

Função de leitura de Matriz

```
void leitura (int mat[3][3]){
    int l, c;
    /*ciclo de leitura*/
    for (l=0; l<3; l++) {
        for (c=0; c<3; c++) {
            scanf("%d", &mat[l][c]);
        }
    }
}
```

Função para imprimir Matriz

```
void imprimir (int mat[3][3]){
    int l, c;
    /*ciclo de escrita*/
    for (l=0; l<3; l++) {
        for (c=0; c<3; c++) {
            printf("%d ", mat[l][c]);
        }
        printf("\n")
    }
}
```

Alocação dinâmica para Matrizes

```
int** alocarMatriz(int Linhas, int Colunas){
    int i,j;
    int **m = (int**)malloc(Linhas * sizeof(int));
    for(i=0; i<Linhas; i++) {
        m[i] = (int*)malloc(Colunas * sizeof(int));
        for(j=0; j<Colunas; j++){
            m[i][j] = 0;
        }
    }
    return m; //retorna o Ponteiro para a Matriz
}
```

Função para somar valores de uma Matriz

```
void somaMatriz (int mat[3][3]) {
    int l, c, soma = 0;
    for (l=0; l<3; l++) {
        for (c=0; c<3; c++) {
            soma = soma + mat[l][c];
        }
    }
    printf("soma = %d\n", soma);
}
```

Função para somar 2 Matrizes

```
void somaMatrizes (float A[2][3], float B[2][3]) {
    float S[2][3];
    int l, c;
    for (l=0; l<2; l++) {
        for (c=0; c<3; c++) {
            S[l][c] = A[l][c] + B[l][c];
        }
    }
    for (l=0; l<2; l++) {
        for (c=0; c<3; c++) {
            printf("%d ", S[l][c]);
        }
        printf("\n")
    }
}
```

Função para criar Matriz Identidade

```
#define L 10
#define C 10
void matrizIdentidade(int m[L][C]){
    int i, j;
    for(i=0; i<L; i++){
        for(j=0; j<C; j++){
            /*if(i==j || j==C-i-1) {
                deixa as duas diagonais com valor 1 e os demais 0
            }*/
            if(i==j){
                m[i][j] = 1;
            } else{
                m[i][j] = 0;
            }
        }
    }
}
```

Função para criar Matriz Transposta

```
void matrizTransposta(int mA[3][4]){
    int i, j;
    int mB[4][3];
    for(i=0; i<4; i++){
        for(j=0; j<3; j++){
            mB[i][j] = mA[j][i];
        }
    }
}
```