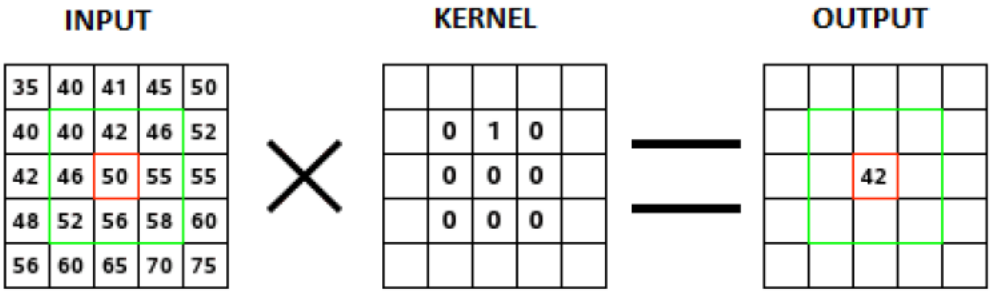
# 2D Convolution

Convolution is a common operation in image processing. An image can be considered as a bidimensional matrix that associates to each pixel $(x, y)$ a corresponding luminosity value for the original image. Convolution consists in modifying an original input matrix by means of a kernel, generally of smaller dimensions. Given a pixel $(x, y)$, so called central pixel, of the image matrix, the convolution consists in centering the kernel matrix on this value. Its value and the neighbors values are multiplied for the corresponding values in the kernel. Results are sum up and the final sum is saved in the original $(x, y)$ initial coordinates. Given an input matrix $I^{m \times n}$ and a Kernel $K^{p \times q}$, every value of the output matrix $O^{m \times n}$ is computed as:

$$O(x, y) = \sum_{i=0}^{p} \sum_{j=0}^{q} I(x + i - \frac{p}{2}, y + j - \frac{q}{2}) * K(i, j)$$

For example:




Assumptions:

- We consider kernels with odd number of rows and columns
- We consider just integer matrixes for input and kernels
- for the edge points we consider the matrix as toroidal (border columns and rows are neighbors).

This project will consider 4 small programs for computing convolutions of matrixes of any sizes.

- Single thread implementation.
- Shared memory solution with OpenMP

- Distributed memory solution with MPI
- Hybrid solution OpenMP + MPI

We use the following comand for the execution:

- `hpc_conv_st rows_input cols_input rows_kernel cols_kernel input_matrix kernel output_matrix`
- `OMP_NUM_THREADS=T hpc_conv_omp rows_input cols_input rows_kernel cols_kernel input_matrix kernel output_matrix`
- `mpirun -np P hpc_conv_mpi rows_input cols_input rows_kernel cols_kernel input_matrix kernel output_matrix`
- `OMP_NUM_THREADS=T mpirun -np P hpc_conv_hybrid rows_input cols_input rows_kernel cols_kernel input_matrix kernel output_matrix`

Every program accepts input matrix of dimensions rows_input $\times$ cols_input from the input_matrix file. Analogously for the kernel. Every program generates a file called output_matrix with the matrix convolution result.

Every file contains space-separated values. Every line ends with newline. The makefile compiles and generates the 4 executables.