

Danrley Pereira
23551470

Resultados da execução:

```
Danrley-PC 福 ~/Documentos/udf/third/tecnica de programacao/codes/struct1
1206 0 : ./61.out
titulo: titulo qualquer,
autor: Joseleide da Silva,
editora: Editora Atila,
ano de publicação: 1997

Danrley-PC 福 ~/Documentos/udf/third/tecnica de programacao/codes/struct1
1207 0 : ./65.out

Danrley-PC 福 ~/Documentos/udf/third/tecnica de programacao/codes/struct1
1208 0 : ./67.out
preenchendo dados
not found => fone-1234567
encontrado = fone-123
agenda[2].fone = fone-12

Danrley-PC 福 ~/Documentos/udf/third/tecnica de programacao/codes/struct1
1209 0 : █
```

Exercício 6.1

```
/*  Name:

    Copyright:  CC0 license
    Author:  Danrley Pereira
    Date:  03/05/21 11:06
    Description:  Struct anônimo;

    Enunciado.:  Codifique um programa para criar uma variável de tipo anô-
    nimo, capaz de armazenar o título, o autor, a editora e o ano de
publicação
    de um livro; atribuir valores aos seus campos e exibi-la no vídeo.
*/

#include <stdio.h>
#include <string.h>
// struct anonima livro
typedef struct {
    char titulo[40];
```

```

    char autor[40];
    char editora[40];
    int anoPublicacao;
} Livro;

int main(){
    //declara livro
    Livro livro;
    //defini valores
    strcpy(livro.titulo, "titulo qualquer");
    strcpy(livro.autor, "Joseleide da Silva");
    strcpy(livro.editora, "Editora ATila");
    livro.anoPublicacao = 1997;
    //printa valores
    printf("titulo: %s, \nautor: %s, \neditora: %s, \nano de
publicação: %d\n", livro.titulo, livro.autor, livro.editora,
livro.anoPublicacao);

    return 0;
}

```

Exercício 6.5

```

/*  Name:
    Copyright:  CC0 license
    Author:  Danrley Pereira
    Date:  03/05/21 11:06
    Description:  Struct anônimo;

    Enunciado.: Defina um tipo de estrutura para armazenar os dados de um
    vôo como, por exemplo, os nomes das cidades de origem e destino, datas
    e
    horários de partida e chegada. Crie uma variável desse tipo e atribua
    valo-
    res aos seus memros usando a notação de ponto e, depois, inicialização.
*/

#include <stdio.h>
#include <string.h>
// estrutura data
typedef struct {

```

```
    int dia;
    int mes;
    int ano;
} Data;

//estrutura horario
typedef struct {
    int hora;
    int minuto;
    int segundo;
} Horario;

//estrutura Voo
typedef struct {
    char origem[40];
    char destino[40];
    Data dataPartida;
    Data dataChegada;
    Horario horarioPartida;
    Horario horarioChegada;
} Voo;

int main() {
    // declara voo
    Voo voo;

    //atribuindo valores usando notação de ponto
    strcpy(voo.origem, "Piracicaba");
    strcpy(voo.destino, "Cafundó do Judas");

    Data dataPartida = {13, 4, 1997};
    Data dataChegada = {13, 4, 1997};
    voo.dataPartida = dataPartida;
    voo.dataChegada = dataChegada;

    Horario horarioPartida = {16, 25, 12};
    Horario horarioChegada = {22, 36, 15};
    voo.horarioPartida = horarioPartida;
    voo.horarioChegada = horarioChegada;
    // fim da inicialização de ponto
```

```

        //atribuindo valores usando inicialização
        Voo voo1 = {"Piracicaba", "Cafundó do Judas", {13, 4, 1997}, {13,
4, 1997}, {16, 25, 12}, {22, 36, 15}};

        // printf("origem: %s \n destino: %s ", voo.origem, voo.destino);
        // printf("data de partida: %d/%d/%d\n", voo.dataPartida.dia,
voo.dataPartida.mes, voo.dataPartida.ano);

        return 0;
    }

```

Exercício 6.7

```

/*  Name:
    Copyright:  CC0 license
    Author:  Danrley Pereira
    Date:  03/05/21 11:06
    Description:  Struct anônimo;

    Enunciado.: Usando o tipo PESSOA , definido no exemplo 6.6 , crie uma
fun-
ção para preencher uma agenda com os dados de 5 pessoas. Crie também
uma função para procurar na agenda o telefone de uma determinada
pessoa e codifique um programa para testar essas funções.
*/

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

// criando função que converte int para string
char* itoa(int val, int base){
    static char buf[32] = {0};
    int i = 30;
    for(; val && i ; --i, val /= base)
        buf[i] = "0123456789abcdef"[val % base];
    return &buf[i+1];
}

//estrutura DATA

```

```

typedef struct {
    int dia;
    int mes;
    int ano;
} DATA;
//estrutura PESSOA
typedef struct {
    char nome[31];
    char fone[21];
    DATA nasc;
} PESSOA;

// variável global agenda
PESSOA agenda[5];

// rotina (void) para procurar telefone
void procurarTelefone(char* telefone) {
    //percorre a agenda
    for(int i = 0; i<5; i++){
        //compara as strings
        if (strcmp(agenda[i].fone, telefone) == 0)
        {
            //se for igual é pq encontrou o telefone passado como
parâmetro

            printf("encontrado = ");
            //criar uma variável provisória para ler o telefone
            char *string = telefone;
            //percorre a variável imprimindo cada caractere até o fim
da string

            while(*string != '\0'){
                printf("%c", *string++);
            }
            printf("\n");
            //finaliza a rotina
            return;
        }
    }
    printf("not found => ");
    char *string = telefone;
    while(*string != '\0'){

```

```

        printf("%c", *string++);
    }
    printf("\n");
}

// preenche os dados da agenda
void preencherDados() {
    //cria ponteiro onde vai ser salvo o número que será convertido em
string
    char* number;
    //prefixo do nome e do telefone
    char nome[10] = "nome-";
    char fone[10] = "fone-";
    // percorre a agenda
    for(int i = 0; i<5; i++){
        //passa i e a base do número para retornar a string equivalente
        number = itoa(i, 10);
        //concatena o número. eg.: nome-1; nome-12, nome-123 etc
        strcat(nome, number);
        //mesma coisa do nome só que usando o prefixo fone
        strcat(fone, number);
        //copia a concatenação na agenda
        strcpy(agenda[i].nome, nome);
        strcpy(agenda[i].fone, fone);
        //declara data
        DATA data = {13,04,1941};
        //atribui data na agenda para a pessoa atual
        agenda[i].nasc = data;
    }
}

int main(){
    printf("preenchendo dados\n");
    preencherDados();
    procurarTelefone("fone-1234567");
    procurarTelefone("fone-123");
    printf("agenda[2].fone = %s \n", agenda[2].fone);
    return 0;
}

```

