Main Task :
Create a neural network in c++ lanquage , based on material in this link http://neuralnetworksanddeeplearning.com/chap1.html . Our neural network must be able to recognize handwritten numbers.

The class Network
As a first approach we will create the **class Network** , wich represents the hole network, including all the layers of neurons .
An object of type Network can be created writing :
Network net = new Network(numOfLayers:integer,sizes:integer[]);*
The first argument is the total number of layers , including the one that we will treat as input .
The second argument is a numOfLayers *x* 1 integer array ,which contains the number of neurons that each layers has.

Each Network object contains two matrixes : weights contains a vector **w** and biases contains a vector **b**, for each layer . Vector w is equal to layer’s size (layer’s size == number of neurons in the layer) and contains the weights that will be used in order to create the w * x sum . Vector b is same size as w and represents the values of biases in each neuron .
When we create a new Network object these two matrixes are being initiallized with random **double** values .

In this first approach we will use a vector w for the input layer (which will be initiallized to [1 1 1 ….]) but no vector b for the input layer . This may be changed in the future.
(actually vector w for input-layer will not be used)

Class network contains the functions :
1. void Network::feedforward(double a)**
the actual argument is a vector a (double* a_array) which contains the input values . But in order to avoid memory issues (when running in a pc with small ram) we pass the memory address of a_array in order to reallocate memory.
Functions modifies a , and when it is complete *a is an array of size sizeOfLayers[end-1] *x* 1 ,which contains the output values of the network , for the given input.

2. double* Network::read_tuple(int offset,int* y)
it can “read” the files in data.zip (they represent handwritten images of numbers and the desired value for each of them) . Function return an array typeof double which represents an specific input to our Network and the desired output (integer value) for this input .

3. int Network::getOutput(double* out)
the argument *out represents the output of our network to a specific input and using this function we can match the network’s ouput (a double vector) to a singleton integer value (which represents a number in 0...9)