# Pointer

You are not allowed to use array indexing. Use pointer arithmetic in each problem.
You need to allocate memories dynamically. You are not allowed to use array of predefined size.
Make sure to free all allocated memory at the end.

1. Given a 2D Array of integers as input,
    - Output the maximum and minimum value of the array.
    - Also given the index of two rows as input, swap the two rows and output the resulting array.
    - Also given the index of two columns as input, swap the two columns and output the resulting array.
2. Given an MxN 2D integer matrix, compute and print its transpose using pointer arithmetic only. Transpose should be stored in a new dynamically allocated matrix.
3. Given an MxN matrix, sort each column individually in ascending order, using pointer arithmetic. You may use any sorting algorithm, but must access elements via pointers only..

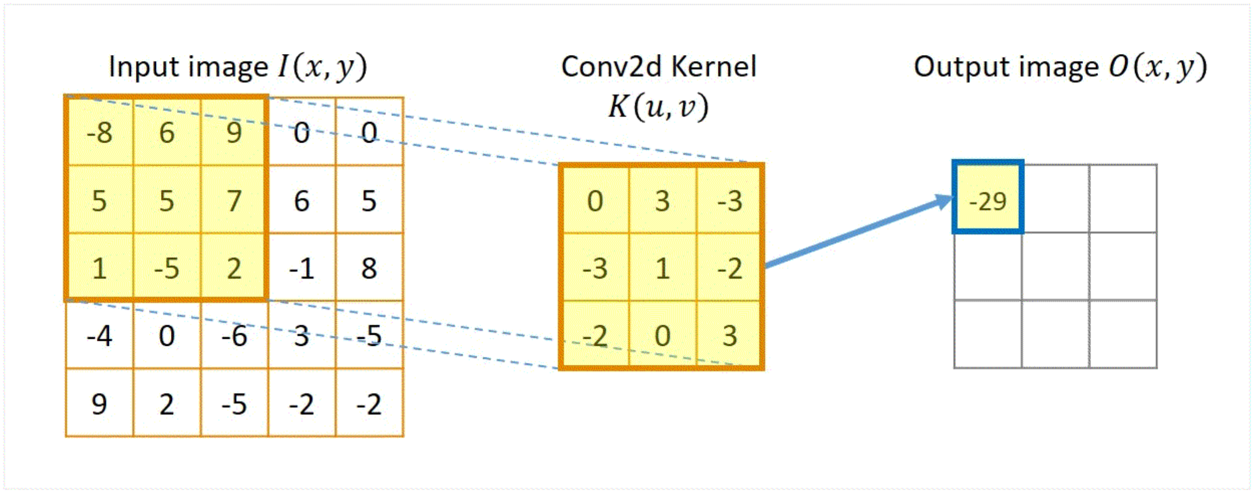
4. Given an integer N, create a 3D integer array, somewhat resembling the shape of a 3d rectangular pyramid, meaning that, the first dimension of the 3d array should be 1x1 matrix, second should be 2x2 and so on. This should go on up to N dimension. Fill the array with random numbers. Then take one of the 5 surfaces as input and print the numbers on that surface in appropriate shape

5. Convolution is an important operation in image processing and deep learning. It involves sliding a small matrix called a kernel (or filter) over a larger 2D input array (like an image or a matrix) to produce an output matrix that is the dot product of the overlapping parts in each step. More specifically, at each position, the kernel and the underlying subregion of the input are multiplied element-wise and summed to produce a single value in the output.

For an input matrix I of size `MxN` and a kernel K of size `KxK` (assume odd dimensions, typically 3x3), the convolution operation produces an output matrix O of size `(M-k+1)x(N-k+1)`

Write a C program that performs 2D convolution of a given matrix (the input array) with a given kernel using pointer arithmetic only. The function should take the input matrix, its dimensions, the kernel, and the kernel dimension as inputs, and output the result of the convolution.
You must not use nested `[][]` array access. Instead, perform all operations using pointer arithmetic `(*(ptr + i * width + j))` to simulate 2D traversal. You are expected to

manually allocate space for the output matrix in main and pass it to the function. You may write helper functions to print matrices, but they should also use pointer arithmetic if accessing matrix elements.


Input image $I(x,y)$
-8 6 9 0 0
5 5 7 6 5
1 -5 2 -1 8
-4 0 -6 3 -5
9 2 -5 -2 -2
Conv2d Kernel $K(u,v)$
0 3 -3
-3 1 -2
-2 0 3
Output image $O(x,y)$
-29