    return 0;
}

```

6. Faça um programa que leia os dados de 10 alunos (nome, matrícula e média 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 média mínima para a aprovação como sendo 5.0. Exibir na tela os dados do vetor de aprovados, seguido dos dados do vetor de reprovados.

```

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

typedef struct aluno Aluno;

struct aluno {
    char nome[30];
    int matricula;
    float media;
};

int main(void) {
    Aluno vetAlunos[5];
    Aluno vetAprovados[5];
    Aluno vetReprovados[5];

    for (int i = 0; i < 5; i++) {
        printf("\\nInforme o nome do aluno: ");
        scanf(" %[^\\n]s", vetAlunos[i].nome);

        printf("Informe a matricula do aluno: ");
        scanf("%d", &vetAlunos[i].matricula);
    }
}

```

```

        printf("Informe a media final do aluno: ");
        scanf("%f", &vetAlunos[i].media);
    }

    int reprovados = 0;
    int aprovados = 0;
    for (int i = 0; i < 5; i++) {
        if (vetAlunos[i].media >= 5.0) {
            vetAprovados[aprovados] = vetAlunos[i];
            aprovados++;
        } else {
            vetReprovados[reprovados] = vetAlunos[i];
            reprovados++;
        }
    }

    for (int i = 0; i < aprovados; i++) {
        printf("\nAprovado: %s - %d - %.2f", vetAprovados[i].nome,
vetAprovados[i].matricula, vetAprovados[i].media);
    }

    for (int i = 0; i < reprovados; i++) {
        printf("\nReprovado: %s - %d - %.2f", vetReprovados[i].nome,
vetReprovados[i].matricula, vetReprovados[i].media);
    }

    return 0;
}

```

7. Crie uma estrutura para armazenar uma coordenada polar, composta por raio (r) e argumento (a). Crie também uma estrutura para armazenar uma coordenada do plano cartesiano, composta por x e y, sabendo que:

$$x = r * \cos(a)$$

$$y = r * \sin(a)$$

No programa principal, leia e salve um ponto em coordenada polar. Faça a conversão de coordenada polar para coordenada cartesiana e salve o resultado como um ponto do plano cartesiano. Mostre o ponto resultante.

```

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

typedef struct coordPolar Polar;
typedef struct coordCartesiana Cartesiana;

struct coordPolar {
    float raio;
    float argumento;
};

struct coordCartesiana {
    float x;
    float y;
};

int main(void) {
    Polar polar1;
    Cartesiana cart1;

    printf("Informe o raio de uma coordenada polar: ");
    scanf("%f", &polar1.raio);

    printf("Informe o argumento de uma coordenada polar: ");

```

```

scanf("%f", &polar1.argumento);

cart1.x = polar1.raio * cos(polar1.argumento);
cart1.y = polar1.raio * sin(polar1.argumento);

printf("\nA coordenada cartesiana calculada com base na coordenada
polar informada eh: ");
printf("\nx: %f \t y: %f\n", cart1.x, cart1.y);

return 0;
}

```

Obs.: Sobre coordenadas polares: <https://www.fc.unesp.br/~mauri/Down/Polares.pdf>

8. Faça um programa que leia um vetor com dados de 5 livros: título, autor e ano. Permita ao usuário:

- Cadastrar um livro;
- Buscar livros pelo título (mostrar todos os livros com o título igual ao buscado);
- Buscar livros pelo ano (mostrar todos os livros com o ano igual ao buscado);

```

#include <math.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

typedef struct livro Livro;

struct livro {
    char titulo[30];
    char autor[30];
    int ano;
};

int main(void) {
    Livro vetLivros[5];
    char tituloBusca[30];
    int anoBusca;

    int opcao = -1;
    int livrosCadastrados = 0;
    int encontrou;
    while (opcao != 3) {
        printf("\nDigite: \n\t0 para cadastrar um livro.");
        printf("\n\t1 para buscar livros pelo titulo. \n\t2 para buscar
livros pelo ano.\n\t3 para sair.\n");
        scanf("%d", &opcao);

        if (opcao == 0) { //Vai cadastrar um livro
            if (livrosCadastrados <= 5) {
                printf("\nInforme o titulo do livro: ");
                scanf(" %[^\\n]s", vetLivros[livrosCadastrados].titulo);

                printf("Informe o autor do livro: ");
                scanf(" %[^\\n]s", vetLivros[livrosCadastrados].autor);

                printf("Informe o ano do livro: ");
                scanf(" %d", &vetLivros[livrosCadastrados].ano);

                livrosCadastrados++;
            } else {
                printf("\nJa existem 5 livros cadastrados, nao e possivel
cadastrar mais.");
            }
        }
    }
}

```

```

    }
} else if (opcao == 1) { //buscar livros por titulo
    printf("\nInforme um titulo para busca: ");
    scanf(" %[^\\n]s", tituloBusca);
    encontrou = 0;
    for (int i = 0; i < livrosCadastrados; i++) {
        if (strcmp(vetLivros[i].titulo, tituloBusca) == 0) {
            printf("Livro encontrado: %s - %s - %d\\n",
vetLivros[i].titulo, vetLivros[i].autor, vetLivros[i].ano);
            encontrou = 1;
        }
    }
    if (encontrou == 0)
        printf("\\nNao localizado.");
} else if (opcao == 2) { //buscar livros por ano
    printf("\nInforme um ano para busca: ");
    scanf("%d", &anoBusca);
    encontrou = 0;
    for (int i = 0; i < livrosCadastrados; i++) {
        if (vetLivros[i].ano == anoBusca) {
            printf("Livro encontrado: %s - %s - %d\\n",
vetLivros[i].titulo, vetLivros[i].autor, vetLivros[i].ano);
            encontrou = 1;
        }
    }
    if (encontrou == 0)
        printf("\\nNao localizado.");
} else if (opcao != 3) {
    printf("Opcao invalida.");
}

return 0;
}

```

9. Escreva um programa 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.
Faça o cadastro de alguns compromissos e imprima todos os compromissos de um mês solicitado pelo usuário.

```

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

typedef struct compromisso Compromisso;
typedef struct data Data;
typedef struct hora Hora;

struct data{
    int dia;
    int mes;
    int ano;
};

struct hora{
    int hora;
    int min;
};

struct compromisso {

```



```

    Data dataC;
    Hora horaC;
    char texto[200];
};

int main(void) {
    Compromisso vetComps[5];
    int opcao = -1;
    int mesBusca;

    int compromissos = 0;
    while (opcao != 2) {
        printf("\nDigite: \n\t0 para cadastrar um compromisso.");
        printf("\n\t1 para listar compromissos de um mes.\n\t2 para sair.\n");
        scanf("%d", &opcao);

        if (opcao == 0) { //Vai cadastrar um compromisso
            if (compromissos < 5) {
                printf("\nInforme a data do compromisso (dd/mm/aaaa): ");
                scanf("%d/%d/%d", &vetComps[compromissos].dataC.dia,
                    &vetComps[compromissos].dataC.mes, &vetComps[compromissos].dataC.ano);

                printf("Informe o horario (hh:mm): ");
                scanf("%d:%d", &vetComps[compromissos].horaC.hora,
                    &vetComps[compromissos].horaC.min);

                printf("Informe a descricao: ");
                scanf(" %[^\n]s", vetComps[compromissos].texto);

                compromissos++;
            } else {
                printf("\nJa existem 5 compromissos cadastrados, nao e possivel cadastrar mais.");
            }
        } else if (opcao == 1) { //buscar livros por titulo
            printf("\nInforme um mes para busca: ");
            scanf("%d", &mesBusca);
            for (int i = 0; i < compromissos; i++) {
                if (vetComps[i].dataC.mes == mesBusca) {
                    printf("Compromisso: %d/%d/%d - %d:%d - %s \n\n",
                        vetComps[i].dataC.dia, vetComps[i].dataC.mes,
                        vetComps[i].dataC.ano, vetComps[i].horaC.hora, vetComps[i].horaC.min,
                        vetComps[i].texto);
                }
            }
        } else if (opcao != 2) {
            printf("Opcao invalida.");
        }
    }

    return 0;
}

```

10. Criar uma estrutura para armazenar dados de filmes: título, diretor, gênero, duração (em minutos) e nota (1, 2, 3, 4 ou 5 estrelas).

- Leia e armazene informações de alguns filmes;
- Crie uma função para retornar o cadastro do filme mais longo.
- Crie uma função para mostrar os dados dos filmes com 5 estrelas.
- Crie uma função para mostrar os dados dos filmes com apenas 1 estrela.
- Crie uma função para mostrar todos os filmes de um determinado diretor.

- Crie uma função para mostrar todos os filmes de um determinado gênero.

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

typedef struct filme Filme;

struct filme{
    char titulo[30];
    char diretor[30];
    char genero[20];
    int duracao; //minutos
    int nota; //(1,2,3,4 ou 5 estrelas)
};

void filmeMaisLongo(Filme vetFilmes[3]) {
    int maiorDuracao = 0;
    int idFilme = 0;
    for (int i = 0; i < 3; i++) {
        if (maiorDuracao < vetFilmes[i].duracao) {
            maiorDuracao = vetFilmes[i].duracao;
            idFilme = i;
        }
    }
    printf("\n\nFilme com maior duracao: %s - %d",
vetFilmes[idFilme].titulo, maiorDuracao);
}

void listaFilmesNota(Filme vetFilmes[3], int nota) {
    printf("\nFilmes com nota igual a %d estrela(s): \n", nota);
    for (int i = 0; i < 3; i++) {
        if (vetFilmes[i].nota == nota) {
            printf("\t%s de %s\n", vetFilmes[i].titulo,
vetFilmes[i].diretor);
        }
    }
}

void listaFilmes(Filme vetFilmes[3], int filtro, char texto[30]) {
    for (int i = 0; i < 3; i++) {
        if (filtro == 0) {
            if (strcmp(vetFilmes[i].titulo, texto) == 0) {
                printf("\t%s de %s\n", vetFilmes[i].titulo,
vetFilmes[i].diretor);
            }
        } else if (filtro == 1) {
            if (strcmp(vetFilmes[i].diretor, texto) == 0) {
                printf("\t%s de %s\n", vetFilmes[i].titulo,
vetFilmes[i].diretor);
            }
        }
    }
}

int main(void) {
    Filme vetFilmes[3];

    for (int i = 0; i < 3; i++) {
        printf("\nInforme o titulo, diretor, genero, duracao (em minutos)
e nota (1,2,3,4 ou 5 estrelas) do filme: ");
    }
}
```

```
        scanf("%s %s %s %d %d", vetFilmes[i].titulo, vetFilmes[i].diretor,
vetFilmes[i].genero, &vetFilmes[i].duracao, &vetFilmes[i].nota);
    }

    filmeMaisLongo(vetFilmes);

    listaFilmesNota(vetFilmes, 5);
    listaFilmesNota(vetFilmes, 1);

    printf("\nFilmes com titulo filmeTeste: \n");
    listaFilmes(vetFilmes, 0, "filmeTeste");

    printf("\nFilmes do diretor diretorTeste: \n");
    listaFilmes(vetFilmes, 1, "diretorTeste");

    return 0;
}
```