White Star Line

**Imagine that you are working for the White Star Line shipping company and you want, more from historical paranoia than any real need, to send out an**[**iceberg-towing**](http://www.shipsandoil.com/Features/Towing%20Icebergs.htm)**tug with all your ships. You need an application to find icebergs and assess their tow-ability.**

You have satellite radar data, and airborne lidar (Light Detection and Ranging) data from an area of sea. The radar data can tell you the texture of objects, while the lidar data can tell you their height. You want to combine the two to spot icebergs and assess their mass. You know from experience that if a berg has a mass of greater than 36 million kg you'll never drag it out of the way of anything in time.

In this example you have an area of sea with a single iceberg in it. The area will be 300m by 300m. The radar file has per m2 values of between 0 and 255. A value of 100 or above is ice. The lidar file data covers the same area at the same m2 resolution, with values between 0 and 255. One Lidar unit equals 10cm height (so 255 is 2550cm high).

**Build a program to do the following...**

1. Pull in the two data files and display them.
2. Assess which areas of the image are ice using the radar data.
3. Assess the total mass of ice above sea level. The lidar returns a value such that 10 units equals a metre of height. The mass density of ice is 900 kg/m3. You'll need to run through the radar image, work out which pixels contain ice and pull out their equivalent heights from the lidar image. Use these to calculate the total mass of the berg above the water. The water level will be at 0m in the lidar file.
4. Calculate the total mass of the iceberg, if 10 percent of its mass is above water.
5. Display the total mass, the total volume and whether you can pull the berg on the GUI.
6. Save the above information to a file.

**Additional marks are awarded for the following.**

Making the program read in a file of multiple bergs and highlight in green the ones you can drag and red those you can't.

**Files for this project.**

2 (300 by 300) raster files representing the radar and lidar images of an area of sea with a single berg in it. The files are laid out at one line per image line, from the top left to bottom right of the raster file.

* [white1.lidar](https://www.geog.leeds.ac.uk/courses/computing/study/core-python-odl2/assessment2/white1.lidar) ([JPG version](https://www.geog.leeds.ac.uk/courses/computing/study/core-python-odl2/assessment2/whitelinesingle.jpg) for comparison - this should not be used in the project.)
* [white1.radar](https://www.geog.leeds.ac.uk/courses/computing/study/core-python-odl2/assessment2/white1.radar) ([GIF version](https://www.geog.leeds.ac.uk/courses/computing/study/core-python-odl2/assessment2/whitelinesingle.gif) for comparison - this should not be used in the project.)

The same with multiple bergs.

* [white2.lidar](https://www.geog.leeds.ac.uk/courses/computing/study/core-python-odl2/assessment2/white2.lidar) ([JPG version](https://www.geog.leeds.ac.uk/courses/computing/study/core-python-odl2/assessment2/whitelinemulti.jpg) for comparison - this should not be used in the project.)
* [white2.radar](https://www.geog.leeds.ac.uk/courses/computing/study/core-python-odl2/assessment2/white2.radar) ([GIF version](https://www.geog.leeds.ac.uk/courses/computing/study/core-python-odl2/assessment2/whitelinemulti.gif) for comparison - this should not be used in the project.)