

# PROGRAMAÇÃO DE COMPUTADORES

PROFESSOR: FÁBIO GARCEZ BETTIO

ESTUDANTE: CLÍSTENES GRIZAFIS BENTO

## APS 1 LISTA DE EXERCÍCIOS STRUCT V1.1

1. Crie um programa em C que preencha uma struct usando scanf depois do preenchimento imprima-a.

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

/* .1) Crie um programa em C que preencha uma struct usando scanf depois do
preenchimento imprima a matriz. */

struct BANDA
{
    char nome_banda [30];
    int musicas_que_gosta;
};

struct BANDA cadastro1;

int main()
{
    setlocale(LC_ALL,"portuguese");

    printf("BEM VINDO AO CADASTRO DE BANDAS\n");
    printf("NESSE PROGRAMA VOCÊ PODE CADASTRAR SUA BANDA PREFERIDA
E A QUANDIDADE DE MÚSICAS QUE GOSTA DESSA BANDA\n\n");

    printf("Por gentileza, digite o nome da banda: ");
    scanf("%s",&cadastro1.nome_banda);
    printf("\n Por gentileza digite a quantidade de músicas que gosta: ");
    scanf("%d",&cadastro1.musicas_que_gosta);

    system("cls");

    printf("Você      gosta      de      %d      músicas      da      banda
%s.",cadastro1.musicas_que_gosta,cadastro1.nome_banda);
    printf("\n\n\n");

    system("pause");
    return 0;
}
```

2. Crie um programa em C que preencha uma struct CADASTRO com duas substructs ENDERECO (comercial e residencial) usando scanf, depois do preenchimento imprima a struct e todo seu conteúdo.

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

/* .2) Crie um programa em C que preencha uma struct CADASTRO com duas substructs
ENDERECO (comercial e residencial) usando scanf,
depois do preenchimento imprima a matriz e todo seu conteúdo. */

struct ENDERECO //substruct
{
    char rua [50]; //variável que vai receber o nome da rua
    int numero=0; //variável que vai receber o número da rua
};

struct CADASTRO //a struct principal
{
    char nome [30];
    struct ENDERECO res; // res = residencial
    struct ENDERECO com; // com = comercial
};

struct CADASTRO cadastro1; //declaração da struct cadastro com a variável "cadastro1"

int main()
{
    setlocale(LC_ALL, "portuguese"); //coloca idioma

    printf("BEM VINDO AO CADASTRO DE ENDEREÇOS\n"); // mensagem de boas
vindas
    printf("NESSE PROGRAMA VOCÊ PODE CADASTRAR SEU ENDEREÇO
RESIDENCIAL E COMERCIAL\n\n");

    printf("Por gentileza, digite o seu nome: ");
    scanf("%s",&cadastro1.nome); //leitura de do nome
    printf("\n Por gentileza digite o seu endereço residencial: ");
    scanf("%s",&cadastro1.res.rua); //leitura da rua residencial
    printf("\n Agora o número: ");
    scanf("%d",&cadastro1.res.numero); //leitura do número residencial
    printf("\n Por gentileza digite o seu endereço comercial: ");
    scanf("%s",&cadastro1.com.rua); //leitura do rua comercial
    printf("\n Agora o número: ");
    scanf("%d",&cadastro1.com.numero); //leitura do número comercial
```

```
system("cls"); // limpa a tela

/*impressão das informações */

printf("NOME: %s",cadastro1.nome);
printf("\nENDEREÇO RESIDENCIAL: %s,
%d",cadastro1.res.rua,cadastro1.res.numero);
printf("\nENDEREÇO COMERCIAL: %s,
%d",cadastro1.com.rua,cadastro1.com.numero);

printf("\n\n\n");

system("pause"); //pausa o programa
return 0;
}
```

3. Crie um vetor com 5 structS CADASTRO com duas substructs ENDERECO (comercial e residencial), preencha cada um com scanf e depois imprima. Importante que existam dois loops distintos, um para o preenchimento e outro para a impressão.

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

/* 3. Crie um vetor com 5 structS CADASTRO com duas substructs ENDERECO (comercial
e residencial), preencha cada um com scanf e depois imprima.
Importante que existam dois loops distintos, um para o preenchimento e outro para a
impressão.*/

struct ENDERECO
{
    char rua [50];
    int numero=0;
};

struct CADASTRO
{
    char nome [30];
    struct ENDERECO res; // res = residencial
    struct ENDERECO com; // com = comercial
};

struct CADASTRO cadastro[5];

int main()
{
    setlocale(LC_ALL, "portuguese");

    int i=0,j=0;

    printf("BEM VINDO AO CADASTRO DE ENDEREÇOS\n");
    printf("NESSE PROGRAMA VOCÊ PODE CADASTRAR SEU ENDEREÇO
RESIDENCIAL E COMERCIAL\n\n");

    for(i=0;i<5;i++)
    {
        printf("\n\nCADASTRO [%d]\nPor gentileza, digite o seu nome: ",i+1);
        scanf("%s",&cadastro[i].nome);
        printf("\n Por gentileza digite o seu endereço residencial: ");
        scanf("%s",&cadastro[i].res.rua);
        printf("\n Agora o número: ");
        scanf("%d",&cadastro[i].res.numero);
```

```

        printf("\n Por gentileza digite o seu endereço comercial: ");
        scanf("%s",&cadastro[i].com.rua);
        printf("\n Agora o número: ");
        scanf("%d",&cadastro[i].com.numero);

    }

    system("cls");

    for(j=0;j<5;j++)
    {

        printf("\nNOME: %s",cadastro[j].nome);
        printf("\nENDEREÇO RESIDENCIAL: %s, %d",cadastro[j].res.rua,cadastro[j].res.numero);
        printf("\nENDEREÇO COMERCIAL: %s, %d",cadastro[j].com.rua,cadastro[j].com.numero);
    }

    printf("\n\n\n");

    system("pause");
    return 0;
}

```

4. Crie duas structs CORRESPONDENCIA (com CEP, rua, numero, bairro, cidade, estado, uma será chamada de A e outra de B. Preencha os dados da A, copie os dados para a struct B e imprima a struct B.

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

/* 4. Crie duas structs CORRESPONDENCIA (com CEP, rua, numero, bairro, cidade, estado,
uma será chamada de A e outra de B. Preencha os dados da A,
copie os dados para a struct B e imprima a struct B. */

struct CORRESPONDENCIA
{
    char rua [50];
    int numero=0;
    char bairro[50];
    char cidade[20];
    char estado[15];
    int cep=0;
};

struct CORRESPONDENCIA A;
struct CORRESPONDENCIA B;

int main()
{
    setlocale(LC_ALL, "portuguese"); //coloca idioma

    printf("BEM VINDO AO CADASTRO DE ENDEREÇOS\n");
    printf("NESSE PROGRAMA VOCÊ PODE CADASTRAR SEU ENDEREÇO\n\n");

    printf("Por gentileza, digite a rua: ");
    scanf("%s",&A.rua);
    printf("\n Agora o número: ");
    scanf("%d",&A.numero);
    printf("\n Digite o bairro: ");
    scanf("%s",&A.bairro);
    printf("\n Digite o nome da cidade: ");
    scanf("%s",&A.cidade);
    printf("\n Digite o nome do estado: ");
    scanf("%s",&A.estado);
    printf("\n Agora digite o CEP: ");
    scanf("%d",&A.cep);

    strcpy(B.rua,A.rua);
```

```
B.numero=A.numero;  
strcpy(B.bairro,A.bairro);  
strcpy(B.cidade,A.cidade);  
strcpy(B.estado,A.estado);  
B.cep=A.cep;
```

```
system("cls"); // limpa a tela
```

```
/*impressão das informações */
```

```
printf("SEGUE ABAIXO ENDEREÇO CASASTRADO");  
printf("\nrUA: %s, %d - %s - %s / %s",B.rua,B.numero,B.bairro,B.cidade,B.estado);  
printf("\nCEP: %d",B.cep);
```

```
printf("\n\n\n");
```

```
system("pause"); //pausa o programa  
return 0;
```

```
}
```

5. Crie um vetor com 5 structs CADASTRO (com nome e idade), cadastre os dados em um loop, em um segundo loop busque a struct que tem a maior idade e imprima.

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

/* 5. Crie um vetor com 5 structs CADASTRO (com nome e idade), cadastre os dados em
um loop,
em um segundo loop busque a struct que tem a maior idade e imprima. */

struct CADASTRO
{
    char nome[50];
    int idade=0;
};

struct CADASTRO cad[5];

int main()
{
    setlocale(LC_ALL, "portuguese"); //coloca idioma

    int maior=0;
    int i=0, j=0, posicao=0;

    printf("BEM VINDO AO CADASTRO DE NOME E IDADE\n");
    printf("NESSE PROGRAMA VOCÊ PODE CADASTRAR NOMES E IDADES\n\n");
    printf("O PROGRAMA IRÁ ANALISAR QUEM TEM A MAIOR IDADE E REVELAR
O NOME\n\n");

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

        printf("\n\nCADASTRO [%d]\nPor gentileza, digite o nome: ", i+1);
        scanf("%s", &cad[i].nome);
        printf("\n Por gentileza digite a idade: ");
        scanf("%d", &cad[i].idade);

        if(cad[i].idade > maior)
        {

            maior = cad[i].idade;

        }

    }
}
```



```
        /*impressão das informações */  
printf("O(S) NOME(S) QUE POSSUE(M) A(S) MAIOR(ES) IDADE(S) É(SÃO): \n");  
for(j=0;j<5;j++)  
    {  
        if(cad[j].idade==maior)  
        {  
            printf("\n%s com %d anos.",cad[j].nome,cad[j].idade);  
        }  
    }  
  
printf("\n\n\n");  
system("pause"); //pausa o programa  
return 0;  
}
```

6. Faça um programa em C que preencha um vetor de struct com a seguinte estrutura:

Nome

Cargo

Telefone

e-mail

Salário

O programa devera:

- a) Cadastrar 5 funcionários, um após o outro.
- b) Imprimir o total dos salários de todos os funcionários
- c) Informar qual o funcionário que possui o maior salário
- d) Informar se algum funcionário está com o número de telefone em branco

```
#include <stdio.h> //biblioteca básica
#include <stdlib.h> //biblioteca básica
#include <string.h> //biblioteca para operar strings
#include <locale.h> //biblioteca de idiomas para acentuação

/* 6. Faça um programa em C que preencha um vetor de struct com a seguinte estrutura:
Nome
Cargo
Telefone
e-mail
Salário

O programa devera:
a) Cadastrar 5 funcionários, um após o outro.
b) Imprimir o total dos salários de todos os funcionários
c) Informar qual o funcionário que possui o maior salário
d) Informar se algum funcionário está com o numero de telefone em branco
*/

struct FUNCIONARIO //criação da struct
{
    char nome [30]; //criação da variável nome
    char cargo [30]; //criação da variável cargo
    int telefone=0; //criação da variável telefone
    char email[30]; //criação da variável email
    float salario=0; //criação da variável salário
};

struct FUNCIONARIO fun[5]; //criação do vetor fun que puxa informações da struct
FUNCIONÁRIOS, com cinco espaços de alocação.
```

```

int main() //Começo do programa
{

    setlocale(LC_ALL,"portuguese"); //definição do idioma para acentuação

    float soma_salario=0; //criação de uma variavel formada pelos números reais com
    valor inicial zero, que servirá para a soma de todos os salários cadastrados
    int i=0,j=0,k=0; //criação de variáveis inteiras que servirão como contadores nos
    loopings
    float maior=0; // criação de variável formada pelos números reais para computar
    qual o maior salário entre os funcionários, com valor inicial em zero

    printf("BEM VINDO AO CADASTRO DE FUNCIONÁRIOS"); //mensagem de boas
    vindas
    printf("NESSE PROGRAGA VOCÊ PODERÁ CADASTRAR CINCO FUNCIONÁRIO
    E RECEBER INFORMAÇÕES RELEVANTES SOBRE ELES"); //breve instrução de como
    funciona o programa

    for(i=0;i<5;i++) //declaração do laço de repetição (loop) chamado for, que inicia em
    0 e vai repetindo toda a instrução do bloco até i chegar em 4
    {
        printf("\n\nCADASTRO [%d]\n\n",i+1); //mensagem que informa qual a
        posição do funcionário cadastrado, obs: o i+1 representa é para que a contagem apareça a
        partir do 1, pois o i inicia em zero
        printf("Por gentileza digite o nome do funcionário: "); //daqui até o final do
        bloco serão coletas de informações
        scanf("%s",&fun[i].nome);
        printf("\nDigite o cargo do funcionário: ");
        scanf("%s",&fun[i].cargo);
        printf("\nDigite o número do telefone: ");
        scanf("%d",&fun[i].telefone);
        printf("\nDigite o e-mail: ");
        scanf("%s",&fun[i].email);
        printf("\nDigite o salário em R$: ");
        scanf("%f",&fun[i].salario);

        soma_salario=soma_salario+fun[i].salario; //aqui é feito o cálculo da soma de
        todos os salários

        if(fun[i].salario>maior) //aqui é feito a comparação entre os salário para
        verificar qual é o maior
        {
            maior=fun[i].salario; //aqui a variavel "maior" recebe o valor do salário
            caso o salpario seja superior
        }

    }

    system("cls"); //essa função limpa a tela de impressão

```

`printf("A soma de todos os salários é igual a R$%.2f",soma_salario); //aqui está sendo impresso a soma dos salários, note que tem a expressão "%.2f" isso significa que eu quero que imprima apenas duas casas depois da vírgula`

`//OBS: note que a soma está sendo impressa fora do laço de repetição, isso é devido ao fato de que se fosse dentro do laço, a variável seria impressa para cada valor de i`

`printf("\n\nO(s) funcionário(s) que possui(em) o maior(es) salário(s) é(são):\n");`

`for(j=0;j<5;j++) //aqui é outro laço, agora utilizando o contador j`  
`{`  
`if(fun[j].salario==maior) //essa condicional verifica quais funcionários que possuem o maior salário`  
`{`  
`printf("%s que recebe R$ %.2f\n",fun[j].nome,fun[j].salario); //aqui imprime o nome e o salário do funcionário com maior salário`  
`}`

`}`

`for(k=0;k<5;k++) //aqui é outro laço, agora utilizando o contador k`  
`{`

`if(fun[k].telefone==0) //essa condicional identifica qual dos números telefônicos estão com o valor inicial da variável. NOTA: não estou satisfeito ainda com como fiz essa parte.`  
`{`

`printf("\nO funcionário %s deixou o número de telefone em branco.",fun[k].nome); //aqui imprime o nome de quem não tem o telefone preenchido`  
`}`  
`}`

`printf("\n\n\n"); // Aqui é detalhe pessoal`

`system("pause"); //Aqui é um comando para o sistema parar antes de fechar o programa, no compilador não precisa, mas se for usar o executavel precisa.`

`return 0; //encerra o programa`  
`}`

7. Crie uma agenda telefônica como no exemplo 6, um sistema de menus deverá cadastrar cada usuário. (1. Novo 2. Remover, 3. Buscar, 4. Editar, 5. Sair)

```
#include <stdio.h> //biblioteca básica
#include <stdlib.h> //biblioteca básica
#include <string.h> //biblioteca para operar strings
#include <locale.h> //biblioteca de idiomas para acentuação

/* 7. Crie uma agenda telefônica como no exemplo 6, um sistema de menus deverá
cadastrar cada usuário. (1. Novo 2. Remover, 3. Buscar, 4. Editar, 5. Sair)
*/

struct AGENDA //criação da struct
{
    char nome [30]; //criação da variável nome
    int telefone=0; //criação da variável telefone
    char email[30]; //criação da variável email
    char endereco [50]; //criação da variável endereço
};

int main() //Começo do programa
{
    setlocale(LC_ALL,"portuguese"); //definição do idioma para acentuação

    int n=0; //número de pessoas cadastradas
    int capacidade = 100; //capacidade de contatos que a agenda comporta
    int opcao=0; //variavel para as opções do menu
    struct AGENDA pes[capacidade]; //criação do vetor pes que puxa informações da
struct AGENDA, com "n" espaços de alocação.
    int i=0, j=0; //Variáveis de contagem para os laços
    int novo_telefone=0; //variável para edição do telefone
    int edit; //Variável para opções de edição
    char busca[30]; //criação de uma variável para o usuário digitar o nome de busca

    a: //é o ponto para onde o goto a; manda

    if(capacidade==n){ //se a capacidade for igual ao número de contatos cadastrados,
dobrar o número da capacidade

        capacidade=2*n;
    }

    system("cls"); //Limpar a tela do programa (NESSA PARTE AQUI É PARA O CASO
DE VOLTAR AO MENU)

    printf("BEM VINDO À AGENDA"); //Mensagem de boas vindas
    printf("\n\nSELECIONE UMA OPÇÃO:"); // mensagem das opções
```

2

```
printf("\n\n[1] Adicionar um novo contato"); //mensagem com informação da opção 1
printf("\n[2] Remover um contato existente"); //mensagem com informação da opção

printf("\n[3] Buscar um contato"); //mensagem com informação da opção 3
printf("\n[4] Editar um contato existente"); //mensagem com informação da opção 4
printf("\n[5] Sair"); //mensagem com informação da opção 5
printf("\n\nOPÇÃO SELECIONADA: "); //mensagem que informa a opção
selecionada
scanf("%d",&opcao); //leitura da opção selecionada

switch(opcao) //switch case é um comando para executar determinada sintaxe de
acordo com a opção desejada (AQUI PODERIA SER UTILIZADO O if else)
{
    case 1: // caso a opção selecionada seja 1

        system("cls"); //limpeza de tela

        printf("Por gentileza, digite o nome do novo contato: ");
        scanf("%s",&pes[n].nome);
        printf("Digite o número de telefone: ");
        scanf("%d",&pes[n].telefone);
        printf("Digite o email: ");
        scanf("%s",&pes[n].email);
        printf("Digite o endereço: ");
        scanf("%s",&pes[n].endereco);

        n=n+1; //aqui aumenta a posição no vetor para o cadastro de novos
contatos

        printf("\n\n\n"); // Aqui é detalhe pessoal
        printf("CONTATO CADASTRADO COM SUCESSO \n%d",n,capacidade); //mensagem final e controle sutil das variáveis n e capacidade
        system("pause"); // pausa o programa antes de voltar para o menu

        goto a; //comando goto label, serve para fazer a leitura do programa
voltar à determinado ponto, nesse caso aponta a:

    case 2:

        system("cls"); //limpeza de tela

        char contato[30]; //criação de uma variável para o usuário digitar o
nome de busca

        printf("\nDigite o nome do contato que pretende apagar: ");
        scanf("%s",&contato);

        for(i=0;i<n+1;i++) //laço que passa por todos os contatos cadastrados
        {
            if(strcmp(contato,pes[i].nome)==0) //condicional que compara
as strings cadastradas com a string de busca, escolhendo seguir se tiver alguma igual
            {
```

```

        for(j=i+1;j<n+1;j++) //laço que passa por todos os
        contatos que estão alocados depois do contato encontrado
        {

            strcpy(pes[j-1].nome,pes[j].nome); //mudança
            de alocação no vetor, trazendo todos os vetores uma casa para baixo. (NOTA: aqui esta
            copiando as informação para a casa anterior)
            pes[j-1].telefone=pes[j].telefone;
            strcpy(pes[j-1].email,pes[j].email);
            strcpy(pes[j-1].endereco,pes[j].endereco);

            printf("\n\nCONTATO      APAGADO      COM
            SUCESSO");

        }
        n--; //remoção das informações alocadas na última
        posição cadastradas
        i=n; //aqui serve para encerrar o primeiro laço do case
2:
        }
    }

    printf("\n\n\n");// Aqui é detalhe pessoal

    system("pause");

    goto a;

case 3:

    system("cls"); //limpeza de tela

    printf("\nDigite o nome do contato que está procurando: ");
    scanf("%s",&busca);
    printf("\n\n");

    for(i=0;i<n+1;i++) //laço que passa por todos os contatos cadastrados
    {
        if(strcmp(busca,pes[i].nome)==0) //aqui está comparando as
        strings com o mesmo nome e imprimindo as informações
        {
            printf("NOME: %s",pes[i].nome);
            printf("\nTELEFONE: %d",pes[i].telefone);
            printf("\nE-MAIL: %s",pes[i].email);
            printf("\nENDEREÇO: %s",pes[i].endereco);
        }
    }

    printf("\nFIM      DA      CONSULTA      \nAPERTE      ENTER      PARA
    CONTINUAR");

    printf("\n\n\n");// Aqui é detalhe pessoal

    system("pause");

```

```

        goto a;

    case 4:

        system("cls"); //limpeza de tela

        printf("\nDigite o nome do contato que pretende editar: ");
        scanf("%s",&busca);
        printf("\n\n");

        for(i=0;i<n+1;i++) //laço que passa por todos os contatos cadastrados
        {
            if(strcmp(busca,pes[i].nome)==0) //condicional que compara
as strings cadastradas com a string de busca, escolhendo seguir se tiver alguma igual
            {

                printf("NOME: %s",pes[i].nome);
                printf("\nTELEFONE: %d",pes[i].telefone);
                printf("\nE-MAIL: %s",pes[i].email);
                printf("\nENDEREÇO: %s",pes[i].endereco);

                printf("\n\nESCOLHA UMA OPÇÃO"); //aqui depois de
imprimir as informações da agenda apresenta um menu de opções para alteração de
cadastro

                printf("\n\n[1] Alterar nome");
                printf("\n\n[2] Alterar telefone");
                printf("\n\n[3] Alterar e-mail");
                printf("\n\n[4] Alterar endereço");
                printf("\n\n[5] Voltar ao menu inicial");

                printf("\n\nESCOLHA: ");
                scanf("%d",&edit);

                switch(edit) //os casos aqui apresentados são para
substituição das informações de um contato já cadastrado, quando é string utiliza-se o
comando strcpy e quando é número basta substituir direto
                {

                    case 1:

                        system("cls"); //limpeza de tela

                        char novo_nome[30];

                        printf("Digite o novo nome: ");
                        scanf("%s",&novo_nome);

                        strcpy(pes[i].nome,novo_nome);

                        printf("\n\nCONTATO ALTERADO");

                        printf("\n\n\n"); // Aqui é detalhe pessoal

                        system("pause");

                        goto a;

```



**case 2:**

```
system("cls"); //limpeza de tela

printf("Digite o novo telefone: ");
scanf("%d",&novo_telefone);

pes[i].telefone=novo_telefone;

printf("\n\nCONTATO ALTERADO");

printf("\n\n"); // Aqui é detalhe pessoal

system("pause");

goto a;
```

**case 3:**

```
system("cls"); //limpeza de tela

char novo_email[30];

printf("Digite o novo e-mail: ");
scanf("%s",&novo_email);

strcpy(pes[i].email,novo_email);

printf("\n\nCONTATO ALTERADO");

printf("\n\n"); // Aqui é detalhe pessoal

system("pause");

goto a;
```

**case 4:**

```
system("cls"); //limpeza de tela

char novo_endereco[50];

printf("Digite o novo endereço: ");
scanf("%s",&novo_endereco);

strcpy(pes[i].endereco,novo_endereco);

printf("\n\nCONTATO ALTERADO");

printf("\n\n"); // Aqui é detalhe pessoal
```

```

        system("pause");

        goto a;

    case 5:

        goto a;

    default:

        printf("\nOPÇÃO INVÁLIDA");

        printf("\n\n");

        system("pause");

        goto a;

    }

}

printf("\n\n"); // Aqui é detalhe pessoal

system("pause");

goto a;

case 5:

    system("cls"); //limpeza de tela

    printf("\nFIM DO PROGRAMA");

    printf("\n\n"); // Aqui é detalhe pessoal

    system("pause");

    return 0;

default:

    printf("\nOPÇÃO INVÁLIDA");

    printf("\n\n");

    system("pause");

    goto a;

}

}

```

8. Desafio. Usando o programa do item 6 tente salvar os dados em arquivo

Imprima o vetor de struct em um arquivo txt

Pesquise sobre fopen(), fwrite(), fread()

```
#include <stdio.h> //biblioteca básica
#include <stdlib.h> //biblioteca básica
#include <string.h> //biblioteca para operar strings
#include <locale.h> //biblioteca de idiomas para acentuação

/* 8. Desafio. Usando o programa do item 6 tente salvar os dados em arquivo
Imprima o vetor de struct em um arquivo txt
Pesquise sobre fopen(), fwrite(), fread()

*/

struct FUNCIONARIO //criação da struct
{

    char nome [30]; //criação da variável nome
    char cargo [30]; //criação da variável cargo
    int telefone=0; //criação da variável telefone
    char email[30]; //criação da variável email
    float salario=0; //criação da variável salário

};

struct FUNCIONARIO fun[5]; //criação do vetor fun que puxa informações da struct
FUNCIONÁRIOS, com cinco espaços de alocação.

int main() //Começo do programa
{

    setlocale(LC_ALL,"portuguese"); //definição do idioma para acentuação

    FILE *save_cadastro; //ponteiro padrão utilizado para salvar em arquivo de txt ou
código com o nome save_contatos
    save_cadastro=fopen("cadastro_funcionarios.txt","r"); //comando para abrir o
arquivo com o nome do arquivo na primeira opção e o na segunda opção é o tipo de
operação, nesse caso é a leitura

    float soma_salario=0; //criação de uma variavel formada pelos números reais com
valor inicial zero, que servirá para a soma de todos os salários cadastrados
    int i=0,j=0,k=0,l=0; //criação de variáveis inteiras que servirão como contadores nos
loopings
    float maior=0; // criação de variável formada pelos números reais para computar
qual o maior salário entre os funcionários, com valor inicial em zero

    if(save_cadastro==NULL){ //condicional para verificar a existencia do arquivo a ser
aberto, se não existir o arquivo exibir uma mensagem

        printf("ARQUIVO NÃO ENCONTRADO");
```

```

        printf("\n\n\n");

        system("pause");

    }

    else //caso contrário ler o documento já existente
    {

        fread(fun,sizeof(struct FUNCIONARIO),5,save_cadastro); //esse aqui é um
comando de leitura do arquivo aberto.

        fclose(save_cadastro); //Esse daqui serve para fechar o arquivo que foi
aberto pelo fopen

    }

    printf("INFORMAÇÕES SOBRE FUNCIONÁRIOS JÁ CADASTRADOS"); //Nesse
bloco até o final do for é para verificar se tem informações no arquivo

    for(l=0;l<5;l++)
    {
        if(fun[l].nome!=NULL)
        {

            printf("\n\nNOME: %S",fun[l].nome);
            printf("\nCARGO: %S",fun[l].cargo);
            printf("\nTELEFONE: %d",fun[l].telefone);
            printf("\nE-MAIL: %S",fun[l].email);
            printf("\nSALÁRIO: %.2f",fun[l].salario);

        }

        else
        {

            printf("\n\nNENHUM FUNCIONÁRIO CADASTRADO");

        }

    }

    printf("\n\n\n");
    system("pause");
    system("cls");

    //Aqui começa o programa de cadastro de funcionários

    printf("BEM VINDO AO CADASTRO DE FUNCIONÁRIOS"); //mensagem de boas
vindas

```

```
printf("NESSE PROGRAGA VOCÊ PODERÁ CADASTRAR CINCO FUNCIONÁRIO  
E RECEBER INFORMAÇÕES RELEVANTES SOBRE ELES"); //breve instrução de como  
funciona o programa
```

```
for(i=0;i<5;i++) //declaração do laço de repetição (loop) chamado for, que inicia em  
0 e vai repetindo toda a instrução do bloco até i chegar em 4
```

```
{  
    printf("\n\nCADASTRO [%d]\n\n",i+1); //mensagem que informa qual a  
posição do funcionário cadastrado, obs: o i+1 representa é para que a contagem apareça a  
partir do 1, pois o i inicia em zero
```

```
    printf("Por gentileza digite o nome do funcionário: "); //daqui até o final do  
bloco serão coletas de informações
```

```
    scanf("%s",&fun[i].nome);
```

```
    printf("\nDigite o cargo do funcionário: ");
```

```
    scanf("%s",&fun[i].cargo);
```

```
    printf("\nDigite o número do telefone: ");
```

```
    scanf("%d",&fun[i].telefone);
```

```
    printf("\nDigite o e-mail: ");
```

```
    scanf("%s",&fun[i].email);
```

```
    printf("\nDigite o salário em R$: ");
```

```
    scanf("%f",&fun[i].salario);
```

```
    soma_salario=soma_salario+fun[i].salario; //aqui é feito o cálculo da soma de  
todos os salários
```

```
    if(fun[i].salario>maior) //aqui é feito a comparação entre os salário para  
verificar qual é o maior
```

```
    {
```

```
        maior=fun[i].salario; //aqui a variável "maior" recebe o valor do salário  
caso o salário seja superior
```

```
    }
```

```
}
```

```
system("cls"); //essa função limpa a tela de impressão
```

```
printf("A soma de todos os salários é igual a R$%.2f",soma_salario); //aqui esta  
sendo impresso a soma dos salários, note que tem a expressão "%.2f" isso significa que eu  
quero que imprima apenas duas casas depois da vírgula
```

```
//OBS: note que a soma está sendo impressa fora do laço de repetição, isso é devido  
ao fato de que se fosse dentro do laço, a variável seria impressa para cada valor de i
```

```
printf("\n\nO(s) funcionário(s) que possui(em) o maior(es) salário(s) é(são):\n");
```

```
for(j=0;j<5;j++) //aqui é outro laço, agora utilizando o contador j
```

```
{
```

```
    if(fun[j].salario==maior) //essa condicional verifica quais funcionários que  
possuem o maior salário
```

```
    {
```

```
        printf("%s que recebe R$ %.2f\n",fun[j].nome,fun[j].salario); //aqui  
imprime o nome e o salário do funcionário com maior salário
```

```

    }

}

for(k=0;k<5;k++) //aqui é outro laço, agora utilizando o contador k
{
    if(fun[k].telefone==0) //essa condicional identifica qual dos números
    telefônicos estão com o valor inicial da variável. NOTA: não estou satisfeito ainda com como
    fiz essa parte.
    {
        printf("\nO funcionário %s deixou o número de telefone em
        branco.",fun[k].nome); //aqui imprime o nome de quem não tem o telefone preenchido
    }
}

save_cadastro=fopen("cadastro_funcionarios.txt","w"); //essa parte aqui serve para
abrir/criar documento para escrita
fwrite(fun,sizeof(struct FUNCIONARIO),5,save_cadastro); //essa parte aqui serve
para escrever as informações no documento
fclose(save_cadastro); //essa parte aqui serve para fechar o documento aberto em
fopen

printf("\n\n\n");// Aqui é detalhe pessoal

system("pause"); //Aqui é um comando para o sistema parar antes de fechar o programa,
no compilador não precisa, mas se for usar o executável precisa.

return 0; //encerra o programa
}

```

Arquivo com os programas criados: [https://drive.google.com/open?id=1zo02-C9dMowO7WvKqpkp11z9RbwmDz\\_Y](https://drive.google.com/open?id=1zo02-C9dMowO7WvKqpkp11z9RbwmDz_Y)