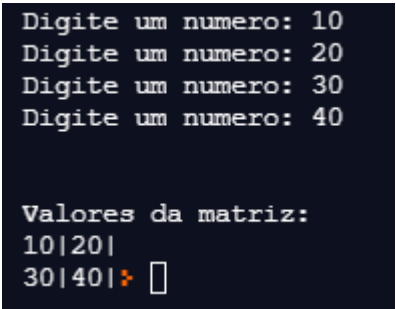


## PESQUISA 3 – TIPOS DE MATRIZES

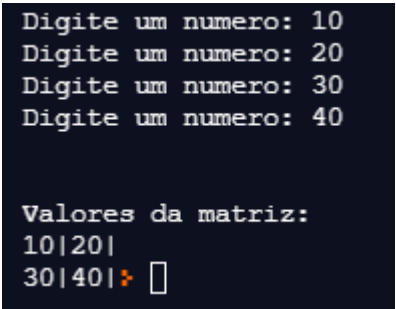
Uma matriz é uma estrutura de dados vetoriais com dimensões entre linhas e colunas, formando tabelas que são armazenadas na memória. Existem 4 tipos principais de matrizes, que serão apresentados abaixo:

- **Matriz quadrada:** são aquelas que o número de linhas é igual ao número de colunas, formando assim um quadrado como podemos ver no código e no resultado abaixo:

```
int main(){
int matriz[2][2];
int l,c,cont=0;
for(l=0; l<2; l++){
for(c=0; c<2; c++){
printf("Digite um
numero: ");
scanf("%d",
&matriz[l][c]);
}
} for(l=0; l<2; l++){
printf("\n");
printf("\n\nValores da
matriz:");
for(c=0; c<2; c++){
printf("%d|", matriz[l][c]);
}
}
return 0;
}
```



```
Digite um numero: 10
Digite um numero: 20
Digite um numero: 30
Digite um numero: 40
```



```
Valores da matriz:
10|20|
30|40|
```

- **Matriz transposta:** a matriz transposta é uma matriz de tamanho n,m , depende do que o usuário deseja, que inverte a posição das linhas e colunas na matriz original, ou seja, se a matriz original possui tamanho 3x4, a transposta terá 4x3, veja o exemplo:

```
int main() {
int matriz[5][5];
int i, j;
for(i = 0; i < 5; i++) {
for(j = 0; j < 5; j++) {
printf("Digite o valor da linha
%d, coluna %d: ", i + 1, j + 1);
scanf("%d",&matriz[i][j]);
}
} printf("\n");
printf("Matriz original: \n");
for(i = 0; i < 5; i++) {
for(j = 0; j < 5; j++) {
printf("%d |", matriz[i][j]);
}
}
```

```

        printf("\n");
    }
    printf("\n");
    printf("Matriz transposta: \n");
    for(i = 0; i < 5; i++) {
        for(j = 0; j < 5; j++) {
            printf("%d |", matriz[j][i]);
        }
        printf("\n");
    }
}

```

```

Digite o valor da linha 1, coluna 1: 1
Digite o valor da linha 1, coluna 2: 2
Digite o valor da linha 1, coluna 3: 3
Digite o valor da linha 1, coluna 4: 4
Digite o valor da linha 1, coluna 5: 5
Digite o valor da linha 2, coluna 1: 6
Digite o valor da linha 2, coluna 2: 7
Digite o valor da linha 2, coluna 3: 8
Digite o valor da linha 2, coluna 4: 9
Digite o valor da linha 2, coluna 5: 9
Digite o valor da linha 3, coluna 1: 8
Digite o valor da linha 3, coluna 2: 7
Digite o valor da linha 3, coluna 3: 4
Digite o valor da linha 3, coluna 4: 5
Digite o valor da linha 3, coluna 5: 6
Digite o valor da linha 4, coluna 1: 3
Digite o valor da linha 4, coluna 2: 2
Digite o valor da linha 4, coluna 3: 1
Digite o valor da linha 4, coluna 4: 0
Digite o valor da linha 4, coluna 5: 1
Digite o valor da linha 5, coluna 1: 2
Digite o valor da linha 5, coluna 2: 3
Digite o valor da linha 5, coluna 3: 4
Digite o valor da linha 5, coluna 4: 5
Digite o valor da linha 5, coluna 5: 6

```

Matriz original:

```

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

```

Matriz transposta:

```

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

```

- **Matriz identidade:** é um exemplo de auxílio de contas que são realizadas com matrizes, onde todos os elementos da diagonal principal são 1 e os demais 0:

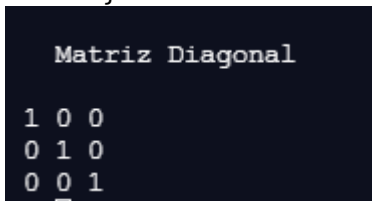
```

int main(){
    int i, j, m[3][3], soma[3];

    for(i=0; i < 3; i++){
        for (j=0; j < 3; j++){
            if(i==j){
                m[i][j] = 1;
            } else {
                m[i][j] = 0;
            }
        }
    }
    printf("\n\n Matriz Diagonal\n\n");
    for(i=0; i < 3; i++){
        for (j=0; j < 3; j++){
            printf("%i ", m[i][j]);
        }
        printf("\n");
    }

    return 0;
}

```



```

Matriz Diagonal
1 0 0
0 1 0
0 0 1

```

- **Matriz inversa:** Matriz inversa é a junção de dois outros tipos que já vimos acima, a matriz quadrada e a matriz identidade. Para que seja inversa, deve ser quadrada e o resultado da multiplicação de duas matrizes que resultam em uma matriz identidade é a inversa. Veja o exemplo abaixo:

```

int main()
{
    int Matriz_A[3][3];
    int linha,coluna, tab;
    int pivo =0, p = 0, m = 0, k = 0;

    for (linha=0; linha<3;linha++){
        for (coluna=0; coluna<3;coluna++){

            printf("Digite o elemento linha %d coluna %d: ",
linha+1, coluna+1);
            scanf("%d", &Matriz_A[linha][coluna]);
        }
    }
}

```

```

    }

    printf("-----\n");
    printf("Matriz Digitada:\n");
    printf("-----\n");

    for (linha=0; linha<3;linha++){
        for (coluna=0; coluna<3;coluna++){
            printf("%d ", Matriz_A[linha][coluna]);
        }
        printf("\n");
    }

    printf("\n\n");
    printf("-----\n");
    printf("DIAGONAL PRINCIPAL\n");
    printf("-----\n");

    for(linha=0; linha < 3; linha++){
        for(coluna=0; coluna < 3; coluna++){
            if (linha == coluna){
                printf("%d ", Matriz_A[linha][coluna]);
            } else{
                printf(" ");
            }
        }
        printf("\n");
    }

    printf("\n\n");
    printf("-----\n");
    printf("MATRIZ IDENTIDADE\n");
    printf("-----\n");

    int identidade[linha][coluna];

    for(linha = 0; linha < 3; linha++){
        for(coluna = 0; coluna < 3; coluna++){
            if(linha == coluna){
                identidade[linha][coluna] = 1;
            }else{
                identidade[linha][coluna] = 0;
            }
        }
    }

    for(linha = 0; linha < 3; linha++){

```

```

        for(coluna = 0; coluna < 3; coluna++){
            printf("%d", identidade[linha][coluna]);
            if(coluna < 3 - 1){
                printf("\t");
            }else{
                printf("\n");
            }
        }
    }

    printf("\n\n");
    printf("-----\n");
    printf("MATRIZ INVERSA\n");
    printf("-----\n");

    for(coluna = 0; coluna < 3; coluna++){
        pivo = Matriz_A[coluna][coluna];
        for(k = 0; k < 3; k++){
            Matriz_A[coluna][k] = (Matriz_A[coluna][k])/(pivo);
            identidade[coluna][k] = (identidade[coluna][k])/(pivo);
        }

        for(linha = 0; linha < 3; linha++){
            if(linha != coluna){
                m = Matriz_A[linha][coluna];
                for(k = 0; k < 3; k++){
                    Matriz_A[linha][k] = (Matriz_A[linha][k]) -
(m*Matriz_A[coluna][k]);
                    identidade[linha][k] = (identidade[linha][k]) -
(m*identidade[coluna][k]);
                }
            }
        }

        for(linha = 0; linha < 3; linha++){
            for(coluna = 0; coluna < 3; coluna++){
                printf("%d \t", identidade[linha][coluna]);

            }

            printf("\n");
        }
    }
}

```

```

Digite o elemento linha 1 coluna 1: 1
Digite o elemento linha 1 coluna 2: 2
Digite o elemento linha 1 coluna 3: 3
Digite o elemento linha 2 coluna 1: 0
Digite o elemento linha 2 coluna 2: 4
Digite o elemento linha 2 coluna 3: 6
Digite o elemento linha 3 coluna 1: 5
Digite o elemento linha 3 coluna 2: 6
Digite o elemento linha 3 coluna 3: 1
-----
Matriz Digitada:
-----
1 2 3
0 4 6
5 6 1

-----
DIAGONAL PRINCIPAL
-----
1
 4
  1

-----
MATRIZ IDENTIDADE
-----
1  0  0
0  1  0
0  0  1

-----
MATRIZ INVERSA
-----
1  0  0
0  1  0
-5 0  1

```

- **Multiplicação de matriz:** Para realizar a multiplicação de matrizes pode-se ser criada uma nova variável que assume o valor da multiplicação, e esta nova é exibida no printf final. Mas veja abaixo o conceito de multiplicação, junto com o código e o resultado final:

$$B \cdot A = \begin{bmatrix} -1 & 3 \\ 4 & 2 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} (-1) \cdot 1 + 3 \cdot 3 & (-1) \cdot 2 + 3 \cdot 4 \\ 4 \cdot 1 + 2 \cdot 3 & 4 \cdot 2 + 2 \cdot 4 \end{bmatrix} = \begin{bmatrix} 8 & 10 \\ 10 & 16 \end{bmatrix}$$

```

int main(int argc, char *argv[])
{ int linha;

```

```

int coluna;
int i;
int somaprod;
int mat1[3][3]={1,2,3},{4,5,6},{7,8,9};
int mat2[3][3]={1,0,0},{0,1,0},{0,0,1};
int mat3[3][3];
int M1L=3, M1C=3, M2L=3, M2C=3;
for(linha=0; linha<M1L; linha++){
    for(coluna=0; coluna<M2C; coluna++){
        somaprod=0;
        for(i=0; i<M1L; i++) somaprod+=mat1[linha][i]*mat2[i][coluna];
        mat3[linha][coluna]=somaprod;
    }
}
for(linha=0; linha<M1L; linha++){
    for(coluna=0; coluna<M2C; coluna++){
        printf("%d ", mat3[linha][coluna]);
        printf("\n");
    }
}
return 0;
}

```

1	2	3
4	5	6
7	8	9

<https://www.guj.com.br/t/matriz-inversa-em-c/327735/3>

<https://mundoeducacao.uol.com.br/matematica/matriz-inversa.htm>

<https://www.todamateria.com.br/matriz-inversa/>

<https://www.todamateria.com.br/matriz-identidade/>

<https://github.com/eltonbaroncello/aula-algoritmos-ifc/blob/master/p03-05-matriz-identidade.c>

<http://acesso.materdei.edu.br/omero/C/Exercicios/Lista%20Matrizes/12.Htm>

<https://github.com/marquesds/linguagemc/blob/master/matriz-transposta.c>

<https://brasilecola.uol.com.br/matematica/matriz-transposta.htm>

<https://www.clubedohardware.com.br/topic/902517-matrizes-quadradas-c/>

<https://www.somatematica.com.br/emedio/matrizes/matrizes4.php>