

**Universidade Positivo****Disciplina: Algoritmos de Programação****Referência: Profa. Fernanda Hemberger e Prof. Diogo Deconto****Professor: Kristian Capeline****Exercícios****• Vetores**

1. Realize o teste de mesa dos algoritmos a seguir, indique o valor das variáveis e o que será apresentado na tela:

**ALGORITMO Vetor1**

VAR

INTEIRO: i, x, y, num[10];

INICIO

x ← 1;

y ← 2;

num[x] ← y\*y;

num[y] ← x;

num[x+y] ← num[x];

num[6] ← num[1]+num[3];

num[num[x]] ← num[6]-num[y];

num[x+y] ← x\*y;

num[3+4] ← num[6]\*2;

num[x-1] ← x+y;

num[num[x]+1] ← num[x]+num[y];

num[num[6]+1] ← num[6+1];

num[4\*2] ← num[9]-num[7]+x;

PARA (i ← 0; i &lt; 10; i ← i + 1) FAÇA

ESCREVA (num[i]);

FIM\_PARA

FIM.

Linha	num[10]	i	x	y	print
1					
2					
3					
4					
5			1		
6				2	
7	num[1] = 4				
8	num[2] = 1				
9	num[3] = 4				
10	num[6] = 8				
11	num[4] = 7				
12	num[3] = 2				
13	num[7] = 16				
14	num[0] = 3				
15	num[5] = 5				
16	num[9] = 16				
17	num[8] = 1				
18					
19					3
20					2
21					1

22					4
23					7
24					5
25					8
26					16
27					1
28					16
29					
30					

# **ALGORITMO Vetor2**

VAR

INTEIRO: i, x, y, num[10] = {3,4,6,2,1,6,8,2,9,5};

INICIO

x←2;

y←3;

num[x]←num[x\*y];

num[num[x]]←num[x+y];

num[y+1]←x+y;

num[y-x]←num[num[y]];

num[y]←num[y]+num[x];

num[x-2]←num[y]+x;

PARA(i ← 0; i < 10; i ← i + 1) FAÇA

ESCREVA (num[i]);

FIM\_PARA

FIM.

Linha	num[10]	i	x	y	print
1					
2					
3	{3,4,6,2,1,6,8,2,9,5}				
4					
5			2		
6				3	
7	num[2] = 8				
8	num[8] = 6				
9	num[4] = 5				
10	num[1] = 6				
11	num[3] = 8				
12	num[0] = 4				
13					
14					4
15					6
16					8
17					8
18					5
19					6
20					8
21					2
22					6
23					5
24					
25					

**2. Faça um Programa que leia um vetor de 5 números inteiros e mostre-os.**

```
//Exerc 2

#include <stdio.h>

int main()
{
    int v[5];

    printf("Digite 5 valores: ");
    for(int i = 0; i < 5; i++){
        scanf("%i", &v[i]);
    }

    for(int i = 0; i < 5; i++){
        printf("%i \n", v[i]);
    }
}
```

**3. Faça um Programa que leia um vetor de 10 números reais e mostre-os na ordem inversa.**

```
//Exerc 3

#include <stdio.h>

int main()
{
    float v[10];

    printf("Digite 10 valores: ");
    for(int i = 0; i < 10; i++){
        scanf("%f", &v[i]);
    }

    for(int i = 0; i < 10; i++){
        printf("%f \n", v[9 - i]);
    }
}
```

4. Faça um Programa que leia 4 notas, armazene-as em um vetor, mostre as notas e a média na tela.

```
//Exerc 4

#include <stdio.h>

int main()
{
    float v[4];

    for(int i = 0; i < 4; i++){
        printf("Digite a nota %i: ", i+1);
        scanf("%f", &v[i]);
    }

    for(int i = 0; i < 10; i++){
        printf("%f \n", v[9 - i]);
    }

    printf("Média: %f", (v[0]+v[1]+v[2]+v[3])/4);
}
```

5. Faça um Programa que leia 20 números inteiros e armazene-os num vetor. Armazene os números pares no vetor PAR e os números IMPARES no vetor ímpar. Imprima os três vetores.

```

//Exerc 5

#include <stdio.h>

int print(int array[20]);

int main()
{
    int v[20];
    int p[20] = {0};
    int i[20] = {0};

    printf("Digite 20 números: ");

    for(int c = 0; c<20; c++)
    {
        scanf("%i", &v[c]);
        if(v[c] % 2 == 0){
            p[c] = v[c];
        }
        else{
            i[c] = v[c];
        }
    }

    printf("Números: \n");
    print(v);

    printf("Números pares: \n");
    print(p);

    printf("Números ímpares: \n");
    print(i);
}

int print(int array[20]){
    for(int i=0;i<20;i++){
        if(array[i] != 0){
            printf("%i \n", array[i]);
        }
    }
}

```

6. Construa um algoritmo que leia um vetor de 15 posições e calcule a média destes valores. Na sequência, apresente na tela os valores que são iguais ou superiores à média.

```

//Exerc 6

#include <stdio.h>

int main()
{
    int v[15];
    float media;

    printf("Digite 15 posições: ");

    for(int c = 0; c<15; c++)
    {
        scanf("%i", &v[c]);
        media += v[c];
    }

    media /= 15;

    printf("Valores maiores ou iguais à média: \n");

    for(int i = 0; i < 15; i++){
        if(v[i] >= media){
            printf("%i \n", v[i]);
        }
    }

}

```

7. Faça um algoritmo que leia 20 números do teclado e os armazene em um vetor. Crie um segundo vetor de 20 posições que armazene o dobro de cada um dos números do primeiro vetor e apresente este vetor.

```
//Exerc 7

#include <stdio.h>

int main()
{
    int v[20];
    int d[20];

    printf("Digite 20 números: ");

    for(int c = 0; c<20; c++)
    {
        scanf("%i", &v[c]);
        d[c] = v[c] * 2;
    }

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

}
```

8. Elabore um algoritmo que leia 50 números do teclado e os armazene em um vetor. Em seguida, leia um número n qualquer e apresente na tela a quantidade de vezes que o número n aparece no vetor.

```
//Exerc 8

#include <stdio.h>

int main()
{
    int v[50];
    int n;
    int count = 0;

    printf("Digite o número n: ");
    scanf("%i", &n);

    printf("Digite 50 números: ");

    for(int c = 0; c<50; c++)
    {
        scanf("%i", &v[c]);
        if(v[c] == n){
            count++;
        }
    }

    printf("%i", count);

}
```

9. Crie um algoritmo que leia o vetor com 1000 números e some apenas os elementos positivos que estão nas posições pares.

```
//Exerc 9

#include <stdio.h>

int main()
{
    int v[1000];
    int s;

    printf("Digite um vetor de 1000 digitos: ");

    for(int c = 0; c<1000; c++)
    {
        scanf("%i", &v[c]);
        if(v[c] > 0 && (c%2 == 0)){
            s += v[c];
        }
    }

    printf("Soma: %i", s);

}
```

10. Faça um algoritmo que leia um vetor A de 30 números. Armazene em um vetor B o maior elemento de A, o menor elemento de A, a soma dos elementos de A e a média dos elementos de A. Apresente o vetor B na tela.



```

//Exerc 10

#include <stdio.h>

int main()
{
    int a[30];
    float b[4];
    int maior;
    int menor;
    int soma;
    float media;

    printf("Digite 30 números: ");

    for(int c = 0; c<30; c++)
    {
        scanf("%i", &a[c]);
        if(c == 0){
            b[0] = a[c];
            b[1] = a[c];
        }
        else{
            if(a[c] > b[0]){
                b[0] = a[c];
            }
            else if(a[c] < b[1]){
                b[1] = a[c];
            }
        }
        b[2] += a[c];
    }

    b[3] = b[2]/30;

    for(int i = 0; i<4; i++){
        printf("%f \n", b[i]);
    }
}

```

- 11.** Crie um algoritmo que leia um vetor A de 10 posições do teclado e, ao final da leitura, copie os elementos de A em B de forma invertida. Ou seja, o primeiro elemento de A é o último elemento de B, o segundo elemento de A é o penúltimo elemento de B, e assim por diante.

```
//Exerc 11

#include <stdio.h>

int main()
{
    int a[10];
    int b[10];

    printf("Digite 10 números: ");

    for(int c = 0; c<10; c++)
    {
        scanf("%i", &a[c]);
    }

    for(int c = 0; c<10; c++)
    {
        b[9 - c] = a[c];
    }

    for(int c = 0; c<10; c++)
    {
        printf("%i ", b[c]);
    }
}
```

- 12.** Elabore um algoritmo que calcule o produto escalar entre dois vetores de inteiros de tamanho igual a 5. Esses números devem ser digitados pelo teclado.

Exemplo:  $\{0, 2, 4, 6, 8\} \bullet \{1, 3, 5, 7, 9\} = 0*1 + 2*3 + 4*5 + 6*7 + 8*9 = 140$

```
//Exerc 12

#include <stdio.h>

int main()
{
    int a[5];
    int b[5];
    int v = 0;

    printf("Digite 5 números: ");
    for(int c = 0; c<5; c++)
    {
        scanf("%i", &a[c]);
    }

    printf("Digite mais 5 números: ");
    for(int c = 0; c<5; c++)
    {
        scanf("%i", &b[c]);
    }

    for(int c = 0; c<5; c++)
    {
        v += a[c]*b[c];
    }
    printf("Resultado: %i", v);
}
```

- 13.** Elabore um algoritmo que leia 30 números do teclado e preencha um vetor de acordo com a seguinte regra: com exceção do 1º número, só é permitido armazenar um número se ele for maior que o anterior.

Ex.: se o primeiro valor lido for 5, o próximo valor lido só poderá ser maior que 5. E assim por diante.

```
//Exerc 13

#include <stdio.h>

int main()
{
    int a[30] = {0};
    int n;
    printf("Digite 30 números: ");

    for(int c = 0; c<30; c++)
    {
        if(c == 0){
            scanf("%i", &a[c]);
        }
        else{
            scanf("%i", &n);
            if(n > a[c-1]){
                scanf("%i", &a[c]);
            }
        }
    }

    for(int c = 0; c<30; c++)
    {
        printf("%i \n", a[c]);
    }
}
```

- 14.** Crie um algoritmo com uma função que receba uma string como parâmetro e devolva outra string com os caracteres embaralhados.

Por exemplo: se a função receber a palavra “python”, pode retornar npthyo, ophtyn ou qualquer outra combinação possível, de forma aleatória. Padronize em sua função que todos os caracteres serão devolvidos em caixa alta, independentemente de como foram digitados.

```
//Exerc 14

#include <stdio.h>

char* embar(char p[30], int size);

char palavra[30];

int main()
{
    int csize = 0;

    printf("Digite uma palavra: ");
    scanf("%s", palavra);

    for(int i=0;i<30;i++){
        if(palavra[i] != '\0'){
            csize++;
        }
    }

    printf("%s", embar(palavra, csize));
}

char* embar(char p[30], int size){

    char newp[size];

    for(int i=0; i<size; i++){
        newp[size-i] = p[i];
    }

    for(int i=0; i<size; i++){
        palavra[i] = newp[i + 1] - 32;
    }
    return palavra;
}
```

- 15.** Em uma competição de salto em distância cada atleta tem direito a cinco saltos. O salto mais baixo e mais alto do atleta deve ser desconsiderado, o resultado do atleta será determinado pela média de apenas três valores restantes. Você deve fazer um programa que receba o nome e as cinco distâncias alcançadas pelo atleta em seus saltos e depois informe o nome, os saltos e a média dos saltos. O programa deve ser encerrado quando não houverem mais atletas. Exemplo de como deve ser a saída para cada atleta lido:

Atleta: Diogo Deconto

Primeiro Salto: 6.5 m - DESCONSIDERADO

Segundo Salto: 6.1 m

Terceiro Salto: 6.2 m

Quarto Salto: 5.4 m

Quinto Salto: 5.3 m - DESCONSIDERADO

Resultado final:

Atleta: Diogo Deconto

Saltos considerados: 6.1 - 6.2 - 5.4

Média dos saltos: 5.9 m

//Exerc 15

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

```
#include <stdbool.h>
```

```
void stats(char nome[50], float salto[5]);
```

```
int main()
```

```
{
```

```
    char nome[50];
```

```
    float saltos[5];
```

```
    bool calc = true;
```

```
    int resp;
```

```
    while(calc){
```

```
        /*Nesse caso o código só funciona usando somente o primeiro nome,
        pois o "scanf" não lê os "space", e parece que pra ler, teria de usar algo
        assim "%[^\\n]s" que parece apontar pra algum outro lugar na segunda
        tentativa, dando erro também... */
```

```
        printf("Digite o nome do atleta: ");
```

```
        scanf("%s", nome);
```

```
        printf("Digite o valor dos 5 saltos: ");
```

```
        scanf("%f %f %f %f %f", &saltos[0], &saltos[1], &saltos[2],
&saltos[3], &saltos[4]);
```

```
        stats(nome, saltos);
```

```
        calc = false;
```

```
        printf("\\n\\nDeseja adicionar outro atleta? (1-SIM | 2-NÃO) \\n");
```

```
        scanf("%i", &resp);
```

```
        if(resp == 1){
```

```
            calc = true;
```

```
        }
```

```
        else{
```

```
            calc = false;
```

```
        }
```

```
    }
```

```
}
```

```

void estats(char nome[50], float salto[5]){

    float maior = salto[0];
    float menor = salto[0];
    float media = 0;
    int mmIndex[2];

    for(int i=0; i<5; i++){
        if(salto[i] > maior){
            maior = salto[i];
            mmIndex[0] = i;
        }
        if(salto[i] < menor){
            menor = salto[i];
            mmIndex[1] = i;
        }
    }
    printf("Atleta: %s \n\n", nome);
    for(int i=0;i<5;i++){
        if(salto[i] != maior && salto[i] != menor){
            media += salto[i];
            printf("%i° salto: %f m \n", i+1, salto[i]);
        }
        else{
            printf("%i° salto: %f m - DESCONSIDERADO \n", i+1, salto[i]);
        }
    }
    printf("\nResultado final: \nAtleta: %s \nSaltos considerados: ", nome);
    for(int i=0;i<5;i++){
        if(salto[i] != maior && salto[i] != menor){
            printf("- %f ", salto[i]);
        }
    }
    media /= 3;
    printf("\nMédia dos saltos: %f m", media);
}

```