

Exercícios complementares

1. Escreva uma função que receba dois números inteiros retorne o menor número¹

```
#include <stdio.h>
int compara(int a, int b);

int main(){
    int n1, n2, res;
    printf("Digite dois valores separados por espaço:\n");
    scanf("%d %d", &n1, &n2);

    res = compara(n1, n2);
    printf("o menor número eh: %d\n", res);

    return 0;
}

int compara(int a, int b){
    if (a > b) {
        return b;
    }else{
        return a;
    }
}
```

2. Escreva uma função que calcule e retorne a distância entre dois pontos (x1, y1) e (x2, y2). Todos os números e valores de retorno devem ser do tipo float.

```
#include <stdio.h>
#include <math.h> // Biblioteca para a função sqrt (square root - raiz quadrada)

float calculaDistancia(float Px1, float Px2, float Py1, float Py2);

int main(){
    float x1, x2, y1, y2, res;
    printf("Digite as coordenadas X e Y do ponto 1:\n");
    scanf("%f %f", &x1, &y1);

    printf("Digite as coordenadas X e Y do ponto 2:\n");
    scanf("%f %f", &x2, &y2);

    res = calculaDistancia(x1, x2, y1, y2);
    printf("A distancia entre os pontos eh: %f\n", res);

    return 0;
}

float calculaDistancia(float Px1, float Px2, float Py1, float Py2){
    float a, b, c, d;
    a = Px1 - Px2; // Encontra a distancia entre os pontos x1 e x2
    b = Py1 - Py2; // Encontra a distancia entre os pontos y1 e y2
    c = a * a + b * b; // faz a conta = a^2 + b^2
    d = sqrt(c); // Acha a raiz da equação.
    return d; // Retorna o resultado.
}
```

3. Escreva uma função *potencia(base, expoente)* que, quando chamada, retorna $\text{base}^{\text{expoente}}$. Por exemplo, *potencia(3, 4)* deve retornar 81. Assuma que *expoente* é um inteiro maior ou igual a 1.

```
#include <stdio.h>

int potencia(int base, int expoente);

int main() {
    int b, e, res;
    printf("Digite a base: ");
    scanf("%d", &b);

    printf("Digite o expoente: ");
    scanf("%d", &e);

    res = potencia(b, e);
    printf("%d elevado %d eh igual a %d\n", b, e, res);

    return 0;
}

int potencia(int base, int expoente) {
    int tmp, i;
    tmp = 1; // atribui o primeiro valor para tmp

    while(i < expoente) {
        tmp *= base;
        i++;
    }
    return tmp;
}
```

4. Escreva um programa em C que leia 5 valores inteiros e imprima para cada um o seu correspondente valor absoluto. Para obter o valor absoluto do número utilize a função *Absoluto* especificada abaixo ²:

Nome: Absoluto

Descrição: Retorna o valor absoluto do número fornecido.

Entrada: int n

Saída: (int) O respectivo valor absoluto de n.

```
#include <stdio.h>

int absoluto(int n);

int main() {
    int numero, i;

    for(i = 0; i < 5; i++) {
        printf("digite o %d número: ", i+1);
        scanf("%d", &numero);
        printf("Valor absoluto: %d\n", absoluto(numero));
    }

    return 0;
}

int absoluto(int n) {
    if(n < 0) {
        return n*-1;
    }
    return n;
}
```

5. Escreva um programa que leia 5 números inteiros positivos (utilize uma função que leia esse número e verifique se ele é positivo). Para cada número informado escrever a soma de seus divisores (exceto ele mesmo). Utilize a função *SomaDivisores* para obter a soma.

Nome: SomaDivisores

Descrição: Calcula a soma dos divisores do número informado (exceto

ele mesmo).

Entrada: Um número inteiro e positivo.

Saída: A soma dos divisores.

Exemplo: Para o valor 8: $1+2+4 = 7$

```
#include <stdio.h>

int SomaDivisores(int n);

int main(){
    int numero, i;

    for(i = 0; i < 5; i++){
        printf("digite o %d número: ", i+1);
        scanf("%d", &numero);
        printf("\n\tSoma dos divisores: %d\n", SomaDivisores(numero));
    }

    return 0;
}

int SomaDivisores(int n){
    int i = 1, soma = 0;
    printf("\tdivisores de %d: ", n);
    while( i < n ){ // Laço de repetição que vai de 1 até o valor de N
        if (n % i == 0){ // O número N é divisível por i.
            soma += i; // Soma os divisores
            printf("%d, ", i );
        }
        i++;
    }

    return soma;
}
```

6. Faça uma função que recebe um valor inteiro e verifica se o valor é positivo, negativo ou zero. A função deve retornar 1 para valores positivos, -1 para negativos e 0 para o valor 0.
7. Faça um programa que realize a leitura dos seguintes dados relativos a um conjunto de alunos: Matrícula, Nome, Código da Disciplina, Nota1 e Nota2. Considere uma turma de ' até 10 alunos. Após ler todos os dados digitados, e depois de armazená-los em um vetor ' de estrutura, exibir na tela a listagem final dos alunos com as suas respectivas medias ' finais (use uma media ponderada: Nota1 com peso=1.0 e Nota2 com peso=2.0).
8. Utilizando uma estrutura, faça um programa que permita a entrada de nome, endereço e telefone de 5 pessoas e os imprima em ordem alfabética.
9. Faça um programa que leia os dados de 10 alunos (Nome, matricula, Media Final), armazenando em um vetor. Uma vez lidos os dados, divida estes dados em 2 novos vetores, o vetor dos aprovados e o vetor dos reprovados, considerando a media mínima para a aprovação como sendo 5.0. Exibir na tela os dados do vetor de aprovados, seguido dos dados do vetor de reprovados.
10. Escolha um jogo de cartas, baseado em um "baralho tradicional" (cada carta tem seu naipe e seu valor) ou tipo "super trunfo" (cada carta possui um conjunto de atributos). Implemente a parte de distribuição (sorteio) de cartas para 2 jogadores, considerando que cada jogador irá receber 5 cartas. Exiba na tela as cartas que cada um dos jogadores recebeu.
11. Peça ao usuário para digitar seus dados pessoais (Nome, Endereço, Data de

Nascimento, Cidade, CEP, email), verifique se as informações de Data de Nascimento, CEP e email fazem sentido, e mostre ao usuário as informações, se estão todas corretas, ou mostre que alguma informação estava errada.