```
!nvcc --version
```

```
    nvcc: NVIDIA (R) Cuda compiler driver
    Copyright (c) 2005-2023 NVIDIA Corporation
    Built on Tue_Aug_15_22:02:13_PDT_2023
    Cuda compilation tools, release 12.2, V12.2.140
    Build cuda_12.2.r12.2/compiler.33191640_0
```

Start coding or generate with AI.

Start coding or generate with AI.

```
%%writefile add.cu
#include <stdio.h>
#define MAX_SIZE 100

void add(int vector1[], int vector2[], int size, int result[]) {
    for (int i = 0; i < size; i++) {
        result[i] = vector1[i] + vector2[i];
    }
}

int main() {
    int size;
    int vector1[MAX_SIZE], vector2[MAX_SIZE], result[MAX_SIZE];

    printf("Enter the size of vectors: ");
    scanf("%d", &size);

    printf("Enter elements of the 1st vector:\n");
    for (int i = 0; i < size; i++) {
        printf("Enter element %d: ", i + 1);
        scanf("%d", &vector1[i]);
    }

    printf("Enter elements of the 2nd vector:\n");
    for (int i = 0; i < size; i++) {
        printf("Enter element %d: ", i + 1);
        scanf("%d", &vector2[i]);
    }

    add(vector1, vector2, size, result);

    printf("Resultant vector after addition:\n");
    for (int i = 0; i < size; i++) {
        printf("%d ", result[i]);
    }
    printf("\n");

    return 0;
}
```

```
Overwriting add.cu
```

```
!nvcc add.cu -o add
!./add
```

```
Enter the size of vectors: 2
Enter elements of the 1st vector:
Enter element 1: 3
Enter element 2: 4
Enter elements of the 2nd vector:
Enter element 1: 5
Enter element 2: 1
Resultant vector after addition:
8 5
```

Start coding or generate with AI.

Start coding or generate with AI.

```
%%writefile matrix.cu
#include <stdio.h>
#define MAX_SIZE 10

void multiplyMatrix(int mat1[][MAX_SIZE], int mat2[][MAX_SIZE], int result[][MAX_
    if (col1 != row2) {
        printf("Matrix multiplication not possible\n");
        return;
    }

    for (int i = 0; i < row1; i++) {
        for (int j = 0; j < col2; j++) {
            result[i][j] = 0;
            for (int k = 0; k < col1; k++) {
                result[i][j] += mat1[i][k] * mat2[k][j];
            }
        }
    }
}

void displayMatrix(int mat[][MAX_SIZE], int row, int col) {
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) {
            printf("%d ", mat[i][j]);
        }
        printf("\n");
    }
}

int main() {
    int row1, col1, row2, col2;
    int mat1[MAX_SIZE][MAX_SIZE], mat2[MAX_SIZE][MAX_SIZE], result[MAX_SIZE][MAX_

    printf("Enter the number of rows and columns of first matrix: ");
    scanf("%d %d", &row1, &col1);

    printf("Enter the elements of first matrix:\n");
    for (int i = 0; i < row1; i++) {
        for (int j = 0; j < col1; j++) {
            printf("Enter element [%d][%d]: ", i, j);
            scanf("%d", &mat1[i][j]);
        }
    }

    printf("Enter the number of rows and columns of second matrix: ");
    scanf("%d %d", &row2, &col2);

    printf("Enter the elements of second matrix:\n");
    for (int i = 0; i < row2; i++) {
        for (int j = 0; j < col2; j++) {
            printf("Enter element [%d][%d]: ", i, j);
            scanf("%d", &mat2[i][j]);
        }
    }
    multiplyMatrix(mat1, mat2, result, row1, col1, row2, col2);

    printf("Resultant matrix after multiplication:\n");
    displayMatrix(result, row1, col2);

    return 0;
}
```

```
Overwriting matrix.cu
```

```
!nvcc matrix.cu -o matrix
!./matrix
```

```
Enter the number of rows and columns of first matrix: 2
2
Enter the elements of first matrix:
Enter element [0][0]: 2
Enter element [0][1]: 3
Enter element [1][0]: 6
```