

We expect that you will complete mini projects *individually*. This means that you *cannot*:

- Share code with friends.
- Find code on the internet.
- Write code together with a partner.
- Submit *any* code that was not written entirely by you.

This does leave room for limited collaboration. The following things *are acceptable*:

- Talking to a partner about strategies you are using to code this project.
- Asking for help *with code that you have written*.
- Offering to help a friend *with code that they have written* as long as you do not give them your code, or copy their code.
- Asking instructors for help.

Mini Project 2 – Mars Rover

This mini-project requires you to create a MATLAB function that will serve as the “brains” for a robot that is traversing the surface of an alien planet. This robot has two missions. One is to search for water. The other is to find the highest ground possible in order to take panoramic photographs back to earth to help scientists survey the landscape.

Write a function called `netIDRover`, where `netID` is your Rutgers netID (the one with the format `aaa111`). It should have four inputs and three outputs, and should be called in the following way:

```
[direction, takeSample, takePhoto] = netIDRover(waterData,  
elevationData, cWater, cElevation);
```

The rest of this document describes what these inputs and outputs are, and how your program should behave. **Note that you do not have to use the same variable names I do, but the inputs and outputs must be in the same order as in the function call above.**

Input Data

Your program will get four inputs, two arrays and two numbers. The two arrays will have three elements, and will be in the following formats:

`waterData` – Water sensor data in parts per million. This array has three elements, in the format [L F R], where

- L – Is the sensor reading for the amount of water vapor to the rover’s left
- F – Is the sensor reading for the amount of water vapor in front of the rover
- R – Is the sensor reading for the amount of water vapor to the rover’s right

`elevationData` – Altimeter readings. This array has three elements, in the format [L F R], where

- L – Is the altimeter reading for the rover’s left
- F – Is the altimeter reading in front of the rover
- R – Is the altimeter reading for the rover’s right

The other two inputs, `cWater` and `cElevation` will be the data for the amount of water vapor and the elevation for the rover’s current position, respectively.

Output Data

Your program will return three outputs:

`direction` – A character (please use vector characters, which use single quotes) for the direction the craft should move. `direction` should be either `'L'`, `'F'`, or `'R'` for left, front, and right, respectively.

`takeSample` – A Boolean variable that is `true` if a soil sample should be taken and `false` if one should not be taken.

`takePhoto` – A Boolean variable that is `true` if a panoramic photo should be taken and `false` if one should not be taken.

Program

The rover should make the following decisions:

- **Highest priority:** Find water. If the water vapor reading to the left, to the right, or in front of the rover is higher than the water vapor reading at the current location AND at least one of those values is greater than 700 parts per million, the rover should move in that direction and not take any samples or photos. *If multiple directions are higher, you should choose the direction with the highest reading.*
 - o If `cWater` is larger than all other sensor readings AND it is higher than 1000 parts per million, `takeSample` should be set to `true`, `direction` should be set to `'F'`, and `takePhoto` should be set to `false`. Otherwise `takeSample` should be set to `false` AND you should proceed to the secondary priority
- **Secondary priority:** Find high points. If the altimeter reading to the left, to the right, or in front of the rover is higher than the altimeter reading at the current location, the rover should move in that direction and not take any samples or photos. *If multiple directions are higher, you should choose the direction with the highest reading.*
 - o If `cElevation` is larger than all other altimeter readings, `takePhoto` should be set to `true` and `direction` should be set to `'F'`.

Testing Your Code

Look for the video tutorial and file about the 3D plotting program you will be able to use to test your code.