

[EXERCICIO 1]

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

int main(){
    float A[30][30];
    float V[30];
    int i, j, m, n;
    printf("Entre com o numero de linhas da matriz (m < 30): \n");
    scanf("%d", &m);
    printf("Entre com o numero de colunas da matriz (n < 30): \n");
    scanf("%d", &n);

    printf("Entre com os elementos da matriz (dimensao = m X n): \n");
    for (i = 0; i < m; i++)
    {
        for (j = 0; j < n; j++)
        {
            scanf("%f",&A[i][j]);
        }
    }

    printf("Entre com os elementos do vetor (dimensao = n): \n");
    for (j = 0; j < n; j++)
    {
        scanf("%f",&V[j]);
    }

    printf("\n*** MATRIZ A ***\n");
    for (i = 0; i < m; i++)
    {
        for (j = 0; j < n; j++)
        {
            printf("%5.2f ",A[i][j]);
        }
        printf("\n");
    }

    printf("\n*** VETOR V ***\n");
    for (j = 0; j < n; j++)
    {
        printf("%5.2f ",V[j]);
    }

    printf("\n*****\n");
    printf("\n*** PRODUTO DE A x V ***\n");
    printf("\n*****\n");
    for (i = 0; i < m; i++)
    {

```

```

        for (j = 0; j < n; j++)
        {
            printf("%5.2f ", A[i][j]*V[j]);
        }
        printf("\n");
    }

    getch();
    return 0;
} // fim do programa

```

[EXERCICIO 2]

```

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

```

```

void produto(float A[], float B[], float C[], int m, int n, int p);

```

```

int main(){
    float A[7][5];
    float B[5][3];
    float C[7][3];
    int i, j, k;

    //Inicializacao de C
    for (i = 0; i < 7; i++)
    {
        for (k = 0; k < 3; k++)
        {
            C[i][k] = 0;
        }
    }

    printf("Entre com os elementos da matriz A (dimensao = 7 X 5): \n");
    for (i = 0; i < 7; i++)
    {
        for (j = 0; j < 5; j++)
        {
            scanf("%f", &A[i][j]);
        }
    }

    printf("Entre com os elementos da matriz B (dimensao = 5 X 3): \n");
    for (j = 0; j < 5; j++)
    {
        for (k = 0; k < 3; k++)
        {

```

```

        scanf("%f",&B[j][k]);
    }
}

produto(A,B,C,7,5,3);

printf("\n*** MATRIZ A ***\n");
for (i = 0; i < 7; i++)
{
    for (j = 0; j < 5; j++)
    {
        printf("%5.2f ",A[i][j]);
    }
    printf("\n");
}

printf("\n*** MATRIZ B ***\n");
for (j = 0; j < 5; j++)
{
    for (k = 0; k < 3; k++)
    {
        printf("%5.2f ",B[j][k]);
    }
    printf("\n");
}

printf("\n*****\n");
printf("\n*** PRODUTO DE A x B ***\n");
printf("\n*****\n");
for (i = 0; i < 7; i++)
{
    for (k = 0; k < 3; k++)
    {
        printf("%5.2f ",C[i][k]);
    }
    printf("\n");
}

getch();
return 0;
} // fim do programa

void produto(float A[7][5], float B[5][3], float C[7][3], int m, int n, int p)
{
    int i, j, k, aux;
    for (i = 0; i < m; i++)
    {
        for (k = 0; k < p; k++)
        {

```

```

        aux = 0;
        for (j = 0; j < n; j++)
        {
            aux = aux + A[i][j]*B[j][k];
        }
        C[i][k] = aux;
    }
}
}

```

[EXERCICIO 4]

```

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

int somaVet(int vetor[10], int n);
int main(){
    int i, j, n, soma, aux, estado;
    int matriz[10][10];
    int vet[10];

    printf("Digite a dimensao da matriz:\n");
    scanf("%d",&n);

    printf("Digite os elementos da matriz:\n");
    for(i = 0; i < n; i++)
    {
        for(j = 0; j < n; j++)
        {
            scanf("%d",&matriz[i][j]);
        }
    }

    for(i = 0; i < n; i++)
    {
        vet[i] = matriz[0][i];
    }

    soma = somaVet(vet,n);
    aux = soma;

    //Testa linhas
    estado = 1;
    for(i = 0; i < n; i++)
    {
        for(j = 0; j < n; j++)
        {
            vet[j] = matriz[i][j];

```

```

    }
    aux = somaVet(vet,n);

    if (aux != soma)
    {
        estado = 0;
        break;
    }
}
if (estado == 1) printf("\nAS LINHAS SAO IGUAIS\n");
else printf("\nAS LINHAS SAO DIFERENTES DIFERENTES\n");

//Testa colunas
if (estado)
{
    for(i = 0; i < n; i++)
    {
        for(j = 0; j < n; j++)
        {
            vet[j] = matriz[j][i];
        }
        aux = somaVet(vet,n);

        if (aux != soma)
        {
            estado = 0;
            break;
        }
    }
    if (estado == 1) printf("\nAS COLUNAS SAO IGUAIS\n");
    else printf("\nAS COLUNAS SAO DIFERENTES DIFERENTES\n");
}

//Testa diagonais
if (estado)
{
    for(i = 0; i < n; i++)
    {
        vet[i] = matriz[i][i];
    }
    aux = somaVet(vet,n);

    if (aux != soma) estado = 0;

    if (estado == 1) printf("\nA DIAGONAL PRINCIPAL E IGUAL\n");
    else printf("\n A DIAGONAL PRINCIPAL E DIFERENTE\n");

    for(i = 0; i < n; i++)
    {
        vet[i] = matriz[i][n-i-1];
    }
}

```

```

    }
    aux = somaVet(vet,n);

    if (aux != soma) estado = 0;

    if (estado == 1) printf("\nA DIAGONAL SECUNDARIA E IGUAL\n");
    else printf("\n A DIAGONAL SECUNDARIA E DIFERENTE\n");
}

//Resultado final
printf("\n\n*****\n");
if (estado) printf("\nA MATRIZ POSSUI A PROPRIEDADE ALFA\n");
else printf("\nA MATRIZ NAO POSSUI A PROPRIEDADE ALFA\n");
printf("\n\n*****\n");
getch();
return 0;
} // fim do programa

int somaVet(int vetor[10], int n)
{
    int k, sum;
    sum = 0;
    for (k = 0; k < n; k++)
        sum = sum + vetor[k];
    return sum;
}

```

[EXERCICIO 6]

```

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

int n = 10;

int main(){
    int i, j, n;
    int **matriz;
    printf("Digite o grau do triangulo de Pascal:\n");
    scanf("%d",&n);

    matriz = (int **) malloc(n*sizeof(int *));
    for (i = 0; i < n; i++)
        matriz[i] = (int *) malloc(n*sizeof(int));

    for(i = 0; i < n; i++)
    {
        for(j = 0; j < n; j++)
        {
            if ((j == 0) || (j == i))

```

```

        matriz[i][j] = 1;
    else
        matriz[i][j] = 0;
    }
}

//Piramide de Pascal
for(i = 2; i < n; i++)
{
    for(j = 1; j < i; j++)
    {
        matriz[i][j] = matriz[i-1][j] + matriz[i-1][j-1];
    }
}

printf("\n\n*****\n");
printf("*** TRIANGULO DE PASCAL ***\n");
printf("*****\n");
for(i = 0; i < n; i++)
{
    for(j = 0; j <= i; j++)
    {
        printf("%d\t", matriz[i][j]);
    }
    printf("\n");
}

getch();
return 0;
} // fim do programa

```