

Roteiro 1 - Laboratório de Programação 2

1.1 -

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

int negativos(float *vet, int N)
{
    int contador = 0;
    for (int i = 0; i < N; i++)
    {
        if (vet[i] < 0)
            contador++;
    }
    return contador;
}

int main()
{
    int N;

    printf("Informe o tamanho do vetor: ");
    scanf("%d", &N);
    float *vetor;
    vetor = malloc(N * sizeof(float));

    printf("Informe os valores:");
    for (int i = 0; i < N; i++)
    {
        scanf("%f", &vetor[i]);
    }

    int resultado = negativos(vetor, N);
    printf("Quantidade de Numeros Negativos no Vetor: %d", resultado);
    return 0;
}
```

Saída:

```
Informe o tamanho do vetor: 10
Informe os valores:-2
-3
-4
5
6
7
-9
1
2
3
Quantidade de Numeros Negativos no Vetor: 4
```

1.2 -

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

int maior_elemento(int *vet, int tamanho){
    int maior = vet[0];
    for (int i = 0; i < tamanho; i++){
        if (vet[i] > maior)
            maior = vet[i];
    }
    return maior;
}

int menor_elemento(int *vet, int tamanho){
    int menor = vet[0];
    for (int i = 0; i < tamanho; i++){
        if (vet[i] < menor)
            menor = vet[i];
    }
    return menor;
}

float media(int *vet, int tamanho){
    float media = 0;
    for (int i = 0; i < tamanho; i++){
        media += vet[i];
    }
    return media / tamanho;
}

int main(){
    srand(time(NULL));
    int vetor1[10];
    int vetor2[20];
```

```

for (int i = 0; i < 10; i++){
    vetor1[i] = rand() % 100;
}

for (int i = 0; i < 20; i++){
    vetor2[i] = rand() % 100;
}

printf("Vetor 1:");
for (int i = 0; i < 10; i++){
    printf("%d ", vetor1[i]);
}

printf("\nVetor 2:");
for (int i = 0; i < 20; i++){
    printf("%d ", vetor2[i]);
}
printf("\n");

printf("Maior Elemento Vetor 1: %d\n", maior_elemento(vetor1, 10));
printf("Menor Elemento Vetor 1: %d\n", menor_elemento(vetor1, 10));
printf("Media Elementos Vetor 1: %f\n", media(vetor1, 10));

printf("Maior Elemento Vetor 2: %d\n", maior_elemento(vetor2, 20));
printf("Menor Elemento Vetor 2: %d\n", menor_elemento(vetor2, 20));
printf("Media Elementos Vetor 2: %f\n", media(vetor2, 20));
}

```

Saída:

```

Vetor 1:72 42 96 45 33 35 32 23 41 9
Vetor 2:18 64 8 67 6 18 46 37 65 60 31 69 71 96 82 97 16 58 49 40
Maior Elemento Vetor 1: 96
Menor Elemento Vetor 1: 9
Media Elementos Vetor 1: 42.800000
Maior Elemento Vetor 2: 97
Menor Elemento Vetor 2: 6
Media Elementos Vetor 2: 49.900000

```

1.3 -

```

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

typedef struct Aluno{
    char nome[100];
    int matricula;
    float nota;
} Aluno;

```

```

void imprimir_infos(Aluno *alunos, int tamanho){
    float maior_nota = alunos[0].nota;
    float menor_nota = alunos[0].nota;
    int indiceMaior = 0;
    int indiceMenor = 0;

    for (int i = 0; i < tamanho; i++){
        if (alunos[i].nota > maior_nota){
            maior_nota = alunos[i].nota;
            indiceMaior = i;
        }
        if (alunos[i].nota < menor_nota){
            menor_nota = alunos[i].nota;
            indiceMenor = i;
        }
    }

    printf("Aluno com Maior Nota:\n");
    printf("Nome: %s\n", alunos[indiceMaior].nome);
    printf("Matricula: %d\n", alunos[indiceMaior].matricula);
    printf("Nota: %.2f\n", maior_nota);

    printf("Aluno com Menor Nota:\n");
    printf("Nome: %s\n", alunos[indiceMenor].nome);
    printf("Matricula: %d\n", alunos[indiceMenor].matricula);
    printf("Nota: %.2f\n", menor_nota);
}

int main(){
    int numeroAlunos;
    printf("Informe o numero de alunos: ");
    scanf("%d", &numeroAlunos);

    Aluno *alunos = malloc(numeroAlunos * sizeof(Aluno));
    for (int i = 0; i < numeroAlunos; i++){
        printf("Informe o nome do aluno %d: ", i + 1);
        scanf("%s", alunos[i].nome);

        printf("Informe a matricula do aluno %d: ", i + 1);
        scanf("%d", &alunos[i].matricula);

        printf("Informe a nota do aluno %d: ", i + 1);
        scanf("%f", &alunos[i].nota);
    }

    imprimir_infos(alunos, numeroAlunos);
    return 0;
}

```

```
}
```

Saída:

```
Aluno com Maior Nota:  
Nome: Gabriel  
Matricula: 1  
Nota: 10.00  
Aluno com Menor Nota:  
Nome: Julio  
Matricula: 3  
Nota: 6.00
```

1.4 -

```
#include <stdio.h>  
#include <stdlib.h>  
#include <math.h>  
  
int raizes(float A, float B, float C, float *X1, float *X2)  
{  
    float delta = pow(B, 2) - 4 * A * C;  
  
    if (delta < 0)  
        return 0;  
    if (delta == 0)  
    {  
        *X1 = -B / (2 * A);  
        return 1;  
    }  
  
    *X1 = (-B + sqrt(delta)) / (2 * A);  
    *X2 = (-B - sqrt(delta)) / (2 * A);  
  
    return 2;  
}  
  
int main()  
{  
    float a, b, c, x1, x2;  
    printf("Informe os coeficientes do polinomio: ");  
    scanf("%f %f %f", &a, &b, &c);  
  
    switch (raizes(a, b, c, &x1, &x2))  
    {  
        case 1:  
            printf("Uma raiz real\n");  
        case 2:  
            printf("Duas raizes reais\n");  
        case 0:  
            printf("Nenhuma raiz real\n");  
    }  
}
```

```

        printf("X1 = %.2f\n", x1);
        break;
    case 2:
        printf("Duas raizes reais\n");
        printf("X1 = %.2f\nX2 = %.2f\n", x1, x2);
        break;
    default:
        printf("Nenhuma raiz real\n");
        break;
}

printf("\n");

return 0;
}

```

Saída:

```

Informe os coeficientes do polinomio: 4
3
5
Nenhuma raiz real

```

2.1 -

```

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

void printarCrescente(int n){
    if (n <= 5){
        printf ("%d\n", n);
        printarCrescente(n+1);
    }
}

void printarDecrescente(int n){
    if (n >= 1){
        printf("%d\n", n);
        printarDecrescente(n-1);
    }
}

int main (){
    printf ("Ordem Crescente:\n");
    printarCrescente(1);
}

```

```

printf ("\n");

printf ("Ordem Decrescente:\n");
printarDecrescente(5);
printf ("\n");

return 0;
}

```

Saída:

```

Ordem Crescente:
1
2
3
4
5

Ordem Decrescente:
5
4
3
2
1

```

2.2 -

```

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

void imprimir_decrescente(char *vetor, int tamanho)
{
    if (tamanho > 0)
    {
        printf("%c\n", vetor[tamanho - 1]);
        imprimir_decrescente(vetor, tamanho - 1);
    }
}

int main()
{
    imprimir_decrescente("ABCDE", 7);
    return 0;
}

```

Saída:

```
PS C:\Users\USER\OneDrive\Área de Trabalho\LabProg2\Lista 1\output> & .\'exercicio2b.exe'
```

```
E  
D  
C  
B  
A
```

2.3 -

```
#include <stdio.h>

int soma(int atual, int maior, int menor)
{
    if (atual < menor)
        return 0;
    return atual + soma(atual - 1, maior, menor);
}

int main()
{
    int menor, maior;
    printf("Menor numero do intervalo:");
    scanf("%d", &menor);
    printf("Maior numero do intervalo:");
    scanf("%d", &maior);
    printf("Somatorio = %d\n", soma(maior, maior, menor));

    return 0;
}
```

Saída:

```
Menor numero do intervalo:1
Maior numero do intervalo:5
Somatorio = 15
```

2.4 -

```
#include <stdio.h>

int multiplica_rekursiva (int n, int m){
    if (n == 0){
        return 0;
    }
    return m + multiplica_rekursiva (n - 1, m);
}

int main(){
    int n, m;
    printf ("Primeiro Numero: ");
```



```

scanf ("%d", &n);
printf ("Segundo Numero: ");
scanf ("%d", &m);

printf ("Resultado: %d", multiplica_rekursiva (n, m));
return 0;
}

```

Saída:

```

Primeiro Numero: 3
Segundo Numero: 5
Resultado: 15

```

2.5 -

```

#include <stdio.h>

void preenche_rekursivo(int i, int n, int* vetor){
    if (i >= n)
        return;

    vetor[i] = i;
    preenche_rekursivo(i+1, n, vetor);
}

void imprime_crescente(int i, int n, int* vetor){
    if (i >= n)
        return;

    printf("%d ", vetor[i]);
    imprime_crescente(i+1, n, vetor);
}

void imprime_decrescente(int i, int n, int* vetor){
    if (i >= n)
        return;

    printf("%d ", vetor[n-1-i]);
    imprime_decrescente(i+1, n, vetor);
}

int maiorElemento_rekursivo(int i, int n, int* vetor){
    int proximo;

```

```

        if (i < n-1){
            proximo = maiorElemento_recursivo(i+1, n, vetor);
        } else {
            proximo = vetor[i];
        }

        if (vetor[i] > proximo){
            return vetor[i];
        } else {
            return proximo;
        }
    }
}

int main(){
    int N = 10;
    int vetor[N];
    preenche_recursivo(0, N, vetor);
    imprime_crescente(0, N, vetor);
    printf ("\n");
    imprime_decrescente(0, N, vetor);
    printf ("\n");
    printf ("Maior Elemento: %d\n", maiorElemento_recursivo(0, N,
vetor));
}

```

Saída:

```

0 1 2 3 4 5 6 7 8 9
9 8 7 6 5 4 3 2 1 0
Maior Elemento: 9

```

2.6-

```

#include <stdio.h>

void hanoi_recursivo(int n, char origem, char destino, char trabalho)
{
    if (n == 1)
    {
        printf("Mover disco %d de %c para %c\n", n, origem, destino);
    }

    if (n > 1)
    {
        // Mover o n-1 disco de A para B, usando o C de trabalho.
    }
}

```

```

        hanoi_recursivo(n - 1, origem, trabalho, destino);
        // Mover os discos de A para C.
        printf("Mover disco %d de %c para %c\n", n, origem, destino);
        // Mover o n-1 disco de B para C, usando o a de trabalho.
        hanoi_recursivo(n - 1, trabalho, destino, origem);
    }
}

int main()
{
    int n;
    printf("Informe o numero de discos: ");
    scanf("%d", &n);
    hanoi_recursivo(n, 'A', 'C', 'B');
    printf("\n");
    return 0;
}

```

Saída:

```

Informe o numero de discos: 3
Mover disco 1 de A para C
Mover disco 2 de A para B
Mover disco 1 de C para B
Mover disco 3 de A para C
Mover disco 1 de B para A
Mover disco 2 de B para C
Mover disco 1 de A para C

```