

# Lista de Exercícios 3 – Algoritmos e Estruturas de Dados I

## Prof. Rafael Fernandes Lopes

### Algoritmos Básicos com Vetores

**Aluno: ADRIANO DOS SANTOS ELIZEU**

1 Faça um programa que pede para o usuário informar o nome e a idade de 50 pessoas. Ao final disso, o programa deve exibir o nome de todas as pessoas menores de 18 anos e depois de todas as pessoas maiores de 18 anos (inclusive com 18 anos).

```
#include <stdio.h>
int main(){

    char nome[50][80];
    int i, idade[50];

    for (i = 0; i < 50; i++){
        printf("NOME: ");
        scanf("%s", nome[i]);
        printf("IDADE: ");
        scanf("%i", &idade[i]);
    }
    printf("\n ----- MENOR DE 18 ANOS ----- \n");
    for (i = 0; i < 50; i++){
        if(idade[i] < 18){
            printf("%s\n", nome[i]);
        }
    }
    printf("\n ----- MAIOR DE 18 ANOS ----- \n");
    for (i = 0; i < 50; i++){
        if(idade[i] >= 18){
            printf("%s\n", nome[i]);
        }
    }
    return 0;
}
```

2 Implemente um programa que efetue a busca de um inteiro N em um vetor com 100 inteiros e, caso encontre N, informe qual a sua posição no vetor

```
#include <stdio.h>
#define lim 100
int main(){
```

```

int i,j,n;
int v[lim];

for(i = 1; i< 100; i++)
{
    printf("Elementos %i: ", i);
    scanf("%i", &v[i]);

}
printf("\nDigite o numero: ");
scanf("%i", &n);
for (i = 0; i < lim; i++)
{
    if (n==v[i])
    {
        printf("\nNumero %i na posição %i",n,i);
    }
}
return 0;
}

```

3 Faça um programa que lê 100 números reais e informa qual é o segundo maior deles.

```

#include <stdio.h>
#define LIM 100
int main (){
float num[LIM], maior, maior2;
int i;
for(i = 0; i < LIM ; i++){
    printf("Entre com número real: ");
    scanf("%f", &num[i]);
}
maior = num[0];
maior2 = maior;
for(i = 0; i < LIM ; i++){
    if (num[i] > maior){
        maior = num[i];
        maior2 = maior;
    }
}
printf("\n:::: SEGUNDO MAIOR NÚMERO É: %.1f", maior2);
return 0;
}

```

4 Faça um programa que informa a quantidade de números repetidos que existem em um vetor com 100 números inteiros ordenados de forma crescente. Assuma que um mesmo inteiro não aparece três ou mais vezes no vetor.

```
#include <stdio.h>
#define LIM 100

int main (){
    int n[LIM], i, j, ordem, cont =0;
    for(i = 0; i < LIM ; i++){
        printf("Entre com número: ");
        scanf("%d", &n[i]);
    }
    for(i = 0; i < LIM ; i++){
        for(j = i + 1 ; j <LIM; j++ ){
            if (n[i]>n[j]){
                ordem = n[i];
                n[i] = n[j];
                n[j] = ordem;
            }
        }
    }
    for(i = 0; i < LIM ; i++){
        for(j = i + 1 ; j <LIM; j++ ){
            if(n[i]==n[j]){
                cont = cont + 1;
                break;
            }
        }
    }

    printf("\n");
    for (i = 0; i < LIM; i++){
        printf("%d\n", n[i]);
    }
    printf("\nQuantidade de números repitidos: %d", cont);
    return 0;
}
```

5 Some os números repetidos de um vetor com 100 inteiros. Como exemplo, o trecho do vetor {2,2,2,3, ...} deve dar como resposta 6.

```
#include <stdio.h>
#include <stdlib.h>
#define N 10
int main(){
```

```

int vetor[N], rep[N];
int i, j, soma = 0;
for(i = 0; i < N; i++){
    printf("Digite um número: ");
    scanf("%d", &vetor[i]);
}

for(int i = 0; i < N; i++){
    for(int j = i + 1; j < N; j++){
        if(vetor[i] == vetor[j]){
            rep[i] = vetor[i];
            soma += rep[i];
            break;
        }
    }
}
printf("\nSOMA: %d", soma);

return 0;
}

```

6 Dados três vetores A, B e C com 50 números reais ordenados (de forma crescente), faça um programa que constrói um quarto vetor D ordenado com esses 150 números reais

```

#include <stdio.h>
#define LIM 50
int main(){
    float A[LIM], B[LIM], C[LIM], D[150];
    int i,j,k,x,y, ordem;

    for(i = 0; i < LIM; i++){
        printf("Vet A[%i]: ", i+1);
        scanf("%f", &A[i]);
    }
    for(i = 0; i < LIM; i++){
        printf("Vet B[%i]: ", i+1);
        scanf("%f", &B[i]);
    }
    for(i = 0; i < LIM; i++){
        printf("Vet C[%i]: ", i+1);
        scanf("%f", &C[i]);
    }

    for(i = 0; i < LIM; i++){
        D[i] = A[i];
        for(j = 0; j < LIM; j++){

```

```

    D[j+50] = B[j];
    for(k = 0; k < LIM; k++) {
        D[k+101] = C[k];
    }
}
}
for(x = 0; x < 150; x++)
    for(y = x + 1; y < 150; y++)
        if ( D[x] > D[y]){
            ordem = D[x];
            D[x] = D[y];
            D[y] = ordem;
        }

printf("\n");
for (i = 0; i < 150 ; i++){
    printf("%.1f\n", D[i]);
}
return 0;
}

```

7 Receber do usuário uma lista de N nomes e idades de pessoas, onde N também é fornecido pelo usuário, e mostrar o nome e a idade da pessoa mais idosa e da pessoa mais jovem.

```

#include <stdio.h>
struct pessoa{
    char nome[50];
    int idade;
};

int main(){
    int n, i, idade_maior = 0, idade_menor = 99;

    do {
        printf("QUANTIDADE DE PESSOAS: ");
        scanf ("%d", &n);
    }
    while (n<0);
    printf("\n");

    struct pessoa pessoas[n];
    for (i = 0; i < n; i++){
        printf("NOME DA %i PESSOA: ", i+1);
        scanf("%s", pessoas[i].nome);
        printf("IDADE: ");
        scanf("%i", &pessoas[i].idade);

        if (pessoas[i].idade > idade_maior){

```

```
        idade_maior = pessoas[i].idade;
    }
    if (pessoas[i].idade < idade_menor){
        idade_menor = pessoas[i].idade;
    }
}
printf("\n");
printf("Pessoa mais jovem:\n");
for (i = 0; i < n; i++){
    if(pessoas[i].idade == idade_menor){
        printf("NOME: %s      IDADE: %d", pessoas[i].nome,
pessoas[i].idade);
    }
}
printf("\n\n");
printf("Pessoa mais idosa:\n");
for (i = 0; i < n; i++){
    if(pessoas[i].idade == idade_maior){
        printf("NOME: %s      IDADE: %d", pessoas[i].nome,
pessoas[i].idade);
    }
}
return 0;
}
```