Exemption Exam for ES2

Date: Practice Exam

*[Practice Exam Note: this set of questions is **LONGER** than the hour long exam will be, as it contains **more** questions than will be actually featured on the real Exemption Exam for ES2.]*

Instructions for Exemption Exam for ES2 (Fall 2020)

Exam Programming Language: You can complete the Exemption Exam for ES2 in any of the following languages. You must select a single language and complete all questions using code and syntax from that language.

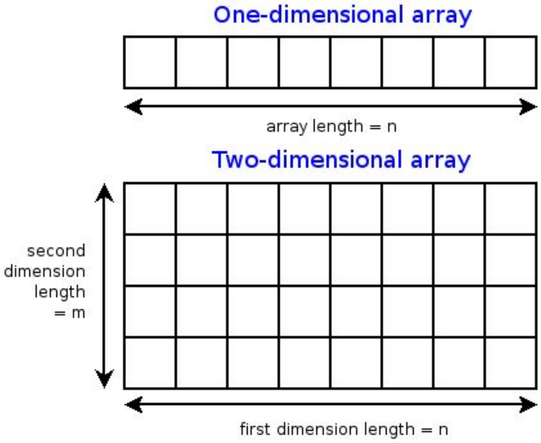
- Matlab
- C
- C++
- Java
- Python

Coding IDE (Integrated Development Environment): You can use any coding environment you wish to complete the exam. If you have a local installation (e.g. Matlab) of software, you are welcome to use that. If you know of an online software you prefer, you are welcome to use that. If you need access to an IDE, then:

- For Matlab, Tufts Engineering students may use the Tufts School of Engineering Virtual Lab that has a current version of Matlab installed.
- For all other languages (C, C++, Java, Python), a CodingRooms Classroom has been created where you can work on completing the exam. See Using a CodingRooms Classroom information on the SoE First Year Courses website.

Format of Code:

- For full credit on each problem (and to help us grade/provide partial credit), please comment your code, at least briefly.
- If a question is unclear to you, or you are not sure about the ***assumptions*** you can make, write a comment in the answer area stating ***HOW*** you interpreted the question and then answer the question following your interpretation.
- Problems may include a data structure used to store data values. Different programming languages may have different names/nomenclature for different data structures or types of data structures. This exam uses the generic term “array” for this, even though a particular language (e.g. the **`list`** data structure in Python) may implement storage in a different way. Use the appropriate language-specific method and the associated access and manipulation functions.



- The term “function” could also represent a function, subfunction, subroutine, or other depending on the language. You should use the appropriate syntax for defining and calling functions.
- If a variable is unnamed in the problem statement, or it is presumed a value or data exists prior to your coded solution, simply state that and indicate the variable/data/etc that you’ll be using. (See “***assumptions***” statement above.) If you create new variables associated with a problem statement, please use good naming conventions so it is clear what that variable represents. We are looking for code blocks and code snippets that solve the specified challenges and problems presented (not necessarily start-to-finish completely working code).

Submitting Your Answers: You should submit your solutions to the questions as individual code files (one for each problem) or as a single ZIP file (compressed archive) of all files together. Use the following Google Form (log in using your *first.last@tufts.edu* email address and Tufts username/password).

Submit answers here: https://forms.gle/bqCD4jE73RkBpj9Y8

Academic Honesty/Statement of Individuality: you must read the following and “sign” your name (part of the Google Form submission above) regarding your submission to the Exemption Exam for ES2.

> *For this exam, I make the following statement: I affirm that I have not given nor received any unauthorized help on this exam, and that all work is my own.*

Questions or Concerns: Leading up to the exam if you have any questions, concerns, or issues please visit the Tufts SoE First Year Courses site with information about the exam or email Prof. Ethan Danahy at ethan.danahy@tufts.edu. During the exam itself, please join the following Zoom session where Prof. Danahy will be available to answer questions about the content of the exam or solve real-time technical issues you might have.

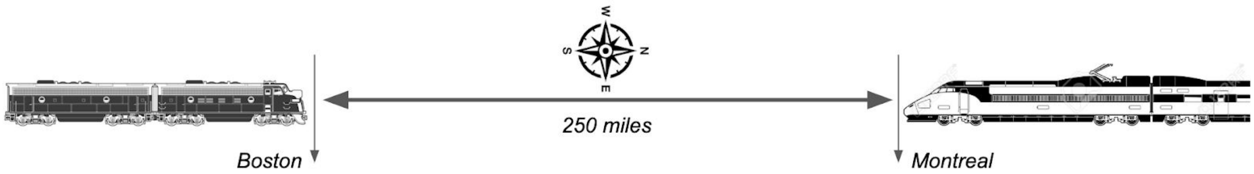
https://tufts.zoom.us/j/94549069221?pwd=VlZFaVNIbzNaTkp3cmlTMWlSejd2UT09
(Meeting ID: 945 4906 9221, Passcode: 226027)

Practice Exam Questions

The following are 7 example questions similar to those found on the Exemption Exam for ES2 (note: this set of questions is LONGER than the hour long exam will be, as it contains more questions than will be actually featured on the real Exemption Exam for ES2.)

Practice Exam Question 1:

A train leaves Boston heading north at 80 miles per hour (mph). A train leaves Montreal (250 miles away) at the same time, traveling south towards Boston, at 180 kilometers per hour (km/h). Write code that prints (to standard output) at what location (in miles north of Boston) they will pass each other. Note 1 mile equals 1.609 km (or 1 mph is equal to 1.609 km/h).



Practice Exam Question 2:

A smart-home thermostat is being connected to a rotary fan in a home to automatically control the fan based on the ambient temperature. The fan's motor (output of the system) is controlled by a voltage ranging from 0-volts (fan/motor is off) to 12-volts (fan/motor is full speed). The system should be programmed to satisfy the following rules: (1) if the temperature in the house is equal to or below 70-degrees, the fan is shut off, (2) if the temperature in the house is equal to or above 106-degrees, the fan is turned on fully, and (3) otherwise, a linear amount of voltage across the possible range is supplied to the fan based on the input temperature across the 70-degree to the 106-degree range. Write code that runs continuously and adjusts the fan speed appropriately according to the temperature. Create necessary input/output functions for reading and setting external values (without the actual internal code for interacting with the hardware; just leave comments where that would be included).

Practice Exam Question 3:

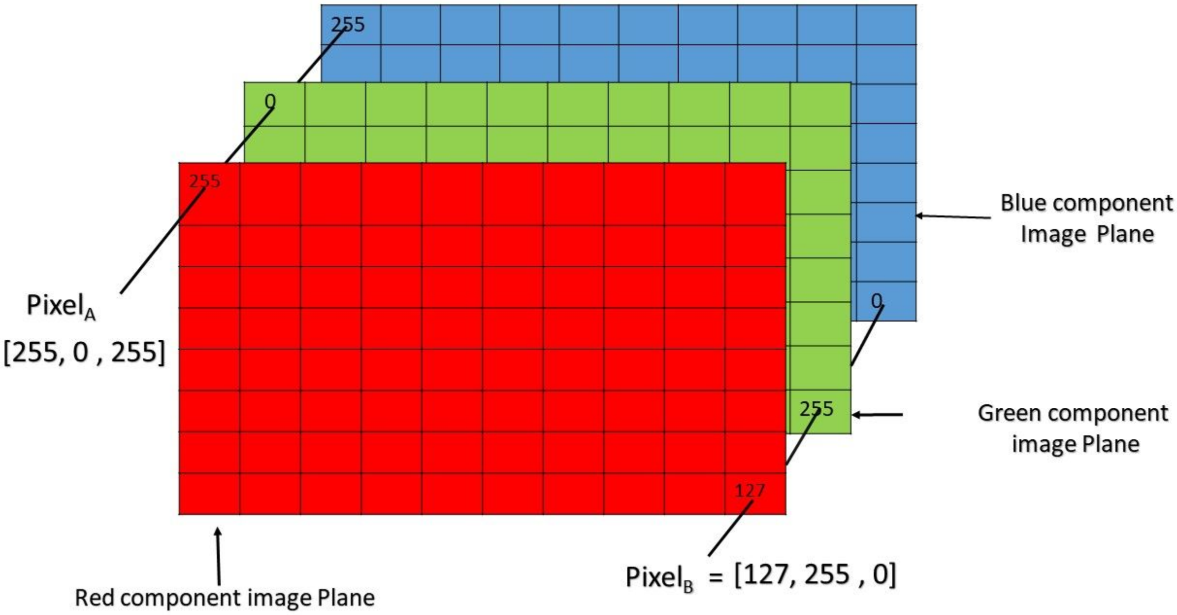
A scientific instrument is attached to your computer for running experiments. When experiments run, the instrument streams data into a buffer, accessible by your code. Your programming language has a **`getVal()`** function that returns the next *positive integer* value sitting on an input buffer. When the buffer is empty (experiment is over), the **`getVal()`** function returns a stop-value integer of **`-1`**. Write code that reads all available valid data values into an array and, once the stop-value is detected, prints to standard output both the total number of data points collected as well as the average (mean) value of the entire data set.

Practice Exam Question 4:

As a car drives across the United States (east to west), the odometer reading (miles) is recorded once per hour from the start of the trip to the finish. Write code, given this data set, to calculate the average velocity (speed, or derivative of the position) for each hour of the trip. Then the code should report out (print to standard output) a specific hour of the trip with the slowest average velocity and a specific hour of the trip with the highest average velocity. (Note: there may be more than one hour that matches either case; simply reporting any single matching hour for each particular case is sufficient.)

Practice Exam Question 5:

A color image is stored in a 3D array with the width and height being the (x,y) pixel locations across the 2D image and the “depth” of the array (3rd dimention) being of size 3, representing the Red, Green, and Blue pixel values (each cell is an integer within the range 0 to 255).


255
0
255
Pixel$_A$
[255, 0 , 255]
Blue component Image Plane
0
255
Green component image Plane
127
Pixel$_B$ = [127, 255 , 0]
Red component image Plane

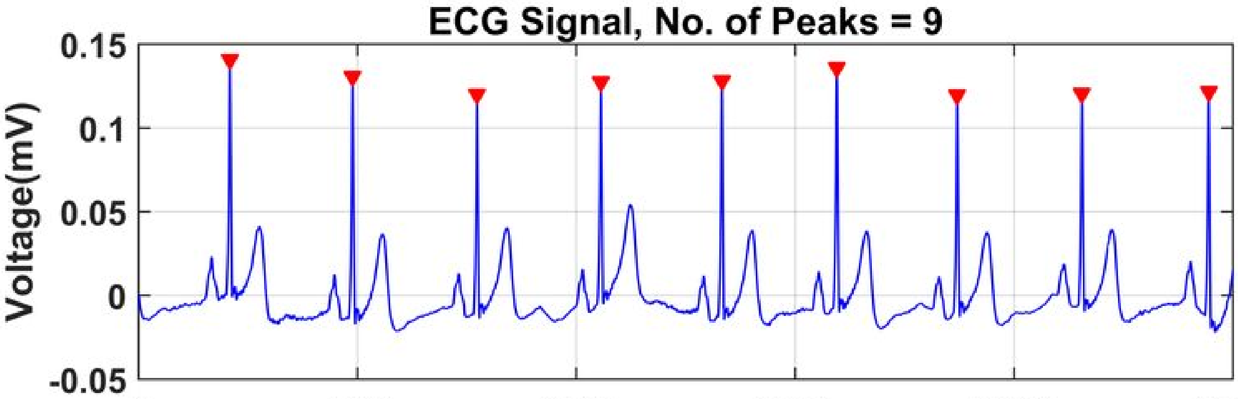
Pixel of an RGB image are formed from the corresponding pixel of the three component images

A grayscale image is a 2D array of the same width and height of the color image, but with one single value (again, in range 0 to 255) representing the grayscale pixel value. This is calculated as an average of the Red, Green, and Blue components for that particular (x,y) pixel location.

Given an input color image (3D array), write code (not using any built-in image converting functions available in your programming language) to create a new output grayscale image (2D array). Note that pixels within grayscale images are also represented with integers, often either the rounded or truncated versions of the floating point calculated averages.

Practice Exam Question 6:

A heartbeat signal is collected from a patient via an Electrocardiogram (ECG), and the ECG data is stored in a 1D array (with millisecond timestamp data for each data point in another, second, matching 1D array). Within a heartbeat signal, a local maximum is defined as a data point within the data set where the two values on either side (before and after the data point) are both lower in value. Heartbeats, e.g. each of the 9 peaks in the following ECG signal, are local maximums that are also above some threshold (e.g. `thresh = 0.1` (mV, in the diagram below)). Programatically compute (identify and store) the collection of associated timestamps of heartbeats (local maximum values above the threshold) and, for the set of resulting values, print out the overall heart rate (average beats per minute) for the patient over the collected data.



Practice Exam Question 7:

One way of computing square roots is via *Newton's method*. Suppose that you want to know the square root of `a`. If you start with almost any estimate, `x`, you can compute a better estimate (`y`) with the following formula:

$$y = \frac{x + a/x}{2}$$

Write code that asks a user to (1) input an initial number `a` and (2) an initial estimate `x`. Then iteratively calculate a new estimate of the square root of `a` using Newton's method until the value of the estimate changes (improves) by a value of less than `0.01`. Print the final estimate to standard output.