

Natürlich! Hier ist ein **komplettes, selbstständiges C-Programm**, das **einfache Tensor-Berechnungen** durchführt

Das Programm ist **für Windows kompatibel**, verwendet **nur ANSI C** und **keine externen Bibliotheken** (keine BLAS, LAPACK, OpenMP, etc.). Es berechnet einen **2x2 Tensor** (also eine 2x2 Matrix), wobei es eine **elementweise Addition** und **Skalarmultiplikation** durchführt – typische Operationen für Tensoren.

🚀 Beispiel: Berechnung eines 2x2 Tensors (Matrix-Operationen)

Wir berechnen:

$$C = A + B$$

$$D = 2 * C$$

mit zwei 2x2 Matrizen A und B.

✔️ C-Quellcode (keine externen Bibliotheken, kompatibel mit Windows)

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

```
// Funktion zur Addition zweier 2x2 Matrizen
```

```
void add_tensors(float a[2][2], float b[2][2], float result[2][2]) {
```

```
    for (int i = 0; i < 2; i++) {
```

```
        for (int j = 0; j < 2; j++) {
```

```
            result[i][j] = a[i][j] + b[i][j];
```

```
        }
```

```
    }
```

```
}
```

```
// Funktion zur Skalarmultiplikation eines 2x2 Tensors
```

```
void scalar_multiply(float tensor[2][2], float scalar, float result[2][2]) {
```

```
    for (int i = 0; i < 2; i++) {
```

```
        for (int j = 0; j < 2; j++) {
```

```
            result[i][j] = tensor[i][j] * scalar;
```

```
        }
```

```
    }
```

```
}
```

```

// Funktion zur Ausgabe eines 2x2 Tensors
void print_tensor(float tensor[2][2]) {
    printf("Tensor:\n");
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 2; j++) {
            printf("%.2f ", tensor[i][j]);
        }
        printf("\n");
    }
}

int main() {
    // Definition der beiden Ausgangstensoren A und B
    float A[2][2] = {{1.0, 2.0}, {3.0, 4.0}};
    float B[2][2] = {{5.0, 6.0}, {7.0, 8.0}};
    // Ergebnis-Tensor für A + B
    float C[2][2];
    // Ergebnis-Tensor für 2 * (A + B)
    float D[2][2];
    // Schritt 1: A + B = C
    add_tensors(A, B, C);
    printf("A =\n");
    print_tensor(A);
    printf("B =\n");
    print_tensor(B);
    printf("A + B = C\n");
    print_tensor(C);
    // Schritt 2: 2 * C = D
    scalar_multiply(C, 2.0, D);
    printf("2 * (A + B) = D\n");
    print_tensor(D);
    return 0;
}

```

🛠️ Wie das Programm funktioniert:

- Addition von Matrizen (Tensor-Addition)
- Skalarmultiplikation (Tensor-Multiplikation mit Skalar)

🖨️ Ausgabe-Beispiel (auf der Konsole):

A =

1.00 2.00

3.00 4.00

B =

5.00 6.00

7.00 8.00

A + B = C

6.00 8.00

10.00 12.00

2 * (A + B) = D

12.00 16.00

20.00 24.00

✅ Fazit

Dieses Beispiel zeigt eine ****einfache, vollständig selbstständige C-Implementation**** für Tensor-Berechnungen .

Wenn du möchtest, kann ich auch eine Erweiterung für 3D-Tensoren oder Tensor-Produkte (z. B. Tensor-Produkt $A \otimes B$) bereitstellen .