Ex 01:

#include <stdio.h>

#include <string.h>

#include <math.h>

struct horario

{

int hora, minuto, segundo;

};

struct data

{

int dia, mes, ano;

};

struct compromisso

{

struct data data\_compromisso;

struct horario horario\_compromisso;

char texto[100];

};

Ex02:

#include <stdio.h>

#include <string.h>

#include <math.h>

struct horario

{

int hora, minuto, segundo;

};

struct data

{

int dia, mes, ano;

};

struct compromisso

{

struct data data\_compromisso;

struct horario horario\_compromisso;

char texto[100];

};

Ex03:

#include <stdio.h>

#include <string.h>

#include <math.h>

struct aluno

{

char nome[100], curso[100];

int matricula;

};

int main()

{

struct aluno alunos[5];

int i;

for(i=0; i<5; i++)

{

printf("Aluno %D:\n", i + 1);

printf("Nome: ");

fgets(alunos[i].nome, 100, stdin);

setbuf(stdin, NULL);

printf("Numero de Matricula: ");

scanf("%d", &alunos[i].matricula);

setbuf(stdin, NULL);

printf("Curso: ");

fgets(alunos[i].curso, 100, stdin);

setbuf(stdin, NULL);

}

printf("Dados dos Alunos:\n");

for(i=0; i<5; i++)

{

printf("Aluno %d:\n", i + 1);

printf("Nome: %s", alunos[i].nome);

printf("Numero de Matricula: %d\n", alunos[i].matricula);

printf("Curso: %s", alunos[i].curso);

}

return 0;

}

Ex04:

#include <stdio.h>

#include <string.h>

#include <math.h>

struct aluno

{

char nome[100], matricula[100];

float nota[3];

};

int main()

{

struct aluno alunos[5];

int i, j, k, l, m;

float media[5], m\_maior, m\_menor, n\_maior;

for(i=0; i<5; i++)

{

media[i] = 0;

}

for(i=0; i<5; i++)

{

printf("\nAluno %d: ", i + 1);

printf("Nome: ");

setbuf(stdin, NULL);

fgets(alunos[i].nome, 100, stdin);

printf("Matricula: ");

setbuf(stdin, NULL);

fgets(alunos[i].matricula, 100, stdin);

for(m=0; m<3; m++)

{

printf("Digite a nota %d: ", m+1);

scanf("%f", &alunos[i].nota[m]);

media[i] += alunos[i].nota[m];

}

}

for(i=0; i<5; i++)

{

media[i] /= 3;

}

m\_maior = media[0];

m\_menor = media[0];

n\_maior = alunos[0].nota[0];

for(i=0; i<5; i++)

{

if(media[i]>=6)

{

printf("Aluno %d aprovado\n", i + 1);

}

if(media[i] > m\_maior)

{

m\_maior = media[i];

j = i;

}

if(media[i] < m\_menor)

{

m\_menor = media[i];

k = i;

}

if(alunos[i].nota[0] > n\_maior)

{

n\_maior = alunos[i].nota[1];

l = i;

}

}

printf("Aluno com maior nota na primeira prova: %d\n", l +1);

printf("Aluno com maior media: %d\n", j + 1);

printf("Aluno com menor media: %d\n", k + 1);

return 0;

}

Ex05:

#include <stdio.h>

struct vetor {

float a;

float b;

float c;

};

struct vetor soma\_vetores(struct vetor v1, struct vetor v2) {

struct vetor resultado;

resultado.a = v1.a + v2.a;

resultado.b = v1.b + v2.b;

resultado.c = v1.c + v2.c;

return resultado;

}

int main() {

struct vetor vetor1, vetor2, soma;

printf("Digite as coordenadas do vetor 1:\n");

printf("a: ");

scanf("%f", &vetor1.a);

printf("b: ");

scanf("%f", &vetor1.b);

printf("c: ");

scanf("%f", &vetor1.c);

printf("Digite as coordenadas do vetor 2:\n");

printf("a: ");

scanf("%f", &vetor2.a);

printf("b: ");

scanf("%f", &vetor2.b);

printf("c: ");

scanf("%f", &vetor2.c);

soma = soma\_vetores(vetor1, vetor2);

printf("A soma dos vetores e: (%.2f, %.2f, %.2f)\n", soma.a, soma.b, soma.c);

return 0;

}

Ex06:

#include <stdio.h>

#include <string.h>

#define MAX\_ALUNOS 10

struct aluno {

int matricula;

char nome[50];

int codigo\_disciplina;

float nota1;

float nota2;

};

float calcular\_media\_ponderada(float nota1, float nota2) {

return (nota1 \* 1.0 + nota2 \* 2.0) / 3.0;

}

int main() {

struct aluno turma[MAX\_ALUNOS];

int num\_alunos;

printf("Digite o numero de alunos (ate %d): ", MAX\_ALUNOS);

scanf("%d", &num\_alunos);

if (num\_alunos > MAX\_ALUNOS) {

printf("Numero de alunos excede o limite.\n");

return 0;

}

printf("Digite os dados dos alunos:\n");

for (int i = 0; i < num\_alunos; i++) {

printf("Aluno %d:\n", i + 1);

printf("Matricula: ");

scanf("%d", &turma[i].matricula);

printf("Nome: ");

getchar();

fgets(turma[i].nome, sizeof(turma[i].nome), stdin);

turma[i].nome[strcspn(turma[i].nome, "\n")] = '\0';

printf("Codigo da Disciplina: ");

scanf("%d", &turma[i].codigo\_disciplina);

printf("Nota 1: ");

scanf("%f", &turma[i].nota1);

printf("Nota 2: ");

scanf("%f", &turma[i].nota2);

printf("\n");

}

printf("Listagem final dos alunos:\n");

for (int i = 0; i < num\_alunos; i++) {

float media = calcular\_media\_ponderada(turma[i].nota1, turma[i].nota2);

printf("Aluno %d\n", i + 1);

printf("Matricula: %d\n", turma[i].matricula);

printf("Nome: %s\n", turma[i].nome);

printf("Codigo da Disciplina: %d\n", turma[i].codigo\_disciplina);

printf("Media Final: %.2f\n\n", media);

}

return 0;

}

Ex07:

#include <stdio.h>

#include <math.h>

struct ponto\_polar {

float raio;

float argumento;

};

struct ponto\_cartesiano {

float x;

float y;

};

struct ponto\_cartesiano converter\_para\_cartesiano(struct ponto\_polar polar) {

struct ponto\_cartesiano cartesiano;

cartesiano.x = polar.raio \* cos(polar.argumento);

cartesiano.y = polar.raio \* sin(polar.argumento);

return cartesiano;

}

int main() {

struct ponto\_polar polar;

struct ponto\_cartesiano cartesiano;

printf("Digite as coordenadas polares (raio e argumento em radianos):\n");

printf("Raio: ");

scanf("%f", &polar.raio);

printf("Argumento: ");

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

cartesiano = converter\_para\_cartesiano(polar);

printf("Coordenadas cartesianas:\n");

printf("x: %.2f\n", cartesiano.x);

printf("y: %.2f\n", cartesiano.y);

return 0;

}

Ex08:

#include <stdio.h>

#define MAX\_CARACTERES 30

struct funcionario {

char nome[MAX\_CARACTERES];

int idade;

char sexo;

char cpf[12];

char data\_nascimento[11];

int codigo\_setor;

char cargo[MAX\_CARACTERES];

float salario;

};

int main() {

struct funcionario f;

printf("Digite os dados do funcionario:\n");

printf("Nome: ");

scanf(" %[^\n]s", f.nome);

printf("Idade: ");

scanf("%d", &f.idade);

printf("Sexo (M/F): ");

scanf(" %c", &f.sexo);

printf("CPF: ");

scanf(" %[^\n]s", f.cpf);

printf("Data de Nascimento (dd/mm/aaaa): ");

scanf(" %[^\n]s", f.data\_nascimento);

printf("Codigo do Setor (0-99): ");

scanf("%d", &f.codigo\_setor);

printf("Cargo: ");

scanf(" %[^\n]s", f.cargo);

printf("Salario: ");

scanf("%f", &f.salario);

printf("\nDados do funcionario:\n");

printf("Nome: %s\n", f.nome);

printf("Idade: %d\n", f.idade);

printf("Sexo: %c\n", f.sexo);

printf("CPF: %s\n", f.cpf);

printf("Data de Nascimento: %s\n", f.data\_nascimento);

printf("Codigo do Setor: %d\n", f.codigo\_setor);

printf("Cargo: %s\n", f.cargo);

printf("Salario: %.2f\n", f.salario);

return 0;

}

Ex09:

#include <stdio.h>

#include <math.h>

struct numero\_complexo {

float parte\_real;

float parte\_imaginaria;

};

struct numero\_complexo somar\_numeros\_complexos(struct numero\_complexo z, struct numero\_complexo w) {

struct numero\_complexo resultado;

resultado.parte\_real = z.parte\_real + w.parte\_real;

resultado.parte\_imaginaria = z.parte\_imaginaria + w.parte\_imaginaria;

return resultado;

}

struct numero\_complexo subtrair\_numeros\_complexos(struct numero\_complexo z, struct numero\_complexo w) {

struct numero\_complexo resultado;

resultado.parte\_real = z.parte\_real - w.parte\_real;

resultado.parte\_imaginaria = z.parte\_imaginaria - w.parte\_imaginaria;

return resultado;

}

struct numero\_complexo multiplicar\_numeros\_complexos(struct numero\_complexo z, struct numero\_complexo w) {

struct numero\_complexo resultado;

resultado.parte\_real = z.parte\_real \* w.parte\_real - z.parte\_imaginaria \* w.parte\_imaginaria;

resultado.parte\_imaginaria = z.parte\_real \* w.parte\_imaginaria + z.parte\_imaginaria \* w.parte\_real;

return resultado;

}

float calcular\_modulo(struct numero\_complexo z) {

return sqrt(pow(z.parte\_real, 2) + pow(z.parte\_imaginaria, 2));

}

int main() {

struct numero\_complexo z, w, resultado;

printf("Digite o primeiro numero complexo:\n");

printf("Parte Real: ");

scanf("%f", &z.parte\_real);

printf("Parte Imaginaria: ");

scanf("%f", &z.parte\_imaginaria);

printf("Digite o segundo numero complexo:\n");

printf("Parte Real: ");

scanf("%f", &w.parte\_real);

printf("Parte Imaginaria: ");

scanf("%f", &w.parte\_imaginaria);

resultado = somar\_numeros\_complexos(z, w);

printf("\nSoma: %.2f + %.2fi\n", resultado.parte\_real, resultado.parte\_imaginaria);

resultado = subtrair\_numeros\_complexos(z, w);

printf("Subtracao: %.2f + %.2fi\n", resultado.parte\_real, resultado.parte\_imaginaria);

resultado = multiplicar\_numeros\_complexos(z, w);

printf("Multiplicacao: %.2f + %.2fi\n", resultado.parte\_real, resultado.parte\_imaginaria);

printf("\nModulo do primeiro numero complexo: %.2f\n", calcular\_modulo(z));

printf("Modulo do segundo numero complexo: %.2f\n", calcular\_modulo(w));

return 0;

}

Ex10:

#include <stdio.h>

#include <string.h>

#define MAX\_PESSOAS 5

#define MAX\_CARACTERES 100

struct pessoa {

char nome[MAX\_CARACTERES];

char endereco[MAX\_CARACTERES];

char telefone[MAX\_CARACTERES];

};

void ordenar\_pessoas(struct pessoa pessoas[], int num\_pessoas) {

int i, j;

struct pessoa temp;

for (i = 0; i < num\_pessoas - 1; i++) {

for (j = i + 1; j < num\_pessoas; j++) {

if (strcmp(pessoas[i].nome, pessoas[j].nome) > 0) {

temp = pessoas[i];

pessoas[i] = pessoas[j];

pessoas[j] = temp;

}

}

}

}

int main() {

struct pessoa pessoas[MAX\_PESSOAS];

int i;

printf("Digite os dados de 5 pessoas:\n");

for (i = 0; i < MAX\_PESSOAS; i++) {

printf("Pessoa %d:\n", i + 1);

printf("Nome: ");

scanf(" %[^\n]s", pessoas[i].nome);

printf("Endereco: ");

scanf(" %[^\n]s", pessoas[i].endereco);

printf("Telefone: ");

scanf(" %[^\n]s", pessoas[i].telefone);

printf("\n");

}

ordenar\_pessoas(pessoas, MAX\_PESSOAS);

printf("Pessoas em ordem alfabetica:\n");

for (i = 0; i < MAX\_PESSOAS; i++) {

printf("Nome: %s\n", pessoas[i].nome);

printf("Endereco: %s\n", pessoas[i].endereco);

printf("Telefone: %s\n", pessoas[i].telefone);

printf("\n");

}

return 0;

}

Ex11:

#include <stdio.h>

#define MAX\_ALUNOS 10

#define TAM\_NOME 100

struct aluno {

char nome[TAM\_NOME];

int matricula;

float media\_final;

};

int main() {

struct aluno alunos[MAX\_ALUNOS];

struct aluno aprovados[MAX\_ALUNOS], reprovados[MAX\_ALUNOS];

int i, num\_aprovados = 0, num\_reprovados = 0;

printf("Digite os dados dos 10 alunos:\n");

for (i = 0; i < MAX\_ALUNOS; i++) {

printf("Aluno %d:\n", i + 1);

printf("Nome: ");

scanf(" %[^\n]s", alunos[i].nome);

printf("Matricula: ");

scanf("%d", &alunos[i].matricula);

printf("Media Final: ");

scanf("%f", &alunos[i].media\_final);

printf("\n");

}

for (i = 0; i < MAX\_ALUNOS; i++) {

if (alunos[i].media\_final >= 5.0) {

aprovados[num\_aprovados] = alunos[i];

num\_aprovados++;

} else {

reprovados[num\_reprovados] = alunos[i];

num\_reprovados++;

}

}

printf("Alunos Aprovados:\n");

for (i = 0; i < num\_aprovados; i++) {

printf("Nome: %s\n", aprovados[i].nome);

printf("Matricula: %d\n", aprovados[i].matricula);

printf("Media Final: %.2f\n", aprovados[i].media\_final);

printf("\n");

}

printf("Alunos Reprovados:\n");

for (i = 0; i < num\_reprovados; i++) {

printf("Nome: %s\n", reprovados[i].nome);

printf("Matricula: %d\n", reprovados[i].matricula);

printf("Media Final: %.2f\n", reprovados[i].media\_final);

printf("\n");

}

return 0;

}

Ex12:

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define NUM\_JOGADORES 2

#define NUM\_CARTAS 3

#define NUM\_NAIPES 4

#define NUM\_VALORES 13

struct carta {

int naipe;

int valor;

};

void inicializar\_baralho(struct carta baralho[]) {

int i, j, k = 0;

for (i = 0; i < NUM\_NAIPES; i++) {

for (j = 0; j < NUM\_VALORES; j++) {

baralho[k].naipe = i;

baralho[k].valor = j;

k++;

}

}

}

void embaralhar\_cartas(struct carta baralho[]) {

int i, j;

struct carta temp;

srand(time(NULL));

for (i = 0; i < 100; i++) {

int carta1 = rand() % 52;

int carta2 = rand() % 52;

temp = baralho[carta1];

baralho[carta1] = baralho[carta2];

baralho[carta2] = temp;

}

}

void distribuir\_cartas(struct carta baralho[], struct carta jogador[][NUM\_CARTAS]) {

int i, j, k = 0;

for (i = 0; i < NUM\_CARTAS; i++) {

for (j = 0; j < NUM\_JOGADORES; j++) {

jogador[j][i] = baralho[k++];

}

}

}

void exibir\_cartas(struct carta jogador[][NUM\_CARTAS]) {

int i, j;

for (i = 0; i < NUM\_JOGADORES; i++) {

printf("Jogador %d:\n", i + 1);

for (j = 0; j < NUM\_CARTAS; j++) {

printf("Naipe: %d | Valor: %d\n", jogador[i][j].naipe, jogador[i][j].valor);

}

printf("\n");

}

}

int main() {

struct carta baralho[52];

struct carta jogador[NUM\_JOGADORES][NUM\_CARTAS];

inicializar\_baralho(baralho);

embaralhar\_cartas(baralho);

distribuir\_cartas(baralho, jogador);

exibir\_cartas(jogador);

return 0;

}

Ex14:

#include <stdio.h>

#include <string.h>

#define TAM\_MARCA 16

#define NUM\_CARROS 5

struct carro {

char marca[TAM\_MARCA];

int ano;

float preco;

};

void ler\_carros(struct carro carros[]) {

int i;

for (i = 0; i < NUM\_CARROS; i++) {

printf("Carro %d:\n", i + 1);

printf("Marca: ");

fgets(carros[i].marca, TAM\_MARCA, stdin);

carros[i].marca[strcspn(carros[i].marca, "\n")] = '\0';

printf("Ano: ");

scanf("%d", &carros[i].ano);

printf("Preco: ");

scanf("%f", &carros[i].preco);

// Limpar o buffer do teclado

getchar();

printf("\n");

}

}

void exibir\_carros\_com\_preco\_menor(struct carro carros[], float p) {

int i;

int num\_carros\_exibidos = 0;

printf("Carros com preco menor que %.2f:\n", p);

for (i = 0; i < NUM\_CARROS; i++) {

if (carros[i].preco < p) {

printf("Carro %d:\n", i + 1);

printf("Marca: %s\n", carros[i].marca);

printf("Ano: %d\n", carros[i].ano);

printf("Preco: %.2f\n\n", carros[i].preco);

num\_carros\_exibidos++;

}

}

if (num\_carros\_exibidos == 0) {

printf("Nenhum carro encontrado com preco menor que %.2f.\n\n", p);

}

}

int main() {

struct carro carros[NUM\_CARROS];

float p;

ler\_carros(carros);

while (1) {

printf("Digite um valor p (0 para sair): ");

scanf("%f", &p);

if (p == 0) {

break;

}

exibir\_carros\_com\_preco\_menor(carros, p);

}

return 0;

}

Ex15:

#include <stdio.h>

#include <string.h>

#define TAM\_TITULO 31

#define TAM\_AUTOR 16

#define NUM\_LIVROS 5

struct livro {

char titulo[TAM\_TITULO];

char autor[TAM\_AUTOR];

int ano;

};

void ler\_livros(struct livro livros[]) {

int i;

for (i = 0; i < NUM\_LIVROS; i++) {

printf("Livro %d:\n", i + 1);

printf("Título: ");

fgets(livros[i].titulo, TAM\_TITULO, stdin);

livros[i].titulo[strcspn(livros[i].titulo, "\n")] = '\0';

printf("Autor: ");

fgets(livros[i].autor, TAM\_AUTOR, stdin);

livros[i].autor[strcspn(livros[i].autor, "\n")] = '\0';

printf("Ano: ");

scanf("%d", &livros[i].ano);

// Limpar o buffer do teclado

getchar();

printf("\n");

}

}

void buscar\_livro\_por\_titulo(struct livro livros[], char titulo\_busca[]) {

int i;

int num\_livros\_encontrados = 0;

printf("Livros encontrados com o título '%s':\n", titulo\_busca);

for (i = 0; i < NUM\_LIVROS; i++) {

if (strcmp(livros[i].titulo, titulo\_busca) == 0) {

printf("Livro %d:\n", i + 1);

printf("Título: %s\n", livros[i].titulo);

printf("Autor: %s\n", livros[i].autor);

printf("Ano: %d\n\n", livros[i].ano);

num\_livros\_encontrados++;

}

}

if (num\_livros\_encontrados == 0) {

printf("Nenhum livro encontrado com o título '%s'.\n\n", titulo\_busca);

}

}

int main() {

struct livro livros[NUM\_LIVROS];

char titulo\_busca[TAM\_TITULO];

ler\_livros(livros);

printf("Digite o título do livro que deseja buscar: ");

fgets(titulo\_busca, TAM\_TITULO, stdin);

titulo\_busca[strcspn(titulo\_busca, "\n")] = '\0';

buscar\_livro\_por\_titulo(livros, titulo\_busca);

return 0;

}

Ex16:

#include <stdio.h>

#include <string.h>

#define TAM\_COMPROMISSO 61

#define NUM\_COMPROMISSOS 5

struct data {

int dia;

int mes;

int ano;

};

struct compromisso {

char descricao[TAM\_COMPROMISSO];

struct data data\_compromisso;

};

void ler\_compromissos(struct compromisso compromissos[]) {

int i;

for (i = 0; i < NUM\_COMPROMISSOS; i++) {

printf("Compromisso %d:\n", i + 1);

printf("Descrição: ");

fgets(compromissos[i].descricao, TAM\_COMPROMISSO, stdin);

compromissos[i].descricao[strcspn(compromissos[i].descricao, "\n")] = '\0';

printf("Data (dia mes ano): ");

scanf("%d %d %d", &compromissos[i].data\_compromisso.dia, &compromissos[i].data\_compromisso.mes, &compromissos[i].data\_compromisso.ano);

// Limpar o buffer do teclado

getchar();

printf("\n");

}

}

void mostrar\_compromissos(struct compromisso compromissos[], int mes, int ano) {

int i;

int num\_compromissos\_encontrados = 0;

printf("Compromissos do mês %d do ano %d:\n", mes, ano);

for (i = 0; i < NUM\_COMPROMISSOS; i++) {

if (compromissos[i].data\_compromisso.mes == mes && compromissos[i].data\_compromisso.ano == ano) {

printf("Compromisso %d:\n", i + 1);

printf("Descrição: %s\n", compromissos[i].descricao);

printf("Data: %d/%d/%d\n\n", compromissos[i].data\_compromisso.dia, compromissos[i].data\_compromisso.mes, compromissos[i].data\_compromisso.ano);

num\_compromissos\_encontrados++;

}

}

if (num\_compromissos\_encontrados == 0) {

printf("Nenhum compromisso encontrado para o mês %d do ano %d.\n\n", mes, ano);

}

}

int main() {

struct compromisso compromissos[NUM\_COMPROMISSOS];

int mes, ano;

ler\_compromissos(compromissos);

while (1) {

printf("Digite o mês e o ano (M A) dos compromissos que deseja visualizar (0 0 para sair): ");

scanf("%d %d", &mes, &ano);

if (mes == 0 && ano == 0) {

break;

}

mostrar\_compromissos(compromissos, mes, ano);

}

return 0;

}

Ex17:

#include <stdio.h>

#include <string.h>

#define TAM\_NOME 16

#define NUM\_ELETRODOMESTICOS 5

struct eletrodomestico {

char nome[TAM\_NOME];

float potencia;

float tempo\_ativo;

};

void ler\_eletrodomesticos(struct eletrodomestico eletrodomesticos[]) {

int i;

for (i = 0; i < NUM\_ELETRODOMESTICOS; i++) {

printf("Eletrodoméstico %d:\n", i + 1);

printf("Nome: ");

fgets(eletrodomesticos[i].nome, TAM\_NOME, stdin);

eletrodomesticos[i].nome[strcspn(eletrodomesticos[i].nome, "\n")] = '\0';

printf("Potência (em kW): ");

scanf("%f", &eletrodomesticos[i].potencia);

printf("Tempo ativo por dia (em horas): ");

scanf("%f", &eletrodomesticos[i].tempo\_ativo);

// Limpar o buffer do teclado

getchar();

printf("\n");

}

}

void calcular\_consumo(struct eletrodomestico eletrodomesticos[], int num\_dias) {

int i;

float consumo\_total = 0.0;

for (i = 0; i < NUM\_ELETRODOMESTICOS; i++) {

consumo\_total += eletrodomesticos[i].potencia \* eletrodomesticos[i].tempo\_ativo \* num\_dias;

}

printf("Consumo total na casa: %.2f kWh\n", consumo\_total);

printf("Consumo relativo de cada eletrodoméstico:\n");

for (i = 0; i < NUM\_ELETRODOMESTICOS; i++) {

float consumo\_relativo = (eletrodomesticos[i].potencia \* eletrodomesticos[i].tempo\_ativo \* num\_dias) / consumo\_total;

printf("Eletrodoméstico %d (%s): %.2f%%\n", i + 1, eletrodomesticos[i].nome, consumo\_relativo \* 100);

}

}

int main() {

struct eletrodomestico eletrodomesticos[NUM\_ELETRODOMESTICOS];

int num\_dias;

ler\_eletrodomesticos(eletrodomesticos);

printf("Digite o número de dias para calcular o consumo: ");

scanf("%d", &num\_dias);

calcular\_consumo(eletrodomesticos, num\_dias);

return 0;

}

Ex18:

#include <stdio.h>

#define V 5 // Quantidade de voos

#define A 5 // Quantidade de aeroportos

struct voo {

int origem;

int destino;

};

struct aeroporto {

int codigo;

int voos\_saida;

int voos\_chegada;

};

void ler\_voos(struct voo voos[]) {

int i;

for (i = 0; i < V; i++) {

printf("Voo %d:\n", i + 1);

printf("Código do aeroporto de origem: ");

scanf("%d", &voos[i].origem);

printf("Código do aeroporto de destino: ");

scanf("%d", &voos[i].destino);

printf("\n");

}

}

void criar\_aeroportos(struct voo voos[], struct aeroporto aeroportos[]) {

int i;

for (i = 0; i < A; i++) {

aeroportos[i].codigo = i;

aeroportos[i].voos\_saida = 0;

aeroportos[i].voos\_chegada = 0;

}

for (i = 0; i < V; i++) {

int origem = voos[i].origem;

int destino = voos[i].destino;

if (origem >= 0 && origem < A && destino >= 0 && destino < A) {

aeroportos[origem].voos\_saida++;

aeroportos[destino].voos\_chegada++;

}

}

}

int main() {

struct voo voos[V];

struct aeroporto aeroportos[A];

int i;

ler\_voos(voos);

criar\_aeroportos(voos, aeroportos);

printf("Relatório de aeroportos:\n");

for (i = 0; i < A; i++) {

printf("Aeroporto %d:\n", aeroportos[i].codigo);

printf("Voos de saída: %d\n", aeroportos[i].voos\_saida);

printf("Voos de chegada: %d\n", aeroportos[i].voos\_chegada);

printf("\n");

}

return 0;

}

Ex20:

#include <stdio.h>

struct dma {

int dia;

int mes;

int ano;

};

int calcular\_dias(struct dma data1, struct dma data2) {

int total\_dias1 = data1.dia + 30 \* (data1.mes - 1) + 365 \* (data1.ano - 1);

int total\_dias2 = data2.dia + 30 \* (data2.mes - 1) + 365 \* (data2.ano - 1);

return total\_dias2 - total\_dias1;

}

int main() {

struct dma data1, data2;

printf("Digite a primeira data (dia mes ano): ");

scanf("%d %d %d", &data1.dia, &data1.mes, &data1.ano);

printf("Digite a segunda data (dia mes ano): ");

scanf("%d %d %d", &data2.dia, &data2.mes, &data2.ano);

int num\_dias = calcular\_dias(data1, data2);

printf("O número de dias que decorreram entre as duas datas é: %d\n", num\_dias);

return 0;

}