

Gabarito - Lista de Exercícios 5 – Vetores e Matrizes

Parte 1: Vetores

- 1) O que será impresso na tela decorrente da execução do seguinte código?

```
1  #include<stdio.h>
2  int main() {
3      int vetorA[10], vetorB[10], i;
4      for(i = 0; i < 10; i++){
5          vetorA[i] = 10;
6          vetorB[i] = i+1;
7      }
8      for(i = 0; i < 10; i++){
9          printf("%d ", vetorA[i]+vetorB[i]);
10     }
11     return 0;
12 }
```

11 12 13 14 15 16 17 18 19 20

- 2) Escreva um programa que leia um vetor de 12 posições e em seguida leia também dois valores X e Y quaisquer correspondentes a duas posições no vetor. Ao final seu programa deverá imprimir a soma dos valores encontrados nas respectivas posições X e Y.

```
#include<stdio.h>
int main() {
    int vet[12], x, y, soma, i;
    printf("\n Entre com os 12 valores: \n");
    for(i=0;i<12;i++){
        scanf("%d",&vet[i]);
    }
    do{
        printf("Entre com o valor de x: ");
        scanf("%d",&x);
        printf("Entre com o valor de y: ");
        scanf("%d",&y);
        if(x<0 || y<0 || x>11 || y>11) printf("\n INDICES INVALIDOS! TENTE
NOVAMENTE \n");
    }while(x<0 || y<0 || x>11 || y>11);
    soma = vet[x]+vet[y];
    printf("\n %d + %d = %d",vet[x],vet[y],soma);
    return 0;
}
```

- 3) Escreva um programa que declare um vetor de 100 posições e o preencha com os 100 primeiros números ímpares e o imprima.

```
#include<stdio.h>
int main() {
    int vet[100], i;
    for(i=0;i<100;i++){
```

```

        vet[i]=2*i+1;
    }
    for(i=0;i<100;i++){
        printf(" %d ", vet[i]);
    }
    return 0;
}

```

- 4) Faça um programa que leia um vetor de 16 posições e troque os 8 primeiros valores pelos 8 últimos e vice-versa. Escreva ao final o vetor obtido.

```

#include<stdio.h>
int main(){
    int vet[16], i, aux;
    printf("\n Entre com os 16 valores:\n");
    for(i=0; i<16; i++){
        scanf("%d", &vet[i]);
    }
    for(i=0;i<8;i++){
        aux = vet[i];
        vet[i] = vet[i+8];
        vet[i+8] = aux;
    }
    printf("\nO vetor invertido:");
    for(i=0;i<16;i++){
        printf("\n vet[%d]=%d",i,vet[i]);
    }
    return 0;
}

```

- 5) Faça um programa que leia um vetor inteiro de 20 posições e em seguida um valor X qualquer (inteiro). Seu programa deverá fazer uma busca do valor de X no vetor lido e informar a posição em que foi encontrado ou se não foi encontrado.

```

#include<stdio.h>
int main(){
    int vet[20], i, x, cont;
    //a variável cont contará quantas vezes o valor x aparece no vetor
    cont = 0;
    printf("\nEntre com os 20 valores:\n");
    for(i=0; i<20; i++){
        scanf("%d",&vet[i]);
    }
    printf("\nEntre com o valor x: ");
    scanf("%d",&x);
    for(i=0;i<20;i++){
        if(vet[i]==x){
            printf("\nO valor %d apareceu na posicao vet[%d].", x,i);
            cont++; //aqui estamos contando quantas vezes o x aparece
        }
    }
    if(cont==0) printf("\nO valor nao foi encontrado!\n");
    return 0;
}

```

- 6) Crie um programa que leia um vetor de 40 posições e atribua valor 0 para todos os elementos que possuírem valores negativos.

```

#include<stdio.h>
int main(){
    int vet[40], i;

```

```

printf("\nEntre com os valores:\n");
for(i=0;i<40;i++){
    scanf("%d",&vet[i]);
}
for(i=0;i<40;i++){
    if(vet[i]<0) vet[i] = 0;
}
printf("\nO vetor corrigido eh:\n");
for(i=0;i<40;i++){
    printf("%d\n",vet[i]);
}
return 0;
}

```

- 7) Crie um programa que leia dois vetores de 20 posições e calcule um outro vetor contendo, nas posições pares os valores do primeiro e nas posições ímpares os valores do segundo.

```

#include<stdio.h>
int main(){
    int vet1[20], vet2[20], vet3[40], i, i1, i2;
    i1 = i2 = 0;
    printf("\nEntre com o primeiro vetor:\n");
    for(i=0; i<20; i++){
        scanf("%d",&vet1[i]);
    }
    printf("\nEntre com o segundo vetor:\n");
    for(i=0; i<20; i++){
        scanf("%d",&vet2[i]);
    }
    for(i=0;i<40;i++){
        if(i%2==0){
            vet3[i]=vet1[i1];
            i1++;
        }
        else{
            vet3[i]=vet2[i2];
            i2++;
        }
        printf("\nvet3[%d]=%d",i, vet3[i]);
    }
    return 0;
}

```

- 8) Escreva um programa para ler dois vetores (A e B), ambos com 5 elementos inteiros. Em seguida crie um vetor X com 10 elementos e copie os valores de A e B para esse vetor (união de A com B).

```

#include<stdio.h>
int main(){
    int a[5], b[5], c[10], i;
    printf("\n Entre com o vetor a: \n");
    for(i=0;i<5;i++) scanf("%d",&a[i]);
    printf("\n Entre com o vetor b: \n");
    for(i=0;i<5;i++) scanf("%d",&b[i]);
    for(i=0;i<5;i++){
        c[i] = a[i];
        c[i+5] = b[i];
    }
    printf("\n O vetor C:");
    for(i=0;i<10;i++) printf(" %d",c[i]);
    return 0;
}

```

- 9) Desenvolva um programa que leia um vetor de 10 posições inteiras e coloque em ordem crescente, utilizando a seguinte estratégia de ordenação:
- Selecione o elemento do vetor de 10 posições que apresenta o menor valor;
 - Troque esse elemento pelo primeiro;
 - Repita estas operações, envolvendo agora apenas os 9 elementos restantes (trocando o de menor valor com a segunda posição), depois os 8 elementos restantes (trocando o de menor valor com a terceira posição), depois os 7 últimos restantes; os 6 e assim por diante, até restar um único elemento, o maior deles.

Exemplo de entrada e saída:

Entrada: 22, 3, 9, -1, 4, 6, 11, 3, 8, 1

Saída: -1, 1, 3, 3, 4, 6, 8, 9, 11, 22

```
#include<stdio.h>

int main(){
    int vet[10], i, k = 0, menor, posicaoEncontrada;
    printf("Digite os elementos:");
    for(i = 0; i < 10; i++){
        scanf("%d", &vet[i]);
    }

    for(k = 0; k < 10; k++){
        menor = vet[k];
        posicaoEncontrada = k;
        for(i = k; i < 10; i++){
            if(vet[i] < menor){
                menor = vet[i];
                posicaoEncontrada = i;
            }
        }
        vet[posicaoEncontrada] = vet[k];
        vet[k] = menor;
    }

    printf("O vetor ordenado eh:\n");
    for(i = 0; i < 10; i++){
        printf("%d ", vet[i]);
    }

    return 0;
}
```

- 10) Escreva um programa em que seja declarado um vetor de caracteres de tamanho 80 e lhe seja atribuído um texto qualquer (não é necessário que o usuário entre com o texto). Depois, esse programa deve calcular e imprimir a quantidade de palavras existentes no referido texto. Por exemplo, considerando-se que o símbolo □ representa espaço, para um vetor cujo conteúdo é:

“□□A□linguagem□C□□nasceu□na□decada□de□□1970.□Seu□□□inventor□foi□Dennis
□Ritchie.□”

O resultado a ser impresso deverá ser o seguinte: **13 palavras.**

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

#define MAX 80
```

```

#define TRUE 1
#define FALSE 0

int main()
{
    char texto[MAX] = " A linguagem C nasceu na década de 1970. Seu
inventor foi Dennis Ritchie. ";

    int i, numPal = 1, inicio = TRUE;

    for (i = 0; i < MAX; i++) {
        // verifica se não é um espaço na posição corrente
        if (texto[i] != ' ') {
            // se não achou letras e chegou ao caractere de fim de string
            if ((inicio) && (texto[i] == '\0')) {
                numPal = 0;
            } else {
                inicio = FALSE;
            }
        }
        // se achou um espaço no início, passe para o próximo caractere
        else if ((inicio) && (texto[i] == ' ')) {
            continue;
        }
        // se achou um espaço sozinho e o caractere seguinte não é o delimitador
para fim de string, então uma palavra foi encontrada
        else if ((texto[i] == ' ') && (texto[i+1] != ' ') && (texto[i+1] !=
'\0')) {
            numPal++;
        }
    }

    printf("%d palavra(s)\n", numPal);

    system("pause");
    return 0;
}

```

Parte 2: Matrizes

11) Faça um programa que leia uma matriz M[2,2] e uma matriz N[2,2]. A seguir, calcule a soma de M e N, colocando os resultados em uma matriz SOMA[2,2]. Imprima a matriz SOMA.

```

#include<stdio.h>
int main(){
int m[2][2], n[2][2], matrizsoma[2][2], l, c;
printf(" Digite a Matriz M");
for(l=0;l<2;l++){
for(c=0; c<2; c++){
scanf("%d",&m[l][c]);
}
}
printf(" Digite a Matriz N");
for(l=0;l<2;l++){
for(c=0; c<2; c++){
scanf("%d",&n[l][c]);
}
}
for (l=0;l<2;l++){
for(c=0; c<2; c++){

```

```

        matrizsoma[l][c]=m[l][c]+n[l][c];
    }
}
printf("a matriz soma eh\n ");
for (l=0;l<2;l++){
    for (c=0; c<2; c++){
        printf("%d ", matrizsoma[l][c]);
    }
    printf("\n");
}
return 0;
}

```

12) Faça um programa que leia uma matriz S[2,2] e um valor X. A seguir, multiplique a matriz S pelo valor X, colocando o resultado em um vetor V[4]. Mostre o vetor V[4].

```

#include<stdio.h>
int main(){
    int S[2][2], V[4], i, j, k = 0, x;
    printf("Digite x:");
    scanf("%d", &x);
    printf("Digite a matriz S:\n");
    for(i = 0; i < 2; i++){
        for(j = 0; j < 2; j++){
            scanf("%d", &S[i][j]);
            V[k] = S[i][j] * x;
            k++;
        }
    }

    for(k = 0; k < 4; k++){
        printf("%d ", V[k]);
    }
    return 0;
}

```

13) Escreva um programa que lê uma matriz M[5,5] e calcule as seguintes somas:

- a) Da linha 4 de M;
- b) Da coluna 2 de M;
- c) Da diagonal principal;
- d) Da diagonal secundária;
- e) De todos os elementos da matriz M.

Escreva a matriz lida e cada uma destas somas. Exemplo:

Entrada:

```

1 4 5 6 8
6 3 3 7 1
5 1 1 0 0
1 2 3 4 5
2 8 9 5 5

```

Saída:

```

Soma linha 4: 15
Soma coluna 2: 18
Soma diagonal principal: 14
Soma diagonal secundaria: 20
Soma todos os elementos: 95

```

14) Faça um programa que leia uma matriz 3x2, calcule e imprima a sua transposta. Exemplo:

$$M = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \quad M^T = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$$

```
#include<stdio.h>
int main(){
    int matriz[3][2], i,j;
    printf("Digite a matriz:\n");
    for(i = 0; i < 3; i++){
        for(j = 0; j < 2; j++){
            scanf("%d", &matriz[i][j]);
        }
    }
    printf("A sua transposta e:\n");
    for(j = 0; j < 2; j++){
        for(i = 0; i < 3; i++){
            printf("%d ", matriz[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

15) Dizemos que uma matriz quadrada inteira é um *quadrado mágico* se a soma dos elementos de cada linha, a soma dos elementos de cada coluna e a soma dos elementos das diagonais principal e secundária são todas iguais. Exemplo de matriz quadrado mágico:

$$\begin{bmatrix} 3 & 4 & 8 \\ 10 & 5 & 0 \\ 2 & 6 & 7 \end{bmatrix}$$

Fazer um algoritmo que leia uma matriz 3x3 e verifique se ela é um quadrado mágico.