Implement image reconstruction from parallel-projection sinograms using Python.  Use your code to reconstruct an image from the given sinogram (available as an image file, sinogram.png, in the Resources/Test Images folder).  Investigate the behaviour of backprojection reconstruction with and without ramp-filtering., and with Hamming-windowed ramp-filtering.

Provide your Python code and a short (3-5 page) report in pdf showing the behaviour of your code for: (a) Reconstruction without ramp-filtering from the given sinogram; (b) reconstruction with ramp-filtering, and; (c) reconstruction using a Hamming-windowed ramp-filter.  Result images should be given in the report, and commented on.  Please do not submit any .rar files.

You may work by yourself or in a group with up to two others (max group size is three).

Hints:  Use the scipack.fft routines from Python's scipack package for efficient discrete forward and inverse fast Fourier Transforms.  Use skimage.image.rotate from the skimage package for image rotation.

The image to be reconstructed has a 4:3 aspect ratio (i.e., it has 4/3 times more columns than rows), this will allow you to recover the image size from the sinogram.  The angle step can be recovered similarly form the sinogram dimensions.

Note also that this is a 3-channel sinogram, it represents a colour (RGB) image.  You will have to reconstruct each channel separately, then combine them into a final colour result image.