# GA Group Projects

- Today we will form teams of several students;
- Each team will implement a GA in Matlab (or C/Java/VB?) to restore a corrupted image:

- Each team should have one good programmer, and access to a notebook computer (preferably with Matlab)!
- You will submit a written report in week 14 and give a short presentation in week 15 (in English)

# GA Group Project: details

- The form of the corruption source is additive noise:

  **N(row,col)=** NoiseAmp**×sin([2π×**NoiseFreqRow**×row]+[2π ×**NoiseFreqCol**×col]))**

- Teams must code a simple GA that optimises the three unknown constants **NoiseAmp**, **NoiseFreqRow**, and **NoiseFreqCol** such that the restoration error (the difference between the original and GA-optimised restored image) is minimised.
- To make things easy, we will measure the average per-pixel restoration error, thus:

  **Restoration error = (Ioriginal + Noise$^{GA}$)-Icorrupted**

  where **Ioriginal** is the original uncorrupted Lena image, **Icorrupted** is the corrupted image (I will give you), and **Noise$^{GA}$** is the modelled GA corruption noise using the noise equation above.

# GA Group Project: details

- Each iteration of your GA will, *for each gene in the population*:
  - Generate new values for **`NoiseAmp`**, **`NoiseFreqRow`**, and **`NoiseFreqCol`**.
  - Corrupt the original image using the equation

`N(row,col)=NoiseAmp×sin([2π×NoiseFreqRow×row]+[2π×NoiseFreqCol×col]))`

  - Measure the restoration error (subtract the GA corrupted image from the original corrupted image). This becomes the (inverse of) this gene's fitness
  - Make new child genes using selection, crossover, and mutation functions.
- The search ranges for the three variables are:
  - **`NoiseAmp`** `0 to 30.0`
  - **`NoiseFreqRow`** `0 to 0.01`
  - **`NoiseFreqCol`** `0 to 0.01`
- Each gene encodes all three variables. If you use 1 byte per variable, each gene will be 24-bits, if you use 2-bytes per variable, 48 bits:

10110111 01010001 11001010 (24-bits per gene)

**`NoiseAmp NoiseFreqRow NoiseFreqCol`**

- You need to map the (binary) integer values of each gene to floating point values for the variables. I.e, for **`NoiseAmp`**, **00000000=0.0 and 11111111=30.0**