

Os exemplos aqui apresentados referem-se aos “Slides Tipos Estruturados (parte 3)” postados no Moodle.

EXEMPLO 1:

```
// EXEMPLOS DE USO DE ESTRUTURAS
// Leitura e impressão dos dados de UMA pessoa
// NESTE EXEMPLO: uma função para ler e outra para mostrar

#include <stdio.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
void le_pessoa( struct pessoa *p );
void mostra_pessoa( struct pessoa x );

// Função principal
int main(){
    // Declaração de uma pessoa (fulano) com a estrutura previamente criada
    struct pessoa fulano;

    le_pessoa( &fulano );
    mostra_pessoa( fulano );
    return 0;
}

// Definições das funções declaradas
void le_pessoa( struct pessoa *p ){
    printf("Digite o nome: ");
    gets(p->nome);
    printf("Digite o numero de telefone: ");
    gets(p->telefone);
    printf("Digite a data de nascimento:\n");
    printf("Digite o dia: ");
    scanf("%d", &p->nascimento.dia );
    printf("Digite o mes: ");
    scanf("%d", &p->nascimento.mes );
    printf("Digite o ano: ");
    scanf("%d", &p->nascimento.ano );
}
```

```
}  
  
void mostra_pessoa( struct pessoa x ){  
    printf("Nome: %s\n", x.nome );  
    printf("Telefone: %s\n", x.telefone );  
    printf("Data de nascimento: %2d/%2d/%4d\n",  
          x.nascimento.dia,  
          x.nascimento.mes,  
          x.nascimento.ano );  
}
```

EXEMPLO 2:

```
// EXEMPLOS DE USO DE ESTRUTURAS
// Leitura e impressão dos dados de UMA pessoa
// NESTE EXEMPLO: uma função para ler e outra para mostrar
//                uma função especial para ler só a data de nascimento
//                uma função especial para mostrar só a data de nascimento

#include <stdio.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
void le_pessoa( struct pessoa *p );
void mostra_pessoa( struct pessoa x );
void le_data( struct data *p );
void mostra_data( struct data x );

// Função principal
int main(){
    // Declaração de uma pessoa (fulano) com a estrutura previamente criada
    struct pessoa fulano;

    le_pessoa( &fulano );
    mostra_pessoa( fulano );
    return 0;
}

// Definições das funções declaradas
void le_pessoa( struct pessoa *p ){
    printf("Digite o nome: ");
    gets(p->nome);
    printf("Digite o numero de telefone: ");
    gets(p->telefone);
    printf("Digite a data de nascimento:\n");
    printf("Digite o dia: ");
    le_data(&p->nascimento);
}

void mostra_pessoa( struct pessoa x ){
    printf("Nome: %s\n", x.nome );
```

```
        printf("Telefone: %s\n", x.telefone );
        mostra_data(x.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("%2d/%2d/%4d\n", x.dia, x.mes, x.ano);
}
```

EXEMPLO 3 - versão 1:

```
// EXEMPLOS DE USO DE ESTRUTURAS
// Leitura e impressão dos dados de VARIAS pessoas
// 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
//          uma função para ordenar os dados das pessoas pelo nome
//          uma função para mostrar pessoas com uma letra inicial no nome
//          uma função para mostrar pessoas a partir de um ano de nascimento
// VERSÃO 1: usando o ponteiro com o operador seta Ex.: (p+i)->nome

#include <stdio.h>
#include <stdlib.h>
#include <string.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* le_pessoa( struct pessoa *p, int q );
void mostra_pessoas( struct pessoa *p, int q );
void ordena_pessoas( struct pessoa *p, int q );
void mostra_pela_inicial( struct pessoa *p, int q );
void mostra_pelo_ano( 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 quant=0;
    p=NULL; // ponteiro de pessoa aponta para NULL (permite realocar depois)

    do {
        quant++; // incrementa a quantidade de pessoas armazenadas
        p=le_pessoa(p,quant);
        printf("Deseja incluir outra pessoa? (1=SIM e 2=NAO):");
        scanf("%d",&resposta);
    }
```

```

    } while(resposta==1);
    mostra_pessoas(p,quant);
    ordena_pessoas(p,quant);
    mostra_pessoas(p,quant);
    mostra_pela_inicial(p,quant);
    mostra_pelo_ano(p,quant);

    return 0;
}

// Definições das funções declaradas
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("\n\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 ordena_pessoas( struct pessoa *p, int q ){
    int i,j;
    struct pessoa aux; // permite armazenar UMA pessoa para fazer a troca
    printf("\n\nOrdenando dados das pessoas pelo nome.\n");
    for(i=0;i<q-1;i++){
        for(j=i+1;j<q;j++){
            if(strcmp((p+i)->nome,(p+j)->nome)>0) { // se estiver desordenado
                aux=*(p+i);
                *(p+i)=*(p+j);
                *(p+j)=aux;
            }
        }
    }
}

void mostra_pela_inicial( struct pessoa *p, int q ){
    int i;
    char letra;
    printf("\n\nConsulta pessoas cadastradas pela inicial do nome:\n");
    printf("Digite a inicial do nome a consultar:");
    scanf(" %c",&letra);
    for(i=0;i<q;i++){
        if((p+i)->nome[0]==letra){
            printf("Nome: %s - ", (p+i)->nome );
            printf("Telefone: %s - ", (p+i)->telefone );
            mostra_data((p+i)->nascimento);
        }
    }
}

void mostra_pelo_ano( struct pessoa *p, int q ){
    int i,ano_inicial;
    printf("\n\nConsulta pessoas cadastradas pelo ano de nascimento:\n");
    printf("Digite o ano a partir do qual deseja consultar:");
    scanf("%d",&ano_inicial);
    for(i=0;i<q;i++){
        if((p+i)->nascimento.ano>=ano_inicial){
            printf("Nome: %s - ", (p+i)->nome );
            printf("Telefone: %s - ", (p+i)->telefone );
            mostra_data((p+i)->nascimento);
        }
    }
}

```

EXEMPLO 3 - versão 2:

```
// EXEMPLOS DE USO DE ESTRUTURAS
// Leitura e impressão dos dados de VARIAS pessoas
// 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
//          uma função para ordenar os dados das pessoas pelo nome
//          uma função para mostrar pessoas com uma letra inicial no nome
//          uma função para mostrar pessoas a partir de um ano de nascimento
// VERSÃO 2: usando o ponteiro com colchetes Ex.: p[i].nome

#include <stdio.h>
#include <stdlib.h>
#include <string.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* le_pessoa( struct pessoa *p, int q );
void mostra_pessoas( struct pessoa *p, int q );
void ordena_pessoas( struct pessoa *p, int q );
void mostra_pela_inicial( struct pessoa *p, int q );
void mostra_pelo_ano( 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 quant=0;
    p=NULL; // ponteiro de pessoa aponta para NULL (permite realocar depois)

    do {
        quant++; // incrementa a quantidade de pessoas armazenadas
        p=le_pessoa(p,quant);
        printf("Deseja incluir outra pessoa? (1=SIM e 2=NAO):");
        scanf("%d",&resposta);
    }
```



```

    } while(resposta==1);
    mostra_pessoas(p,quant);
    ordena_pessoas(p,quant);
    mostra_pessoas(p,quant);
    mostra_pela_inicial(p,quant);
    mostra_pelo_ano(p,quant);

    return 0;
}

// Definições das funções declaradas
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("\n\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 ordena_pessoas( struct pessoa *p, int q ){
    int i,j;
    struct pessoa aux; // permite armazenar UMA pessoa para fazer a troca
    printf("\n\nOrdenando dados das pessoas pelo nome.\n");
    for(i=0;i<q-1;i++){
        for(j=i+1;j<q;j++){
            if(strcmp(p[i].nome,p[j].nome)>0) { // se estiver fora de ordem
                aux=p[i];
                p[i]=p[j];
                p[j]=aux;
            }
        }
    }
}

void mostra_pela_inicial( struct pessoa *p, int q ){
    int i;
    char letra;
    printf("\n\nConsulta pessoas cadastradas pela inicial do nome:\n");
    printf("Digite a inicial do nome a consultar:");
    scanf(" %c",&letra);
    for(i=0;i<q;i++){
        if(p[i].nome[0]==letra){
            printf("Nome: %s - ", p[i].nome );
            printf("Telefone: %s - ", p[i].telefone );
            mostra_data(p[i].nascimento);
        }
    }
}

void mostra_pelo_ano( struct pessoa *p, int q ){
    int i,ano_inicial;
    printf("\n\nConsulta pessoas cadastradas pelo ano de nascimento:\n");
    printf("Digite o ano a partir do qual deseja consultar:");
    scanf("%d",&ano_inicial);
    for(i=0;i<q;i++){
        if(p[i].nascimento.ano>=ano_inicial){
            printf("Nome: %s - ", p[i].nome );
            printf("Telefone: %s - ", p[i].telefone );
            mostra_data(p[i].nascimento);
        }
    }
}

```