

Algoritmos e Estruturas de Dados I (DCC/003)

Lista de Exercícios 06 – Função / Procedimento, Parâmetro, Vetor

Nome: Guilherme de Abreu Lima Buitrago Miranda

Matrícula: 2018054788

Exercício 1

```
#include <stdio.h>

int convHorMin (int hora, int minutos){
    return (hora * 60 + minutos);
}

int diferencaHoras (int horarios[]){
    int minutos1 = convHorMin(horarios[0], horarios[1]);
    int minutos2 = convHorMin(horarios[2], horarios[3]);

    return (minutos2 - minutos1);
}

int main (){

    int horarios[4];

    printf("Entre com os dois horários (horas e minutos)\n");
    scanf("%d %d %d %d", &horarios[0], &horarios[1], &horarios[2],
    &horarios[3]);

    printf("A diferenca entre as horas será de %d minutos \n",
    diferencaHoras(horarios));

    return 0;
}
```

Exercicio 2

```
#include <stdio.h>

void decrescente(int num1, int num2, int num3){

    int aux;

    if (num3 > num2){
        aux = num2;
        num2 = num3;
```

```

        num3 = aux;
    }

    if (num3 > num1){
        aux = num1;
        num1 = num3;
        num3 = aux;
    }

    if (num2 > num1){
        aux = num1;
        num1 = num2;
        num2 = aux;
    }

    printf("Ordem decrescente: %d %d %d \n", num1, num2, num3);
}

int main(){

    int num1, num2, num3, aux;
    printf("Entre com três números\n");
    scanf("%d %d %d", &num1, &num2, &num3);

    decrescente(num1, num2, num3);

    /*if (num3 > num2){
        aux = num2;
        num2 = num3;
        num3 = aux;
    }

    if (num3 > num1){
        aux = num1;
        num1 = num3;
        num3 = aux;
    }

    if (num2 > num1){
        aux = num1;
        num1 = num2;
        num2 = aux;
    }

```

```
printf("Ordem decrescente: %d %d %d \n", num1, num2, num3);*/
```

```
return 0;
```

```
}
```

Exercício 3

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

```
float mediaArit(int valor1, int valor2){
```

```
    return((valor1+valor2)/2);
```

```
}
```

```
float percentualFalt(int aulasTurma, int presenca){
```

```
    return(100-(((float)presenca/aulasTurma)*100));
```

```
}
```

```
void situacaoAluno(int mediaArit){
```

```
    if (mediaArit >= 60){
```

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

```
    }else{
```

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

```
    }
```

```
}
```

```
int main(){
```

```
    int aulasTurma;
```

```
    printf("Entre com o total de aulas da turma: \n");
```

```
    scanf("%d", &aulasTurma);
```

```
    int matricula, presenca;
```

```
    float prova1, prova2, trab1, trab2;
```

```
    printf("Entre com as seguintes informações do aluno: matrícula, nota da primeira  
prova, nota da segunda prova, nota do primeiro trabalho, nota do segundo trabalho e  
presença total\n");
```

```
    scanf("%d %f %f %f %f %d", &matricula, &prova1, &prova2, &trab1, &trab2,  
&presenca);
```

```
    printf("Matrícula: %d\n", matricula);
```

```
    printf("Média das provas: %f\n", mediaArit(prova1, prova2));
```

```
    printf("Média dos trabalhos: %f\n", mediaArit(trab1, trab2));
```

```
    printf("Média Final: %f \n", mediaArit(mediaArit(prova1, prova2),
```

```
mediaArit(trab1, trab2)));
```

```
    situacaoAluno(mediaArit(mediaArit(prova1, prova2), mediaArit(trab1, trab2)));
```

```
    printf("Percentual de faltas: %f %% \n", percentualFalt(aulasTurma, presenca));
```

```
}
```

Exercício 4 ~ 7

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

```
//Exercicio 4
```

```
int negativos (int tamanho, float vet []){  
    int neg = 0;  
  
    for (int i = 0; i < tamanho; ++i){  
        if (vet[i] < 0){  
            neg++;  
        }  
    }  
  
    return neg;  
}
```

```
//Exercicio 5
```

```
int pares (int n, int vet[]){  
    int numPares = 0;  
  
    for (int i = 0; i < n; ++i){  
        if (vet[i] % 2 == 0){  
            numPares++;  
        }  
    }  
  
    return numPares;  
}
```

```
//Exercicio 6
```

```
int maiores (int n, int vet[], int x){  
    int numMaiores = 0;  
  
    for (int i = 0; i < n; ++i){  
        if (vet[i] > x){  
            numMaiores++;  
        }  
    }  
  
    return numMaiores;  
}
```

```
//Exercicio 7
```

```
int * somatorios (int n, int v[], int w[]){  
    w[0] = v[0];  
    int i = 1;
```

```

    for (i; i < n; i++){
        w[i] = v[i] + w[i - 1];
    }

    return w;
}

int main(){
    return 0;
}

```

Exercício 8

```

#include <stdio.h>
#include <math.h>

int raizes(double a, double b, double c, double *raiz1, double *raiz2){
    //delta = 4ac
    //bhaskara = (b*b +- raiz(delta))/2a

    float delta = (4*a*c);

    if (delta < 0){
        return 0;
    }else{
        if (delta == 0){
            *raiz1 = ((b*b)+sqrt(delta))/2*a;
            *raiz2 = ((b*b)+sqrt(delta))/2*a;
            return 1;
        }else{
            *raiz1 = ((b*b)+sqrt(delta))/2*a;
            *raiz2 = ((b*b)-sqrt(delta))/2*a;
            return 2;
        }
    }
}

int main(){

    double a, b, c;
    double raiz1 = 0;
    double raiz2 = 0;

    scanf("%lf %lf %lf", &a, &b, &c);
    int numRaizes = raizes(a, b, c, &raiz1, &raiz2);
    printf("%lf %lf %lf %lf %lf %d \n", a, b, c, raiz1, raiz2, numRaizes);

    return 0;
}

```