Assignment 4 extended

#include <stdio.h>  
#include <stdlib.h>  
#include <math.h>  
  
#define row1 2 /\* Number of rows of first matrix \*/  
#define col1 3 /\* Number of columns of first matrix \*/  
#define row2 3 /\* Number of rows of second matrix \*/  
#define col2 2 /\* Number of columns of second matrix \*/  
  
\_\_global\_\_ void matproductsharedmemory(int \*l, int \*m, int \*n) {  
    int x = blockIdx.x;  
    int y = blockIdx.y;  
    \_\_shared\_\_ int p[col1];  
  
    int i;  
    int k = threadIdx.x;  
  
    n[col2 \* y + x] = 0;  
  
    p[k] = l[col1 \* y + k] \* m[col2 \* k + x];  
  
    \_\_syncthreads();  
  
    for (i = 0; i < col1; i++)  
        n[col2 \* y + x] = n[col2 \* y + x] + p[i];  
}  
  
int main() {  
    int a[row1][col1];  
    int b[row2][col2];  
    int c[row1][col2];  
    int \*d, \*e, \*f;  
    int i, j;  
  
    printf("\nEnter elements of first matrix of size %d\*%d:\n", row1, col1);  
    for (i = 0; i < row1; i++) {  
        for (j = 0; j < col1; j++) {  
            printf("Enter element at position (%d, %d): ", i + 1, j + 1);  
            scanf("%d", &a[i][j]);  
        }  
    }  
  
    printf("\nEnter elements of second matrix of size %d\*%d:\n", row2, col2);  
    for (i = 0; i < row2; i++) {  
        for (j = 0; j < col2; j++) {  
            printf("Enter element at position (%d, %d): ", i + 1, j + 1);  
            scanf("%d", &b[i][j]);  
        }  
    }  
  
    cudaMalloc((void \*\*)&d, row1 \* col1 \* sizeof(int));  
    cudaMalloc((void \*\*)&e, row2 \* col2 \* sizeof(int));  
    cudaMalloc((void \*\*)&f, row1 \* col2 \* sizeof(int));  
  
    cudaMemcpy(d, a, row1 \* col1 \* sizeof(int), cudaMemcpyHostToDevice);  
    cudaMemcpy(e, b, row2 \* col2 \* sizeof(int), cudaMemcpyHostToDevice);  
  
    dim3 grid(col2, row1);  
  
    matproductsharedmemory<<<grid, col1>>>(d, e, f);  
  
    cudaMemcpy(c, f, row1 \* col2 \* sizeof(int), cudaMemcpyDeviceToHost);  
  
    printf("\nProduct of two matrices:\n");  
    for (i = 0; i < row1; i++) {  
        for (j = 0; j < col2; j++) {  
            printf("%d\t", c[i][j]);  
        }  
        printf("\n");  
    }  
  
    cudaFree(d);  
    cudaFree(e);  
    cudaFree(f);  
  
    return 0;  
}