

Os exemplos aqui apresentados referem-se aos “Slides Arquivos de Texto” postados no Moodle.

EXEMPLO 1:

```
// EXEMPLOS DE USO DE ARQUIVOS DE TEXTO
// Gravação dos valores de um vetor
// Obs.: arquivo será criado na mesma pasta do programa
#include <stdio.h>

int main(){
    float v[] = { 1.5, 2.75, -7, 0, 45.99 };
    int i;
    FILE *f = fopen( "exemplo1.txt","wt");
    if( f == NULL){
        printf("Erro na abertura do arquivo!\n");
        return -1;
    }
    for( i = 0 ; i < 5 ; i++ )
        fprintf( f, "%.3f\n", v[i] );
    printf("Valores gravados!\n");
    fclose( f );
    return 0;
}
```

EXEMPLO 2:

```
// EXEMPLOS DE USO DE ARQUIVOS DE TEXTO
// Leitura dos valores gravados
// Obs.: arquivo lido deve estar na mesma pasta do programa
#include <stdio.h>

int main(){
    float valor;
    int cont = 0;
    FILE *f = fopen( "exemplo1.txt","rt" );
    if( f == NULL){
        printf("Erro na abertura do arquivo!\n");
        return -1;
    }
    int x = fscanf( f, "%f", &valor );
    while( x == 1 ){
        cont++;
        printf( "%do. valor: %.3f\n", cont, valor );
        x = fscanf( f, "%f", &valor );
    }
    fclose( f );
    printf("Foram lidos %d valores.\n", cont);
    return 0;
}
```

EXEMPLO 3:

```
// EXEMPLOS DE USO DE ARQUIVOS DE TEXTO
// Leitura de um arquivo e gravação em outro
/* Arquivo de entrada contendo 3 valores float
por linha; programa lê arquivo e gera como
saída outro arquivo contendo, por linha, a
média dos 3 valores */
// Aqui foi feita também a GRAVAÇÃO dos valores float (exemplo não pedia)
#include <stdio.h>

int main(){
    // temperaturas em 3 horários distintos do dia de 4 cidades no planeta
    float v[4][3] = { { 15.5, 22.75, 13.2},
                      { 0, 8.3, -7 },
                      { 22.8, 33.5, 20 },
                      { -13.1, 0.4, -5.1} };

    // *** 1a. etapa: gravando dados num arquivo ***
    int i,j;
    FILE *f1 = fopen( "temperaturas.txt","wt"); // f1 irá referenciar arq.gravação
    if( f1 == NULL){
        printf("Erro na abertura do arquivo!\n");
        return -1;
    }
    for( i = 0 ; i < 4 ; i++ ) {
        for( j = 0 ; j < 3 ; j++ ) {
            fprintf( f1, "%.3f\t", v[i][j] );
        }
        fprintf( f1, "\n");
    }
    printf("Valores gravados!\n");
    fclose(f1); // tarefa de gravação encerrada, arquivo pode ser fechado

    // *** 2a. etapa: lendo dados num arquivo e gravando em outro ***
    int x,cont=0,quant=0;
    float temp[3],media=0; // vetor para guardar 3 temperaturas simultaneamente
    // abre arquivo de temperaturas para LEITURA
    f1 = fopen( "temperaturas.txt","rt" ); // f2 irá referenciar o arq.de leitura
    if( f1 == NULL){
        printf("Erro na abertura do arquivo!\n");
        return -1;
    }
    // abre arquivo de temperaturas médias para GRAVAÇÃO
    FILE *f2 = fopen( "temp_medias.txt","wt"); // f2 irá referenciar arq.gravação
    if( f2 == NULL){
        printf("Erro na abertura do arquivo!\n");
        return -1;
    }
    x = fscanf( f1, "%f", &temp[cont]); // lê UM valor a cada vez
    while( x == 1 ){
```

```

printf( "%o. valor: %.3f\n", cont+1,temp[cont]);
media+=temp[cont];
cont++;
if(cont>2) {
    media=media/3.0;
    fprintf( f2, "%.3f\n", media); // grava media no segundo arquivo
    quant++;
    cont=0;
    media=0;
}
x = fscanf( f1, "%f", &temp[cont]); // lê UM valor a cada vez
}
fclose( f1 );
fclose( f2 );
printf("Foram gravadas %d medias.\n", quant);

// *** 3a. etapa: lendo dados no arquivo de medias e mostra na tela ***
f2 = fopen( "temp_medias.txt","rt" );
if( f2 == NULL){
    printf("Erro na abertura do arquivo!\n");
    return -1;
}
x = fscanf( f2, "%f", &media );
while( x == 1 ){
    cont++;
    printf( "%do. valor: %.3f\n", cont, media );
    x = fscanf( f2, "%f", &media );
}

return 0;
}

```

EXEMPLO 4:

```
// EXEMPLOS DE USO DE ARQUIVOS
// Leitura e impressão dos dados de VARIAS pessoas
// GRAVAÇÃO E LEITURA DOS DADOS EM ARQUIVO
// Exemplo 3: uma função para ler uma pessoa de cada vez
//          uma função especial para ler só a data de nascimento
//          uma função especial para mostrar só a data de nascimento
//          uma função para mostrar TODAS as pessoa de uma só vez
//          usando o ponteiro com colchetes Ex.: p[i].nome
//          uma função que carrega dados do arquivo para o vetor de pessoas

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

// Definição das estruturas (global)
struct data{ // estrutura de data
    int dia;
    int mes;
    int ano;
};
struct pessoa{ // estrutura de pessoa
    char nome[30];
    char telefone[20];
    struct data nascimento;
};

// Protótipos das funções
struct pessoa* carrega_pessoas(struct pessoa *p, int *q);
struct pessoa* le_pessoa( struct pessoa *p, int q);
void mostra_pessoas(struct pessoa *p, int q);
void grava_pessoa(struct pessoa *p, int q);
void le_data(struct data *p);
void mostra_data(struct data x);

///////////////////////////////// FUNÇÃO PRINCIPAL ///////////////////////////////////
int main(){
    // Declaração de um ponteiro para a estrutura de pessoa previamente criada
    // Isto irá permitir que existam várias instâncias de pessoas (várias pessoas)
    struct pessoa *p; // p será um vetor de pessoas
    int resposta;
    int q=0; // guarda o número de pessoas armazenadas
    p=NULL; // ponteiro de pessoa aponta para NULL (permite realocar depois)

    printf("CADASTRO DE PESSOAS COM USO DE ARQUIVOS\n");
    printf("=====\n\n");
    p=carrega_pessoas(p,&q); // lê dados gravados no arq. e armazena no ponteiro
    printf("\n---> Listagem de pessoas cadastradas APOS CARREGAR\n");
    mostra_pessoas(p,q);
    printf("\n---> Cadastro de novas pessoas\n");
    do {
```

```

    printf("Deseja incluir uma nova pessoa no arquivo? (1=SIM e 2=NAO):");
    scanf("%d",&resposta);
    if(resposta==1){
        q++; // incrementa a quantidade de pessoas armazenadas
        p=le_pessoa(p,q); // lê os dados de UMA pessoa, alocando memória
        grava_pessoa(p,q); // grava apenas a última armazenada no vetor
    }
} while(resposta==1);
printf("\n---> Listagem de pessoas cadastradas AO FINAL\n");
mostra_pessoas(p,q);

return 0;
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

// DEMAIS FUNÇÕES
struct pessoa* carrega_pessoas( struct pessoa *p, int *q ){
    int i=0,x; // i guarda a quantidade de registros que terá no arquivo
    printf("\nCarregando arquivo com as pessoas cadastradas:");
    FILE *f1 = fopen( "pessoas.txt","a+t");
    if( f1 == NULL){
        printf("Erro na abertura do arquivo!\n");
        exit(0);
    }
    // aloca espaço de memória para armazenar UMA pessoa
    p=(struct pessoa*)realloc(p,sizeof(struct pessoa));
    if(p==NULL) {
        printf("Problema na realocacao!");
        exit(0);
    }
    // Tenta ler o primeiro registro do arquivo
    x = fscanf( f1, "%s %s %d %d %d", p[i].nome,p[i].telefone,
                &p[i].nascimento.dia, &p[i].nascimento.mes,
                &p[i].nascimento.ano);
    while(x!=EOF){ // se não chegou ao "fim do arquivo" (se há dados)
        i++;
        p=(struct pessoa*)realloc(p,sizeof(struct pessoa)*(i+1));
        if(p==NULL) {
            printf("Problema na realocacao!");
            exit(0);
        }
        // Tenta ler o próximo registro do arquivo
        x = fscanf( f1, "%s %s %d %d %d", p[i].nome,p[i].telefone,
                    &p[i].nascimento.dia, &p[i].nascimento.mes,
                    &p[i].nascimento.ano);
    }
    *q=i;
    printf("\nCarregou %d pessoas.\n\n",*q);
    fclose(f1);
    return p;
}

```

```

struct pessoa* le_pessoa( struct pessoa *p, int q ){
    int i;
    p=(struct pessoa*)realloc(p,sizeof(struct pessoa)*q);
    if(p==NULL) {
        printf("Problema na realocacao!");
        exit(0);
    }
    else
    {
        fflush(stdin);
        printf("Digite o nome: ");
        gets(p[q-1].nome);
        printf("Digite o numero de telefone: ");
        gets(p[q-1].telefone);
        printf("Digite a data de nascimento:\n");
        le_data(&p[q-1].nascimento);
    }
    return p;
}

void mostra_pessoas( struct pessoa *p, int q ){
    int i;
    printf("\nDados das pessoas cadastradas:\n");
    for(i=0;i<q;i++){
        printf("Nome: %s - ", p[i].nome );
        printf("Telefone: %s - ", p[i].telefone );
        mostra_data(p[i].nascimento);
    }
}

void le_data( struct data *p ){
    printf("Digite o dia: ");
    scanf("%d", &p->dia );
    printf("Digite o mes: ");
    scanf("%d", &p->mes );
    printf("Digite o ano: ");
    scanf("%d", &p->ano );
}

void mostra_data( struct data x ){
    printf("Nascimento: %2d/%2d/%4d\n", x.dia, x.mes, x.ano);
}

void grava_pessoa( struct pessoa *p, int q){
    FILE *f1 = fopen( "pessoas.txt","a+t");
    if( f1 == NULL){
        printf("Erro na abertura do arquivo!\n");
        exit(0);
    }
    fprintf( f1, "%s\t%s\t%d\t%d\t%d\n", p[q-1].nome,p[q-1].telefone,

```

```
        p[q-1].nascimento.dia,  
        p[q-1].nascimento.mes,  
        p[q-1].nascimento.ano);  
    fclose(f1);  
}
```