

```

void mul (const double A[n][n], const double B[n][n], double C[n][n]) {
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            for (int k = 0; k < n; k++)
                C[i][j] += A[i][k] * B[k][j];
    //          (I, K)  x  (K, J),
    // da eine Zeile der ersten Matrix K viele (Spalten viele) Einträge hat
    // da eine Spalte der zweiten Matrix K viele (Zeilen viele) Einträge hat
    // i für Spalten, j für Zeilen in der Matrix C
    // k iteriert durch Zeilen der Matrix A und Spalten der Matrix B
    // i sind Spalten der Matrix A, die Spalten der Matrix C entsprechen
    // j sind Zeilen der Matrix B, die Zeilen der Matrix C entsprechen
    // |a*x+b*y|    |ab|    |x|
    // |-          -| = |--| x |y|
    // da:
    // (J, I) = (I, K) x (K, J)
    // (J, K) x (K, I) = (J, I)
    // (zeilen1, spalten1) x (zeilen2, spalten2) = (zeilen1, spalten2)
    // es muss gelten: spalten1 = zeilen2
    // |ab|    |x|
    //      x |y| = a*x+b*y
    // (1, 2) x (2, 1) = (1, 1)
    // allgemein bei Array: A[spalten][zeilen]
    // |ab|
    // |cd| = {{a, b},{c, d}} = a[Arrays im äußeren Array][Elemente der inneren Arrays]
}

```