

```
/*  
Nome: Euler Henrique Bandeira Oliveira  
Matrícula: 11821BSI210  
*/
```

```
/*LISTA_7_Estruturas*/
```

```
/*Questão 01*/
```

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

```
typedef struct{  
    int DDD;  
    int numero;
```

```
}Telefone;
```

```
int main(void){  
    Telefone telefone[3] = {0, 0};
```

```
    int i = 0;
```

```
    for(i = 0; i < 3; i++){  
        printf("Digite o DDD e o telefone: ");  
        scanf("%d",&telefone[i].DDD);  
        scanf("%d",&telefone[i].numero);  
    }
```

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

```
    for(i = 0; i < 3; i++){  
        printf("(%d) %d ",telefone[i].DDD, telefone[i].numero);  
        printf("\n");  
    }
```

```
    return 0;  
}
```

/*Questão 02*/

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

```
typedef struct{  
    int DDD;  
    int numero;
```

```
}Telefone;
```

```
int main(void){  
    Telefone telefone[3] = {0, 0};
```

```
    int i = 0;
```

```
    for(i = 0; i < 3; i++){  
        printf("Digite o DDD e o telefone: ");  
        scanf("%d",&telefone[i].DDD);  
        scanf("%d",&telefone[i].numero);  
    }
```

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

```
    for(i = 0; i < 3; i++){  
        printf("(%d) %d ",telefone[i].DDD, telefone[i].numero);  
        printf("\n");  
    }
```

```
    return 0;
```

```
}
```

/*Questão 03*/

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

```
typedef struct{
    char rua[50];
    char cidade[50];
    int  numero;
    char cep[50];
    char estado[4];
    char complemento[50];
}Endereco;
```

```
int main(void){
    Endereco endereco[3];

    int i = 0;

    for(i = 0; i < 3; i++){

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

        printf("Digite a rua: ");
        fgets(endereco[i].rua,sizeof(endereco[i].rua),stdin);
        endereco[i].rua[strcspn(endereco[i].rua,"\n")] = '\0';

        printf("Digite a cidade: ");
        fgets(endereco[i].cidade,sizeof(endereco[i].cidade),stdin);
        endereco[i].cidade[strcspn(endereco[i].cidade,"\n")] = '\0';

        printf("Digite o número: ");
        scanf("%d",&endereco[i].numero);
        getchar();

        printf("Digite o cep: ");
        fgets(endereco[i].cep,sizeof(endereco[i].cep),stdin);
        endereco[i].cep[strcspn(endereco[i].cep,"\n")] = '\0';

        printf("Digite o estado: ");
        fgets(endereco[i].estado,sizeof(endereco[i].estado),stdin);
        endereco[i].estado[strcspn(endereco[i].estado,"\n")] = '\0';

        printf("Digite o complemento: ");
        fgets(endereco[i].complemento,sizeof(endereco[i].complemento),stdin);
        endereco[i].complemento[strcspn(endereco[i].complemento,"\n")] = '\0';

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

    }
}
```

```
for(i = 0; i < 3; i++){

    if(strcmp(endereco[i].estado,"MG") == 0){
        printf("Endereço %d",i+1);
        printf("\n");

        printf("Rua: %s",endereco[i].rua);
        printf("\n");
        printf("Cidade: %s",endereco[i].cidade);
        printf("\n");
        printf("Numero: %d",endereco[i].numero);
        printf("\n");
        printf("Cep: %s ",endereco[i].cep);
        printf("\n");
        printf("Estado: %s",endereco[i].estado);
        printf("\n");
        printf("complemento: %s",endereco[i].complemento);
        printf("\n");
        printf("\n");
    }
}

return 0;
}
```

/*Questão 04*/

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

```
typedef struct{
    char rua[50];
    char cidade[50];
    unsigned int numero;
    char cep[50];
    char estado[4];
    char complemento[50];
}Endereco;
```

```
typedef struct{
    unsigned int DDD;
    unsigned int numero;
}Telefone;
```

```
typedef struct{
    unsigned int dia;
    unsigned int mes;
    unsigned int ano;
}Data;
```

```
typedef struct{
    char nome[50];
    char CPF[20];
    char sexo;
    char estado_civil[20];
    unsigned int salario;
    char rg[20];
    Endereco endereco_comercial;
    Endereco endereco_residencial;
    Telefone telefone;
    Telefone telefone_residencial;
    Data data_de_nascimento;
    Data data_de_ingresso;
}Cadastro;
```

```
int menu(void){
    char opcao = '\0';

    printf("=====\n");
    printf("=====Cadastro=====\n");
    printf("=====\n\n");

    printf("a - Listar todos os cadastros\n");
    printf("b - Cadastrar/Sobrescrever nova pessoa\n");
    printf("c - Listar as pessoas que nasceram depois de 1990\n");
    printf("d - Copiar dados de um cadastro para outro\n");
    printf("\n");

    printf("Digite a opção desejada: ");
    scanf("\n%c",&opcao);

    if(opcao == 'a'){
        return 1;
    }
    if(opcao == 'b'){
        return 2;
    }
    if(opcao == 'c'){
        return 3;
    }
    if(opcao == 'd'){
        return 4;
    }

    return 0;
}
```

```

void listar(Cadastro cadastro[]){
    printf("\n");
    for(int i = 0; i < 5; i++){

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

        printf("Nome: %s\n",cadastro[i].nome);
        printf("CPF: %s\n",cadastro[i].CPF);
        printf("Sexo: %c\n",cadastro[i].sexo);
        printf("Estado Civil: %s\n",cadastro[i].estado_civil);
        printf("Salario: %d\n",cadastro[i].salario);
        printf("RG: %s\n",cadastro[i].rg);

        printf("\nEndereço comercial\n\n");

        printf("Rua: %s\n", cadastro[i].endereco_comercial.rua);
        printf("Cidade: %s\n", cadastro[i].endereco_comercial.cidade);
        printf("Numero: %d\n", cadastro[i].endereco_comercial.numero);
        printf("Cep: %s\n", cadastro[i].endereco_comercial.cep);
        printf("Estado: %s\n",cadastro[i].endereco_comercial.estado);
        printf("Complemento: %s\n", cadastro[i].endereco_comercial.complemento);
        printf("\n");

        printf("Endereço residencial\n\n");

        printf("Rua: %s\n", cadastro[i].endereco_residencial.rua);
        printf("Cidade: %s\n", cadastro[i].endereco_residencial.cidade);
        printf("Numero: %d\n", cadastro[i].endereco_residencial.numero);
        printf("Cep: %s\n", cadastro[i].endereco_residencial.cep);
        printf("Estado: %s\n",cadastro[i].endereco_residencial.estado);
        printf("Complemento: %s\n", cadastro[i].endereco_residencial.complemento);
        printf("\n");

        printf("Telefone: (%d) %d",cadastro[i].telefone.DDD, cadastro[i].telefone.numero);
        printf("\n");
        printf("Telefone residencia: (%d) %d",
cadastro[i].telefone_residencial.DDD,cadastro[i].telefone_residencial.numero);
        printf("\n\n");
    }
return;
}

```

```

void cadastrar(Cadastro cadastro[]){
    int opcao = 0;

    printf("\n");
    printf("Cadastrar / Sobrescrever nova pessoa\n\n");
    printf("Em que posição você deseja cadastrar/Sobrescrever?\n");
    printf("Opções disponíveis 0, 1, 2,3 ou 4: ");

    scanf("%d",&opcao);
    getchar();

    printf("Nome: ");
    fgets(cadastro[opcao].nome,sizeof(cadastro[opcao].nome),stdin);
    cadastro[opcao].nome[strcspn(cadastro[opcao].nome,"\n")] = '\0';

    printf("CPF: ");
    fgets(cadastro[opcao].CPF,sizeof(cadastro[opcao].CPF),stdin);
    cadastro[opcao].CPF[strcspn(cadastro[opcao].CPF,"\n")] = '\0';

    printf("Sexo: ");
    scanf("\n%c",&cadastro[opcao].sexo);
    getchar();

    printf("Estado Civil: ");
    fgets(cadastro[opcao].estado_civil,sizeof(cadastro[opcao].estado_civil),stdin);
    cadastro[opcao].estado_civil[strcspn(cadastro[opcao].estado_civil,"\n")] = '\0';

    printf("Salario: ");
    scanf("%u",&cadastro[opcao].salario);

    printf("RG: ");
    fgets(cadastro[opcao].rg,sizeof(cadastro[opcao].rg),stdin);
    cadastro[opcao].rg[strcspn(cadastro[opcao].rg,"\n")] = '\0';

    printf("\nEndereço comercial\n\n");

    printf("Rua: ");
    fgets(cadastro[opcao].endereco_comercial.rua,sizeof(cadastro[opcao].endereco_comercial.rua),stdin);
    cadastro[opcao].endereco_comercial.rua[strcspn(cadastro[opcao].endereco_comercial.rua,"\n")] = '\0';

    printf("Cidade: ");
    fgets(cadastro[opcao].endereco_comercial.cidade,sizeof(cadastro[opcao].endereco_comercial.cidade),stdin)
;
    cadastro[opcao].endereco_comercial.cidade[strcspn(cadastro[opcao].endereco_comercial.cidade,"\n")] =
'\0';

    printf("Numero: ");
    scanf("%u",&cadastro[opcao].endereco_comercial.numero);
    getchar();

    printf("Cep: ");
    fgets(cadastro[opcao].endereco_comercial.cep,sizeof(cadastro[opcao].endereco_comercial.cep),stdin);
    cadastro[opcao].endereco_comercial.cep[strcspn(cadastro[opcao].endereco_comercial.cep,"\n")] = '\0';

```



```

printf("Estado: ");
fgets(cadastro[opcao].endereco_comercial.estado,sizeof(cadastro[opcao].endereco_comercial.estado),stdin);
cadastro[opcao].endereco_comercial.estado[strcspn(cadastro[opcao].endereco_comercial.estado,"\n")] =
'\0';

printf("Complemento: ");
fgets(cadastro[opcao].endereco_comercial.complemento,sizeof(cadastro[opcao].endereco_comercial.compl
emento),stdin);
cadastro[opcao].endereco_comercial.complemento[strcspn(cadastro[opcao].endereco_comercial.compleme
nto,"\n")] = '\0';

printf("Endereço residencial\n\n");

printf("Rua: ");
fgets(cadastro[opcao].endereco_residencial.rua,sizeof(cadastro[opcao].endereco_residencial.rua),stdin);
cadastro[opcao].endereco_residencial.rua[strcspn(cadastro[opcao].endereco_residencial.rua,"\n")] = '\0';

printf("Cidade: ");
fgets(cadastro[opcao].endereco_residencial.cidade,sizeof(cadastro[opcao].endereco_residencial.cidade),stdi
n);
cadastro[opcao].endereco_residencial.cidade[strcspn(cadastro[opcao].endereco_residencial.cidade,"\n")] =
'\0';

printf("Numero: ");
scanf("%u",&cadastro[opcao].endereco_residencial.numero);
getchar();

printf("Cep: ");
fgets(cadastro[opcao].endereco_residencial.cep,sizeof(cadastro[opcao].endereco_residencial.cep),stdin);
cadastro[opcao].endereco_residencial.cep[strcspn(cadastro[opcao].endereco_residencial.cep,"\n")] = '\0';

printf("Estado: ");
fgets(cadastro[opcao].endereco_residencial.estado,sizeof(cadastro[opcao].endereco_residencial.estado),stdi
n);
cadastro[opcao].endereco_residencial.estado[strcspn(cadastro[opcao].endereco_residencial.estado,"\n")] =
'\0';

printf("Complemento: ");
fgets(cadastro[opcao].endereco_residencial.complemento,sizeof(cadastro[opcao].endereco_residencial.com
plemento),stdin);
cadastro[opcao].endereco_residencial.complemento[strcspn(cadastro[opcao].endereco_residencial.comple
mento,"\n")] = '\0';

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

printf("Digite o DDD: ");
scanf("%u",&cadastro[opcao].telefone.DDD);

printf("Digite o número: ");
scanf("%u",&cadastro[opcao].telefone.numero);

```

```

printf("\nTelefone residencial \n");

printf("Digite o DDD: ");
scanf("%u", &cadastro[opcao].telefone_residencial.DDD);

printf("Digite o número: ");
scanf("%u", &cadastro[opcao].telefone_residencial.numero);
return;
}

void anos_90(Cadastro cadastro[]){
    printf("\n");
    printf("As pessoas que nasceram depois dos anos 90 são: \n\n");

    for(int i = 0; i < 5; i++){
        if(cadastro[i].data_de_nascimento.ano > 1990){
            printf("Nome: %s\n", cadastro[i].nome);
            printf("CPF: %s\n", cadastro[i].CPF);
            printf("Sexo: %c\n", cadastro[i].sexo);
            printf("Estado Civil: %s\n", cadastro[i].estado_civil);
            printf("Salario: %d\n", cadastro[i].salario);
            printf("RG: %s\n", cadastro[i].rg);

            printf("\nEndereço comercial\n\n");

            printf("Rua: %s\n", cadastro[i].endereco_comercial.rua);
            printf("Cidade: %s\n", cadastro[i].endereco_comercial.cidade);
            printf("Numero: %d\n", cadastro[i].endereco_comercial.numero);
            printf("Cep: %s\n", cadastro[i].endereco_comercial.cep);
            printf("Estado: %s\n", cadastro[i].endereco_comercial.estado);
            printf("Complemento: %s\n", cadastro[i].endereco_comercial.complemento);
            printf("\n");

            printf("Endereço residencial\n\n");

            printf("Rua: %s\n", cadastro[i].endereco_residencial.rua);
            printf("Cidade: %s\n", cadastro[i].endereco_residencial.cidade);
            printf("Numero: %d\n", cadastro[i].endereco_residencial.numero);
            printf("Cep: %s\n", cadastro[i].endereco_residencial.cep);
            printf("Estado: %s\n", cadastro[i].endereco_residencial.estado);
            printf("Complemento: %s\n", cadastro[i].endereco_residencial.complemento);
            printf("\n");

            printf("Telefone: (%d) %d", cadastro[i].telefone.DDD, cadastro[i].telefone.numero);
            printf("\n");
            printf("Telefone residencia: (%d) %d",
cadastro[i].telefone_residencial.DDD, cadastro[i].telefone_residencial.numero);
            printf("\n\n");
        }
    }
    printf("\n");
return;
}

```

```

void copiar(Cadastro cadastro[]){
    int origem = 0;
    int destino = 0;

    printf("Insira a origem e o destino dos dados \n");
    printf("Opções disponíveis: 0, 1, 2, 3 ou 4 \n");

    printf("Origem: ");
    scanf("%d",&origem);

    printf("Destino: ");
    scanf("%d",&destino);

    strcpy(cadastro[destino].nome,cadastro[origem].nome);
    strcpy(cadastro[destino].CPF,cadastro[origem].CPF);

    cadastro[destino].sexo = cadastro[origem].sexo;

    cadastro[destino].salario = cadastro[origem].salario;
    strcpy(cadastro[destino].rg,cadastro[origem].rg);

    strcpy(cadastro[destino].endereco_comercial.rua,cadastro[origem].endereco_comercial.rua);
    strcpy(cadastro[destino].endereco_comercial.cidade,cadastro[origem].endereco_comercial.cidade);

    cadastro[destino].endereco_comercial.numero = cadastro[origem].endereco_comercial.numero;

    strcpy(cadastro[destino].endereco_comercial.cep,cadastro[origem].endereco_comercial.cep);
    strcpy(cadastro[destino].endereco_comercial.estado,cadastro[origem].endereco_comercial.estado);
    strcpy(cadastro[destino].endereco_comercial.complemento,cadastro[origem].endereco_comercial.comple
ento);

    strcpy(cadastro[destino].endereco_residencial.rua,cadastro[origem].endereco_residencial.rua);
    strcpy(cadastro[destino].endereco_residencial.cidade,cadastro[origem].endereco_residencial.cidade);

    cadastro[destino].endereco_residencial.numero = cadastro[origem].endereco_residencial.numero;

    strcpy(cadastro[destino].endereco_residencial.cep,cadastro[origem].endereco_residencial.cep);
    strcpy(cadastro[destino].endereco_residencial.estado,cadastro[origem].endereco_residencial.estado);
    strcpy(cadastro[destino].endereco_residencial.complemento,cadastro[origem].endereco_residencial.comple
mento);

    cadastro[destino].telefone.DDD = cadastro[origem].telefone.DDD;
    cadastro[destino].telefone.numero = cadastro[origem].telefone.numero;

    cadastro[destino].telefone_residencial.DDD = cadastro[origem].telefone_residencial.DDD;
    cadastro[destino].telefone_residencial.numero = cadastro[origem].telefone_residencial.numero;

    return;
}

```

```

int main(void){

    Cadastro cadastro[5] = {

        {"Euler", "222.222.222-22", 'M', "Solteiro", 2000, "MG-22.222.222"
        , {"Gétulio Vargas", "Uberlândia", 222, "38222-022", "MG", "Casa"}
        , {"Rondon Pacheco", "Uberlândia", 345, "38345-45", "MG", "Apartamento"}
        , {34, 992345678}, {34, 994356789}, {18, 5, 2000}, {10, 8, 2018}}

        , {"João", "333.333.333-33", 'M', "Solteiro", 3000, "MG-33.333.333"
        , {"Gétulio Vargas", "Uberlândia", 333, "38333-033", "MG", "Casa"}
        , {"Rondon Pacheco", "Uberlândia", 343, "38343-43", "MG", "Apartamento"}
        , {34, 992345673}, {34, 994356783}, {18, 5, 1989}, {10, 8, 2015}}

        , {"Maria", "444.444.444-44", 'F', "Solteira", 4000, "MG-44.444.444"
        , {"Gétulio Vargas", "Uberlândia", 444, "38444-044", "MG", "Casa"}
        , {"Rondon Pacheco", "Uberlândia", 344, "38344-44", "MG", "Apartamento"}
        , {34, 992345674}, {34, 994356789}, {18, 5, 1985}, {10, 8, 2017}}

        , {"José", "555.555.555-55", 'M', "Solteiro", 5000, "MG-55.555.555"
        , {"Gétulio Vargas", "Uberlândia", 555, "38555-055", "MG", "Casa"}
        , {"Rondon Pacheco", "Uberlândia", 345, "38345-45", "MG", "Apartamento"}
        , {34, 992345675}, {34, 994356785}, {18, 5, 2002}, {10, 8, 2019}}

        , {"Henrique", "666.666.666-66", 'M', "Solteiro", 6000, "MG-66.666.666"
        , {"Gétulio Vargas", "Uberlândia", 666, "38666-066", "MG", "Casa"}
        , {"Rondon Pacheco", "Uberlândia", 346, "38346-46", "MG", "Apartamento"}
        , {34, 992345676}, {34, 994356786}, {18, 5, 1999}, {10, 8, 2016}}

    };

    char opcao = '\0';
    int loop = 0;

    while(loop == 0){
        opcao = menu();

        if(opcao == 1){
            listar(cadastro);
        }
        if(opcao == 2){
            cadastrar(cadastro);
        }
        if(opcao == 3){
            anos_90(cadastro);
        }
        if(opcao == 4){
            copiar(cadastro);
        }
    }

    return 0;
}

```

/*Questão 05*/

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

```
typedef struct
{
    char nome[51];
    char matricula[12];
    double notas_provas[3];
    int faltas;
    double media;
} Estudantes;
```

```
int main(void){
    Estudantes estudantes[3];
    double nota_final[3] = {0};
    for(int i = 0; i < 3; i++){
        printf("Estudante %d\n",i+1);

        printf("Nome: ");
        fgets(estudantes[i].nome,sizeof(estudantes[i].nome),stdin);
        estudantes[i].nome[strcspn(estudantes[i].nome,"\n")] = '\0';

        printf("Matrícula: ");
        fgets(estudantes[i].matricula,sizeof(estudantes[i].matricula),stdin);
        estudantes[i].matricula[strcspn(estudantes[i].matricula,"\n")] = '\0';

        printf("Nota da primeira prova : ");
        scanf("%lf",&estudantes[i].notas_provas[0]);

        printf("Nota da segunda prova: ");
        scanf("%lf",&estudantes[i].notas_provas[1]);

        printf("Nota da terceira prova: ");
        scanf("%lf",&estudantes[i].notas_provas[2]);

        printf("Quantidade de faltas: ");
        scanf("%d",&estudantes[i].faltas);
        getchar();

        nota_final[i] = estudantes[i].notas_provas[0] + estudantes[i].notas_provas[1] +
estudantes[i].notas_provas[2];

        printf("\n");
    }

    int posicao1 = 0;
    int posicao2 = 0;
    int posicao3 = 0;
```

```

for(int i = 0; i < 3; i++){
    estudantes[i].media = nota_final[i] / 3;
}

double maior = estudantes[0].media;
double menor = estudantes[0].media;
double maiorp1 = estudantes[0].notas_provas[0];

for(int i = 0; i < 3; i++){
    if(estudantes[i].notas_provas[0] > maiorp1){
        maiorp1 = estudantes[i].notas_provas[0];
        posicao1 = i;
    }

    if(estudantes[i].media > maior){
        maior = estudantes[i].media;
        posicao2 = i;
    }

    if(estudantes[i].media < menor){
        menor = estudantes[i].media;
        posicao3 = i;
    }
}

printf("Aluno com a maior nota na P1 e sua nota: %s, %.2lf\n",estudantes[posicao1].nome,maiorp1);
printf("Aluno com a maior média e sua nota: %s, %.2lf\n",estudantes[posicao2].nome,maior);
printf("Aluno com a menor média e sua nota: %s, %.2lf\n",estudantes[posicao3].nome,menor);
printf("\n");
for(int i = 0; i < 3; i++){

    if(estudantes[i].faltas <= 18 && nota_final[i] >= 60){
        printf("Nome do aluno %d: %s\n",i+1,estudantes[i].nome);
        printf("Situação final: Aprovado\n\n");
    }
    else
    if(estudantes[i].faltas > 18 && nota_final[i] < 60){
        printf("Nome do aluno %d: %s\n",i+1,estudantes[i].nome);
        printf("Situação final: Reprovado por falta e por nota\n\n");
    }
    else
    if(estudantes[i].faltas > 18){
        printf("Nome do aluno %d: %s\n",i+1,estudantes[i].nome);
        printf("Situação final: Reprovado por falta\n\n");
    }
    else
    if(nota_final[i] < 60){
        printf("Nome do aluno %d: %s\n",i+1,estudantes[i].nome);
        printf("Situação final: Reprovado por nota\n\n");
    }
}

return 0;
}

```

/*Questão 06*/

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

```
typedef struct{
    char nome[16];
    float potencia;
    float tempo_ativo;
}Elerodomestico;
```

```
int main(void){

    float consumo_total = 0;
    float consumo[5] = {0};
    float consumo_relativo[5] = {0};
    float tempo_dias = 0;
    Elerodomestico eletrodomestico[5];

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

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

        printf("Nome: ");
        fgets(eletrodomestico[i].nome,sizeof(eletrodomestico[i].nome),stdin);
        eletrodomestico[i].nome[strcspn(eletrodomestico[i].nome,"\n")] = '\0';

        printf("Potencia(em KW): ");
        scanf("%f",&eletrodomestico[i].potencia);

        printf("Tempo ativo: ");
        scanf("%f",&eletrodomestico[i].tempo_ativo);
        getchar();
    }
    printf("\n");
    printf("Tempo em dias: ");
    scanf("%f",&tempo_dias);

    for(int i = 0; i < 5; i++){
        consumo[i] = eletrodomestico[i].potencia * eletrodomestico[i].tempo_ativo;
        consumo_total = consumo[i] + consumo_total;
    }
    consumo_total = consumo_total * tempo_dias;

    for(int i = 0; i < 5; i++){
        consumo_relativo[i] = ((consumo[i] * tempo_dias / consumo_total) * 100);
    }
    printf("\n");
    printf("Consumo total na casa: %.2f",consumo_total);
    printf("\n");
```

```
for(int i = 0; i < 5; i++){  
  
    printf("\n");  
    printf("Eletrodomestico %d\n",i+1);  
    printf("Consumo: %.2f",consumo[i]);  
    printf("\n");  
    printf("Consumo relativo: %.2f", consumo_relativo[i]);  
    printf("\n");  
}  
return 0;  
}
```


/*Questão 07*/

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

```
typedef struct{
    char nome[16];
    float potencia;
    float tempo_ativo;
}Elerodomestico;
```

```
int main(void){

    float consumo_total = 0;
    float consumo[5] = {0};
    float consumo_relativo[5] = {0};
    float tempo_dias = 0;
    Elerodomestico eletrodomestico[5];

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

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

        printf("Nome: ");
        fgets(eletrodomestico[i].nome,sizeof(eletrodomestico[i].nome),stdin);
        eletrodomestico[i].nome[strcspn(eletrodomestico[i].nome,"\n")] = '\0';

        printf("Potencia(em KW): ");
        scanf("%f",&eletrodomestico[i].potencia);

        printf("Tempo ativo: ");
        scanf("%f",&eletrodomestico[i].tempo_ativo);
        getchar();
    }
    printf("\n");
    printf("Tempo em dias: ");
    scanf("%f",&tempo_dias);

    for(int i = 0; i < 5; i++){
        consumo[i] = eletrodomestico[i].potencia * eletrodomestico[i].tempo_ativo;
        consumo_total = consumo[i] + consumo_total;
    }
    consumo_total = consumo_total * tempo_dias;

    for(int i = 0; i < 5; i++){
        consumo_relativo[i] = ((consumo[i] * tempo_dias / consumo_total) * 100);
    }
    printf("\n");
    printf("Consumo total na casa: %.2f",consumo_total);
    printf("\n");
}
```

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

    printf("\n");
    printf("Eletrodomestico %d\n",i+1);
    printf("Consumo: %.2f",consumo[i]);
    printf("\n");
    printf("Consumo relativo: %.2f", consumo_relativo[i]);
    printf("\n");
}

printf("\n");
printf("Endereço do consumo_total: %u\n",&consumo_total);
printf("Endereço do consumo[0]: %u\n",&consumo[0]);
printf("Endereço do consumo[1]: %u\n",&consumo[1]);
printf("Endereço do consumo[2]: %u\n",&consumo[2]);
printf("Endereço do consumo[3]: %u\n",&consumo[3]);
printf("Endereço do consumo[4]: %u\n",&consumo[4]);

printf("Endereço do consumo_relativo[0]: %u\n",&consumo_relativo[0]);
printf("Endereço do consumo_relativo[1]: %u\n",&consumo_relativo[1]);
printf("Endereço do consumo_relativo[2]: %u\n",&consumo_relativo[2]);
printf("Endereço do consumo_relativo[3]: %u\n",&consumo_relativo[3]);
printf("Endereço do consumo_relativo[4]: %u\n",&consumo_relativo[4]);

printf("Endereço do tempo_dias: %u\n",&tempo_dias);

return 0;
}
```

1	'M'	eletrodomestico[0].nome	char
2	'i'		
3	'c'		
4	'r'		
5	'o'		
6	'_'		
7	'o'		
8	'n'		
9	'd'		
10	'a'		
11	's'		
12	'\0'		
13	lx		
14	lx		
15	lx		
16	lx		
17		eletrodomestico[0].potencia	float
18	8.5		
19			
20			
21		eletrodomestico[0].tempo_ativo	float
22	8.5		
23			
24			
25	'G'	eletrodomestico[1].nome	char
26	'e'		
27	'l'		
28	'a'		
29	'd'		
30	'e'		
31	'i'		
32	'r'		
33	'a'		
34	'\0'		
35	lx		
36	lx		
37	lx		
38	lx		
39	lx		
40	lx		
41		eletrodomestico[1].potencia	float
42	10.5		
43			
44			
45		eletrodomestico[1].tempo_ativo	float
46	10.5		
47			
48			
49	'F'	eletrodomestico[2].nome	char
50	'o'		
51	'g'		

52	'ã'		
53	'o'		
54	'\0'		
55	lx		
56	lx		
57	lx		
58	lx		
59	lx		
60	lx		
61	lx		
62	lx		
63	lx		
64	lx		
65		eletrodomestico[2].potencia	float
66			
67	12.5		
68			
69		eletrodomestico[2].tempo_ativo	float
70	12.5		
71			
72			
73	'C'	eletrodomestico[3].nome	char
74	'a'		
75	'f'		
76	'e'		
77	't'		
78	'e'		
79	'i'		
80	'r'		
81	'a'		
82	'\0'		
83	lx		
84	lx		
85	lx		
86	lx		
87	lx		
88	lx		
89		eletrodomestico[3].potencia	float
90			
91	14.5		
92			
93		eletrodomestico[3].tempo_ativo	float
94			
95	14.5		
96			
97	'S'	eletrodomestico[4].nome	char
98	'a'		
99	'n'		
100	'd'		
101	'u'		
102	'i'		

103	'c'		
104	'h'		
105	'e'		
106	'i'		
107	'r'		
108	'a'		
109	'\0'		
110	lx		
111	lx		
112	lx		
113	eletrodomestico[4].potencia	float	
114			
115	16.5		
116			
117	eletrodomestico[4].tempo_ativo	float	
118			
119	16.5		
120			
121	consumo_total	float	
122	12318.75		
123			
124			
125	consumo[0]	float	
126	72.25		
127			
128			
129	consumo[1]	float	
130	110.25		
131			
132			
133	consumo[2]	float	
134	156.25		
135			
136			
137	consumo[3]	float	
138	210.25		
139			
140			
141	consumo[4]	float	
142	272.25		
143			
144			
145	consumo_relativo[0]	float	
146	8.80		
147			
148			
149	consumo_relativo[1]	float	
150	13.42		
151			
152			
153	consumo_relativo[2]	float	

154	19.03		
155			
156			
157		consumo_relativo[3]	float
158	25.60		
159			
160			
161		consumo_relativo[4]	float
162	33.15		
163			
164			
165		tempo_dias	float
166	10.0		
167			
168			

```
/*LISTA_8-_Funções*/
```

```
/*Questão 1*/
```

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

```
void DesenhaLinha(void){  
    printf("=====\n");  
    return;  
}
```

```
int main(void){  
  
    for(int i = 0; i < 20; i++){  
        DesenhaLinha();  
    }  
    return 0;  
}
```

```
/*Questão 2*/
```

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

```
void DesenhaLinha(int qtd_caracteres){  
    for(int i = 0; i < qtd_caracteres; i++){  
        printf("=");  
    }  
    printf("\n");  
    return;  
}
```

```
int main(void){  
    int qtd_caracteres = 0;  
  
    printf("Quantos sinais de igual cada linha deve ter? \n");  
    scanf("%d",&qtd_caracteres);  
  
    for(int i = 0; i < 20; i++){  
        DesenhaLinha(qtd_caracteres);  
    }  
    return 0;  
}
```

```
/*Questão 3*/
```

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

```
int fatorial(int x){  
    int fat = 0;  
    fat = 1;  
  
    for(int i = 1; i <= x; i++){  
        fat = i * fat;  
    }  
    return fat;  
}
```

```
void DesenhaLinha(void){  
    printf("=====\n");  
    return;  
}
```

```
int main(void){  
    int x = 0;  
    int fat = 0;  
  
    for(int i = 0; i < 5; i++){  
        DesenhaLinha();  
        printf("Fatorial\n");  
        DesenhaLinha();  
  
        printf("Digite o fatorial a ser calculado: ");  
        scanf("%d",&x);  
  
        fat = fatorial(x);  
        printf("Fat de %d: %d\n",x,fat);  
    }  
    return 0;  
}
```


/*Questão 4*/

#include <stdio.h>

float potencia(int base, int expoente){

float x = 1;

if(expoente > 0){

for(int i = 0; i < expoente; i++){

x = base * x;

}

}

if(expoente < 0){

expoente = expoente * -1;

for(int i = 0; i < expoente; i++){

x = base * x;

}

x = 1 / x;

}

return x;

}

int main(void){

int base = 0;

int expoente = 0;

float resultado = 0;

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

printf("Potência\n");

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

printf("Digite a base da potência: ");

scanf("%d",&base);

printf("Digite o expoente da potência: ");

scanf("%d",&expoente);

resultado = potencia(base,expoente);

printf("%.2d ^ %.2d = %.2f \n",base,expoente,resultado);

return 0;

}

/*Questão 5*/

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

```
#include <math.h>
```

```
int quadrado_perfeito(double x){  
    x = sqrt(x);  
    double inteiro = 0;  
    double fracionario = 0;  
  
    fracionario = modf(x,&inteiro);  
    if(fracionario == 0){  
        return 1;  
    }  
    else{  
        return 0;  
    }  
}
```

```
int main(void){  
    double numero = 0;  
    int opcao = 0;  
  
    printf("Digite o número: ");  
    scanf("%lf",&numero);  
  
    opcao = quadrado_perfeito(numero);  
    if(opcao == 1){  
        printf("True\n");  
    }  
    else{  
        printf("False\n");  
    }  
    return 0;  
}
```

/*Questão 6*/

#include <stdio.h>

#include <stdlib.h>

```
void DesenhaLinha(void){  
    printf("=====\n");  
    return;  
}
```

```
double fatorial(int x){  
    double fat = 0;  
    fat = 1;  
  
    for(int i = 1; i <= x; i++){  
        fat = i * fat;  
    }  
    return fat;  
}
```

```
double numero_neperiano(int qtd_termos){  
  
    double e = 0;  
  
    for(int i = 0; i < qtd_termos; i++){  
        e = 1 / fatorial(i) + e;  
    }  
    return e;  
}
```

```
int main(void){  
    int qtd_termos = 0;  
    double e = 0;  
  
    for(int i = 0; i < 5; i++){  
        DesenhaLinha();  
        printf("Número neperiano\n");  
        DesenhaLinha();  
  
        printf("Digite a quantidade de termos a serem calculados: ");  
        scanf("%d",&qtd_termos);  
  
        e = numero_neperiano(qtd_termos);  
        printf("O número neperiano é: %lf\n",e);  
    }  
  
    return 0;  
}
```

/*Questão 7*/

#include <stdio.h>

```
void troque(int* a, int* b){
    int temp = 0;

    temp = *a;
    *a = *b;
    *b = temp;

    return;
}
```

```
int main(void){
    printf("=====\n");
    printf("Troque\n");
    printf("=====\n");

    int a = 0;
    int b = 0;

    printf("Digite o valor de a: ");
    scanf("%d",&a);

    printf("Digite o valor de b: ");
    scanf("%d",&b);

    troque(&a,&b);
    printf("Os valores trocados são: %d %d ",a,b);
    printf("\n");

    return 0;
}
```

/*Questão 8*/

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

```
float incp(float y, float x){  
    if(x > 0){  
        x = (x * y / 100) + x;  
    }  
    else{  
        x = (x * y/ 100) - x;  
    }  
    return x;  
}
```

```
int main(void){  
  
    float x = 0;  
    float y = 0;  
  
    printf("Digite o valor de x: ");  
    scanf("%f",&x);  
    if(x > 0){  
        printf("Digite o quanto você deseja aumentar o valor de x (em porcentagem): ");  
        scanf("%f",&y);  
    }  
    else{  
        printf("Digite o quanto você deseja diminuir o valor de x (em porcentagem): ");  
        scanf("%f",&y);  
    }  
  
    x = incp(y,x);  
    printf("O novo valor de x é: %.2f ",x);  
    printf("\n");  
    return 0;  
}
```

```
/*Questão 9*/
```

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

```
void incp(float* y, float* x){  
    if(x > 0){  
        *x = (*x * *y / 100) + *x;  
    }  
    else{  
        *x = (*x * *y / 100) - *x;  
    }  
    return ;  
}
```

```
int main(void){  
  
    float x = 0;  
    float y = 0;  
  
    printf("Digite o valor de x: ");  
    scanf("%f",&x);  
    if(x > 0){  
        printf("Digite o quanto você deseja aumentar o valor de x (em porcentagem): ");  
        scanf("%f",&y);  
    }  
    else{  
        printf("Digite o quanto você deseja diminuir o valor de x (em porcentagem): ");  
        scanf("%f",&y);  
    }  
  
    incp(&y,&x);  
  
    printf("O novo valor de x é: %.2f ",x);  
    printf("\n");  
    return 0;  
}
```

/*Questão 10*/

#include <stdio.h>

typedef struct {

int x;

int y;

}Ponto;

void imprime_ponto(Ponto ponto){

printf("(%d, %d)", ponto.x, ponto.y);

printf("\n");

return;

}

int main(void){

Ponto ponto;

printf("Digite o ponto x: ");

scanf("%d",&ponto.x);

printf("Digite o ponto y: ");

scanf("%d",&ponto.y);

imprime_ponto(ponto);

return 0;

}

```
/*Questão 11*/
```

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

```
typedef struct {
```

```
    int x;
```

```
    int y;
```

```
}Ponto;
```

```
Ponto soma_ponto(Ponto p1, Ponto p2){
```

```
    Ponto p3;
```

```
    p3.x = p1.x + p2.x;
```

```
    p3.y = p1.y + p2.y;
```

```
    return p3;
```

```
}
```

```
void imprime_ponto(Ponto p1,Ponto p2, Ponto p3){
```

```
    printf("A soma de (%d, %d) com (%d, %d) é: (%d,%d) ",p1.x,p1.y,p2.x,p2.y,p3.x,p3.y);
```

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

```
    return;
```

```
}
```

```
int main(void){
```

```
    Ponto p1;
```

```
    Ponto p2;
```

```
    Ponto p3;
```

```
    printf("Digite o ponto x1: ");
```

```
    scanf("%d",&p1.x);
```

```
    printf("Digite o ponto y1: ");
```

```
    scanf("%d",&p1.y);
```

```
    printf("Digite o ponto x2: ");
```

```
    scanf("%d",&p2.x);
```

```
    printf("Digite o ponto y2: ");
```

```
    scanf("%d",&p2.y);
```

```
    p3 = soma_ponto(p1,p2);
```

```
    imprime_ponto(p1,p2,p3);
```

```
    return 0;
```

```
}
```


/*Questão 12*/

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

```
typedef struct {  
    int x;  
    int y;
```

```
}Ponto;
```

```
void soma_ponto(Ponto p1, Ponto p2, Ponto *p3){  
    (*p3).x = p1.x + p2.x; //(*p3).x = p3->x  
    (*p3).y = p1.y + p2.y;  
    return;  
}
```

```
void imprime_ponto(Ponto p1,Ponto p2, Ponto p3){  
    printf("A soma de (%d, %d) com (%d, %d) é: (%d,%d) ",p1.x,p1.y,p2.x,p2.y,p3.x,p3.y);  
    printf("\n");  
    return;  
}
```

```
int main(void){
```

```
    Ponto p1;  
    Ponto p2;  
    Ponto p3;
```

```
    printf("Digite o ponto x1: ");  
    scanf("%d",&p1.x);
```

```
    printf("Digite o ponto y1: ");  
    scanf("%d",&p1.y);
```

```
    printf("Digite o ponto x2: ");  
    scanf("%d",&p2.x);
```

```
    printf("Digite o ponto y2: ");  
    scanf("%d",&p2.y);
```

```
    soma_ponto(p1,p2,&p3);  
    imprime_ponto(p1,p2,p3);
```

```
    return 0;  
}
```

/*Questão 13*/

#include <stdio.h>

```
typedef struct {  
    int x;  
    int y;  
}Ponto;
```

```
int calc_area(Ponto p1, Ponto p2){
```

```
    int base = 0;
```

```
    int altura = 0;
```

```
    int area = 0;
```

```
    if(p1.x > p2.x){
```

```
        base = p1.x - p2.x;
```

```
    }
```

```
    else{
```

```
        base = p2.x - p1.x;
```

```
    }
```

```
    if(p1.y > p2.y){
```

```
        altura = p1.y - p2.y;
```

```
    }
```

```
    else{
```

```
        altura = p2.y - p1.y;
```

```
    }
```

```
    area = base * altura;
```

```
    return area;
```

```
}
```

```
int main(void){
```

```
    Ponto p1;
```

```
    Ponto p2;
```

```
    int area = 0;
```

```
    printf("Digite o ponto x1: ");
```

```
    scanf("%d",&p1.x);
```

```
    printf("Digite o ponto y1: ");
```

```
    scanf("%d",&p1.y);
```

```
    printf("Digite o ponto x2: ");
```

```
    scanf("%d",&p2.x);
```

```
    printf("Digite o ponto y2: ");
```

```
    scanf("%d",&p2.y);
```

```
    area = calc_area(p1,p2);
```

```
    printf("A área o retângulo definido por (%d,%d) e (%d,%d) é %d \n",p1.x,p1.y, p2.x, p2.y,area);
```

```
    return 0;
```

```
}
```

/*Questão 14*/

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

```
typedef struct {  
    int x;  
    int y;  
}Ponto;
```

```
void multiplica(Ponto* p1, int constante){  
    p1->x = p1->x * constante;  
    p1->y = p1->y * constante;  
return;  
}
```

```
int main(void){  
    Ponto p1;  
  
    int constante = 0;  
  
    printf("Digite o valor do ponto x: ");  
    scanf("%d",&p1.x);  
  
    printf("Digite o valor do ponto y: ");  
    scanf("%d",&p1.y);  
  
    printf("Digite a constante: ");  
    scanf("%d",&constante);  
  
    printf("Resultado: (%d,%d) * %d = ",p1.x,p1.y,constante);  
    multiplica(&p1,constante);  
    printf("(%d,%d)",p1.x,p1.y);  
    printf("\n");  
return 0;  
}
```

/*Questão 15*/

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

```
typedef struct {  
    int x;  
    int y;  
}Ponto;
```

```
void inc_dir(Ponto* ponto,char c){  
    if(c == 'n'){  
        ponto->y = ponto->y + 1;  
    }  
    if(c == 's'){  
        ponto->y = ponto->y - 1;  
    }  
    if(c == 'l'){  
        ponto->x = ponto->x - 1;  
    }  
    if(c == 'o'){  
        ponto->x = ponto->x + 1;  
    }  
    return;  
}
```

```
int main(void){  
    char c = '\\0';  
  
    Ponto ponto;  
  
    printf("Digite o valor de x: ");  
    scanf("%d",&ponto.x);  
  
    printf("Digite o valor de y: ");  
    scanf("%d",&ponto.y);  
  
    printf("Digite a direção: ");  
    scanf("\\n%c",&c);  
  
    inc_dir(&ponto,c);  
  
    printf("A nova coordenada é: %d %d",ponto.x,ponto.y);  
    printf("\\n");  
    return 0;  
}
```

/*Questão 16*/

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

```
typedef struct {
    int x;
    int y;
}Ponto;
```

```
void inc_diag(Ponto** ponto, char direcao[]){
    if(strcmp(direcao, "sudeste") == 0){
        (**ponto).x = (**ponto).y - 1;
        (**ponto).x = (**ponto).x - 1;
    }
    if(strcmp(direcao, "sudoeste") == 0){
        (**ponto).y = (**ponto).y - 1;
        (**ponto).x = (**ponto).x + 1;
    }
    if(strcmp(direcao, "nordeste") == 0){
        (**ponto).y = (**ponto).y + 1;
        (**ponto).x = (**ponto).x - 1;
    }
    if(strcmp(direcao, "noroeste") == 0){
        (**ponto).y = (**ponto).y + 1;
        (**ponto).x = (**ponto).x + 1;
    }
    return;
}
```

```
void inc_dir(Ponto* ponto, char c1, char c2){
    char direcao[15] = {"\0"};

    if(c1 == 's' && c2 == 'l'){
        strcpy(direcao, "sudeste");
        inc_diag(&ponto, direcao);
    }
    if(c1 == 's' && c2 == 'o' ){
        strcpy(direcao, "sudoeste");
        inc_diag(&ponto, direcao);
    }
    if(c1 == 'n' && c2 == 'l'){
        strcpy(direcao, "nordeste");
        inc_diag(&ponto, direcao);
    }
    if(c1 == 'n' && c2 == 'o'){
        strcpy(direcao, "noroeste");
        inc_diag(&ponto, direcao);
    }

    return;
}
```

```
int main(void){

    Ponto ponto;

    char c1 = '\0';
    char c2 = '\0';

    printf("Digite o valor de x: ");
    scanf("%d",&ponto.x);

    printf("Digite o valor de y: ");
    scanf("%d",&ponto.y);

    printf("Digite a primeira direção (NORTE OU SUL): ");
    scanf("\n%c",&c1);

    printf("Digite a segunda direção (LESTE OU OESTE): ");
    scanf("\n%c",&c2);

    inc_dir(&ponto,c1,c2);

    printf("A nova coordenada é: (%d, %d)",ponto.x,ponto.y);
    printf("\n");
return 0;
}
```

/*Questão 17*/

```
#include <stdio.h>

void imprime_vet_int(int vet1[], int n1){
    for(int i = 0; i < n1; i++){
        printf("%d ",vet1[i]);
    }
    printf("\n");
return;
}

void imprime_vet_double(double vet2[], int n2){
    for(int i = 0; i < n2; i++){
        printf("%.2lf ", vet2[i]);
    }
    printf("\n");
return;
}

void imprime_vet_float(float vet3[], int n3){
    for(int i = 0; i < n3; i++){
        printf("%.2f ",vet3[i]);
    }
    printf("\n");
return;
}

int main(void){
    int vet1[10] = {0};
    double vet2[10] = {0};
    float vet3[10] = {0};

    printf("Digite os valores do vetor inteiro \n");
    for(int i = 0; i < 10; i++){
        scanf("%d",&vet1[i]);
    }

    printf("Digite os valores do vetor double\n");
    for(int i = 0; i < 10; i++){
        scanf("%lf",&vet2[i]);
    }

    printf("Digite os valores do vetor float \n");
    for(int i = 0; i < 10; i++){
        scanf("%f",&vet3[i]);
    }
    printf("Vetor inteiro: ");
    imprime_vet_int(vet1,10);

    printf("Vetor double: ");
    imprime_vet_double(vet2,10);

    printf("Vetor float: ");
    imprime_vet_float(vet3,10);
return 0;
}
```

/*Questão 18*/

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

```
void maior_menor(double vet[], double* maior, double* menor){
    *maior = vet[0];
    *menor = vet[0];

    for(int i = 0; i < 10; i++){
        if(vet[i] > *maior){
            *maior = vet[i];
        }
        if(vet[i] < *menor){
            *menor = vet[i];
        }
    }
    return;
}
```

```
void imprime_vet_double(double vet[], int n2){
    for(int i = 0; i < n2; i++){
        printf("Valor de vet_double[%d]: %.2lf",i,vet[i]);
        printf("\n");
    }
    printf("\n");
    return;
}
```

```
int main(void){
    double vet[10] = {0};
    double maior = 0;
    double menor = 0;

    printf("Digite 10 valores\n");
    for(int i = 0; i < 10; i++){
        scanf("%lf",&vet[i]);
    }

    maior_menor(vet,&maior,&menor);
    imprime_vet_double(vet,10);
    printf("Maior valor: %.2lf\n",maior);
    printf("Menor valor: %.2lf\n",menor);

    return 0;
}
```


/*Questão 19*/

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

```
int negativos(float* vet, int N){
    int cont = 0;

    for(int i = 0; i < N; i++){
        if(vet[i] < 0){
            cont++;
        }
    }
    return cont;
}
```

```
int main(void){
    float vet[10] = {0};
    int qtd_negativos = 0;

    printf("Digite 10 valores\n");
    for(int i = 0; i < 10; i++){
        printf("Digite o valor de vet[%d]: ", i);
        scanf("%f", &vet[i]);
    }

    qtd_negativos = negativos(vet, 10);
    printf("O vetor lido possui %d valores negativos: ", qtd_negativos);
    printf("\n");

    return 0;
}
```

/*Questão 20*/

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

```
void copiarvet(int vet_origem[],int vet_destino[],int n){
    for(int i = 0; i < n; i++){
        vet_destino[i] = vet_origem[i];
    }
    return;
}
```

```
void imprime_vet_int(int vet[], int n){
    for(int i = 0; i < n; i++){
        printf("%d ",vet[i]);
    }
    printf("\n");
    return;
}
```

```
int main(void){
    int vet1[10] = {0};
    int vet2[10] = {0};

    printf("Digite 10 valores\n");
    for(int i = 0; i < 10; i++){
        scanf("%d",&vet1[i]);
    }

    copiarvet(vet1,vet2,10);

    printf("Vetor original: ");
    imprime_vet_int(vet1,10);

    printf("Vetor com a cópia: ");
    imprime_vet_int(vet2,10);
    return 0;
}
```

/*Questão 21*/

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

```
void imprime_vet(int vet[], int n){  
    for(int i = 0; i < n; i++){  
        printf("%d ",vet[i]);  
    }  
    printf("\n");  
    return;  
}
```

```
void mult_vet(int vet_origem[], int escalar, int n){  
    for(int i = 0 ; i < n; i++){  
        vet_origem[i] = vet_origem[i] * escalar;  
    }  
    return;  
}
```

```
int main(void){  
    int vet[10] = {0};  
    int escalar = 0;  
  
    printf("Digite 10 números\n");  
    for(int i = 0; i < 10; i++){  
        scanf("%d",&vet[i]);  
    }  
  
    printf("Digite o escalar: ");  
    scanf("%d",&escalar);  
  
    printf("Vetor antes da multiplicação: ");  
    imprime_vet(vet,10);  
  
    mult_vet(vet, escalar, 10);  
  
    printf("Vetor depois da multiplicação: ");  
    imprime_vet(vet,10);  
    return 0;  
}
```

/*Questão 22*/

#include <stdio.h>

#include<stdlib.h>

```
void abs_vet(int vet[], int n){
    for(int i = 0; i < n; i++){
        vet[i] = abs(vet[i]);
    }
}
```

```
void imprime_vet_int(int vet[], int n){
    for(int i = 0; i < n; i++){
        printf("%d ",vet[i]);
    }
    printf("\n");
return;
}
```

```
int main(void){
    int vet[10] = {0};

    printf("Digite 10 valores\n");
    for(int i = 0; i < 10; i++){
        scanf("%d",&vet[i]);
    }

    abs_vet(vet,10);

    imprime_vet_int(vet,10);
return 0;
}
```

/*Questão 23*/

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

```
void imprime_vet_int(int vet[], int n){
    for(int i = 0; i < n; i++){
        printf("%d ",vet[i]);
    }
    printf("\n");
return;
}
```

```
void maior_menor(int vet[], int* maior, int* menor, int n){
    *maior = vet[0];
    *menor = vet[0];

    for(int i = 0; i < n; i++){
        if(vet[i] > *maior){
            *maior = vet[i];
        }
        if(vet[i] < *menor){
            *menor = vet[i];
        }
    }
return;
}
```

```
int negativos(int vet[], int n){
    int cont = 0;

    for(int i = 0; i < n; i++){
        if(vet[i] < 0){
            cont++;
        }
    }
return cont;
}
```

```
int* copia_vet_abs(int vet[],int n){
    int *vet_abs =NULL;

    vet_abs = calloc(n,sizeof(double));

    for(int i = 0; i < n; i++){
        vet_abs[i] = vet[i];
    }
return vet_abs;
free(vet_abs);
}
```

```
int* abs_vet(int vet[], int n){
    for(int i = 0; i < n; i++){
        vet[i] = abs(vet[i]);
    }
    return vet;
}
```

```
int* copia_vet(int vet[],int n){
    int *vet2 =NULL;

    vet2 = calloc(n,sizeof(double));

    for(int i = 0; i < n; i++){
        vet2[i] = vet[i];
    }
    return vet2;
    free(vet2);
}
```

```
void mult_vet(int vet_origem[], int escalar, int n){
    for(int i = 0 ; i < n; i++){
        vet_origem[i] = vet_origem[i] * escalar;
    }
    return;
}
```

```
int main(void){
    int* vet = NULL;
    int* vet2 = NULL;
    int* vet_abs = NULL;

    int maior = 0;
    int menor = 0;

    int n = 0;
    int qtd_negativos = 0;
    int escalar = 0;

    printf("Entre com o tamanho do vetor: ");
    scanf("%d",&n);

    vet = calloc(n,sizeof(int));

    for(int i = 0;i < n; i++){
        printf("Entre com o elemento %d: ",i+1);
        scanf("%d",&vet[i]);
    }
    printf("\n\n");

    printf("Vetor: ");
    imprime_vet_int(vet,n);
}
```

```

maior_menor(vet,&maior,&menor,n);
printf("Maior : %d\n",maior);
printf("Menor : %d\n",menor);

qtd_negativos = negativos(vet,n);
printf("Número de negativos: %d",qtd_negativos);
printf("\n");

vet_abs = copia_vet_abs(vet,n);

printf("Valor absoluto: ");
abs_vet(vet_abs,n);
imprime_vet_int(vet_abs,n);
printf("\n\n");

printf("Copiando para outro vetor(alocado dinamicamente)\n");

vet2 = copia_vet(vet,n);

printf("Vetor original: ");
for(int i = 0; i < n; i++){
    printf("%d ",vet[i]);
}
printf("\n");

printf("Vetor copiado: ");
for(int i = 0; i < n; i++){
    printf("%d ",vet2[i]);
}
printf("\n\n");

printf("Entre com o escalar para multiplicar o vetor original: ");
scanf("%d",&escalar);

printf("Vetor antes da multiplicação: ");
imprime_vet_int(vet,n);

mult_vet(vet, escalar,n);

printf("Vetor depois da multiplicação: ");
imprime_vet_int(vet,n);

free(vet);
free(vet2);
free(vet_abs);

return 0;
}

```

/*Questão 24*/

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

```
int* aloca_inteiro(int n){
    int *vet = NULL;

    vet = calloc(n, sizeof(int));

    return vet;
    free(vet);
}
```

```
void imprime_vet(int vet[], int n){
    for(int i = 0; i < n; i++){
        printf("%d ",vet[i]);
    }
    printf("\n");
return;
}
```

```
int main(void){
    int *vet = NULL;
    int n = 0;

    printf("Quantos números você deseja digitar? ");
    scanf("%d",&n);

    vet = aloca_inteiro(n);

    printf("Vetor lido: ");
    imprime_vet(vet,n);

    free(vet);
return 0;
}
```


/*Questão 25*/

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

```
double* copia_vet(double* vet,int n){
    double *vet2 =NULL;

    vet2 = calloc(n,sizeof(double));

    for(int i = 0; i < n; i++){
        vet2[i] = vet[i];
    }
    return vet2;
    free(vet2);
}
```

```
int main(void){
    double *vet = NULL;
    double *vet2 = NULL;
    int n = 0;

    printf("Quantos números você deseja ler?\n");
    scanf("%d",&n);

    vet = calloc(n,sizeof(double));

    printf("Digite os números: ");
    for(int i = 0; i < n; i++){
        scanf("%lf",&vet[i]);
    }

    vet2 = copia_vet(vet,n);

    printf("Vetor original: ");
    for(int i = 0; i < n; i++){
        printf("%.2lf ",vet[i]);
    }
    printf("\n");

    printf("Vetor copiado: ");
    for(int i = 0; i < n; i++){
        printf("%.2lf ",vet2[i]);
    }
    printf("\n");

    free(vet);
    free(vet2);

    return 0;
}
```

/*Questão 26*/

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

```
double* to_double(int *vet_int,int n){

    double *vet_double = NULL;
    vet_double = calloc(n,sizeof(double));

    for(int i=0; i < n; i++){
        vet_double[i] = (double) vet_int[i];
    }

    return vet_double;
    free(vet_double);
}
```

```
int main(void){
    int *vet_int = NULL;
    double *vet_double = NULL;
    int n = 0;

    printf("Quantos valores deseja ler?: ");
    scanf("%d",&n);

    vet_int = calloc(n,sizeof(int));
    for(int i = 0; i < n;i++){
        scanf("%d",&vet_int[i]);
    }
    printf("\n");

    vet_double = to_double(vet_int,n);
    printf("Vetor double: ");
    for(int i = 0; i < n; i++){
        printf("%.2lf ",vet_double[i]);
    }
    printf("\n");

    free(vet_int);
    free(vet_double);

    return 0;
}
```

/*Questão 27*/

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

```
typedef struct{
    int x;
    int y;
}Ponto;
```

```
void pontos(Ponto* ponto, int n){
    for(int i = 0; i < n; i++){
        ponto[i].x = 0;
        ponto[i].y = 0;
    }
    return;
}
```

```
int main(void){
    Ponto* ponto = NULL;
    int n = 0;

    printf("Quantos pontos zerados vc deseja imprimir?: ");
    scanf("%d",&n);

    ponto = malloc(n * sizeof(Ponto));

    pontos(ponto,n);

    for(int i = 0; i < n; i++){
        printf("(%d,%d)",ponto[i].x,ponto[i].y);
        printf("\n");
    }

    free(ponto);

    return 0;
}
```

/*Questão 28*/

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

```
void copiarvet(double* vet_origem,double* vet_destino,int n){

    vet_destino = calloc(n,sizeof(double));

    for(int i = 0; i < n; i++){
        vet_destino[i] = vet_origem[i];
    }

    {
        printf("Endereços durante a execução da função\n");
        printf("%lu",&vet_origem);
        printf("\n");
        printf("%lu",&vet_destino);
        printf("\n");
        printf("%lu",&n);
        printf("\n");
    }

    free(vet_destino);
    return;
}
```

```
int main(void){
    double* vet_origem = NULL;
    double* vet_destino = NULL;
    int n = 0;

    printf("Quantos números você deseja ler?\n");
    scanf("%d",&n);

    vet_origem = calloc(n,sizeof(double));

    printf("Digite os números: ");
    for(int i = 0; i < n; i++){
        scanf("%lf",&vet_origem[i]);
    }

    {
        printf("Endereços antes da execução da função\n");
        printf("%lu",&vet_origem);
        printf("\n");
        printf("%lu",&vet_destino);
        printf("\n");
        printf("%lu",&n);
        printf("\n");
    }

    copiarvet(vet_origem,vet_destino,n);
```

```

{
    printf("Endereços depois da execução da função\n");
    printf("%lu",&vet_origem);
    printf("\n");
    printf("%lu",&vet_destino);
    printf("\n");
    printf("%lu",&n);
    printf("\n");

}

printf("Vetor original: ");
for(int i = 0; i < n; i++){
    printf("%.2lf ",vet_origem[i]);
}
printf("\n");

printf("Vetor copiado: ");
for(int i = 0; i < n; i++){
    printf("%.2lf ",vet_destino[i]);
}
printf("\n");

free(vet_origem);
free(vet_destino);

return 0;
}

```

/*Esta função não funciona porque o endereço do vet_destino não foi retornado pela função, portanto o programa não conseguirá imprimir o conteúdo do ponteiro vet_destino presente na int main.

Solução

Usar um ponteiro para ponteiro para retornar por referência o endereço de vet_destino.

*/