

Assignment 2

Due: Fri, 1/25/2013, 11:59pm EST

Objective: The objective of this assignment is to further familiarize yourself with MATLAB . The following exercises are meant to be answered by either a script or a function m-file. The descriptions are complete but the nature of the input/output and display options is left to you. Also, some problems are much simpler than others. Suggestion: start simple and add complexity.

1 Internet Research

1. **(5 pts)** Please explain what a Minkowski addition is and give at least two examples. DO NOT copy of the examples from the online source verbatim. Remember to cite all your sources.
2. **(5 pts)** Please explain how dead-reckoning is achieved on ground mobile robots. What other types of autonomous vehicles use this approach to figure out its location in the world? Remember to cite all your sources.

2 Programming

3. **(10 pts)** In this exercise you will write a program that converts a Roman numeral to its decimal equivalent.
 1. Write a MATLAB that takes a Roman numeral and converts it to its decimal counterpart following the “old style” where the order of the symbols does not matter. In this case, *IX* and *XI* both mean $10 + 1$ or 11. You should be able to handle the following conversion table:

Roman	Decimal
<i>I</i>	1
<i>V</i>	5
<i>X</i>	10
<i>L</i>	50
<i>C</i>	100
<i>D</i>	500
<i>M</i>	1000

2. Write a separate function that takes a Roman numeral and converts it to its decimal counterpart following the “new” style where the order of the symbols does matter. For example, *IX* is $9 = (10 - 1)$, *XC* is $90 = (100 - 10)$. The conversion table given above still holds and you may assume for this case that the only instances of “order” you will encounter are

IV (4), *IX* (9), *XL* (40), *XC* (90), *CD* (400) and *CM* (900).

IMPORTANT: Name your functions `roman2dec_old_user.m` and `roman2dec_new_user.m` respectively. Replace `user` with your drexel.edu e-mail username. For example, for my two functions, I would name my functions `roman2dec_new_mah349.m`.

5. (20 pts) Implement the Connected-Component Labeling Algorithm discussed in class. In `wk3Matlab.zip` you will find 3 MATLAB m-files and 4 PNG images. First, `ccl_user.m` provides a basic code structure for you. The functions `addToStack.m` and `popOffStack.m` are helper functions that you can use to add and remove elements from a stack. Your code should go in the section indicated by the comment `YOUR CODE HERE`. You may need to change the cutoff value when you convert the images to a binary image since your results will differ for different cutoff values. Additionally, the way `ccl_user.m` is set up, it will segment out the lighter regions of the image rather than the darker ones. Feel free to change any part of this code as you see fit or start from scratch.

I suggest you begin with the image named `bwImage1.png` since this is the simplest image of all four. Once you have debugged your code and it is working, you can try it on the rest of the images. Once your code is working, pick your favorite image and try your code on it.

What to turn in: In a single PDF file, include your typed answers to questions in the Internet Research portion of the homework, a detailed description of the various MATLAB script and functions you have generated for this coding portion of the assignment, AND detailed instructions on how to run your code. The last part is especially important if you decide to NOT use the starter code provided to you. In addition, place all the requested MATLAB script and functions and all supporting files (image files, additional m-files, etc.) in a separate zipped file and submit the zipped file with your single PDF file. You will only get full credit if we are able to run your code without any errors. Submit your assignment by emailing the requested files to `MEM380.grader@gmail.com`.