**ECE 480 Final**

Name: **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Date:** 6/5/18, Tuesday, 3:50 p.m. – 5:50 p.m., at 9-401

**Total Points**: 144 points

**Total Grade Points:** 20 grade points. 100 points = 20 grade points. Points more than 100 points count as extra credit scores.

**Note:** you may redo or do up to 5 **(five)** questions of this final exam and submit by Wednesday 6/6/18, 6 p.m. Clearly specify which 5 questions you want to be reconsidered. Resubmission without specifying, without showing clearly your intention or with more than 3 questions resubmitted will not be considered. *No extension beyond 6/6/18 6:00 p.m. is generally allowed even if you are working on Wednesday.* If your working schedule makes it extremely hard to work on the final redo during the day of 6/6/18, then please tell me how much extension (how many hours) you will need to work on the redo questions. You can either resubmit a question you have tried in the final or submit you never tried during the final exam.

Turn in all source code as separate files for the questions asking you to write and run computer programs. Please turn in the final exam in this format: FirstnameLastNameECE480final such as *MickeyMouse*ECE480 as a zip file, a Word document, a PDF file etc.

**Sommerville Project management example in 8th edition (20%)**

1. (20%) Consider the following activity diagram of a project. In this chart, T1, T2, etc. stand for tasks and M1, M2, etc. stand for milestones (this is slide 21 of chapter 5 of Sommerville, 8th edition).



Dates are displayed in the European d/m/y format, so 14/7/03 for milstone1 (M1) stands fro July 14, 2003. In this year of 2003, July 4 fell on Friday. The number of days for a task count only weekdays, not weekends, and also counts the starting date and end date inclusive. For example, the project started on July 4, 2003 Friday and with task 1 (T1) taking 8 days, it will be complete at milestone 1 (M1) on July 14 Monday since there are a total of 8 working days from July 4 2003 till July 14 2003.

1. (4%) **Count** the number of different paths from start date July 4, 2003 (4/7/03) to milestone 4 (M4) with date 4/8/03. **List** each path by the nodes it passes through (for example, there is a path from T4 to T10, indicated by T4, M2, T5, M7, T10, or shorter as T4M2T5M7T10)
2. (6%) **Count** the number n of different paths from start date July 4, 2003 to the finish date 19/9/03**.** How big is n? Is n < 5? Is n < 10 or bigger than 10? **List** all the pathsif not too many.
3. (4%) If task 1 (T1) is delayed by 2 working days, but T2, T3, and T6 are on schedule, will any of the 3 milestones M1, M3, or M4 be delayed? If so, by how many days? Clearly **explain** your answer.
4. (6%) Continuing part (c). If task T1 is delayed by 2 working days, but all the other tasks (including T2, T3, and T6) are on schedule, will the Finish date 19/9/03 be delayed? If so, by how many days? Clearly **explain** your answer.

**Matrix processing example by Nyhoff (34%)**

* 1. . (12%) Code Analysis and reengineering. Please do this one in C programming language.

We have this piece of *spaghetti* code as an exercise of Nyhoff’s data structure book (ECE 304) (chapter 1).

**int row = 0, col;**

**A: col = 0;**

If (col < n) goto B;

goto A;

B: if (row < n) goto C;

goto E;

C: if (mat[row][col] == item) goto D;

col ++;

If (col < n) goto B;

row ++;

goto A;

D: cout << “item found\n”;

goto F;

E: cout << “item not found\n”;

F: ;

1. **(3%) Follow** the logic of this program and explain what it intends to do (does this compute a row sum, do a linear search, a binary search, a sort etc.?).
2. **(3%) Explain** why this is called *spaghetti* code.
3. **(6%) Write** a C program (in C language) that incorporates this spaghetti code and use a matrix mat =



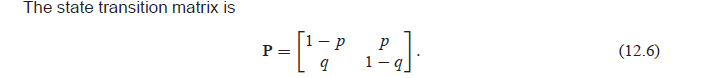
* 1. (12%) **Continue** Q2 in C programming language

1. **Reengineer** or convert this to *structured* code.
2. **Run** the modified structured code with the same matrix. Show the results
3. **Modify** the structured code so that it computes and prints out the row sum for all rows.
4. **Enhance / modify** the code so that it computes the prints out the column sum for all columns as well.
   1. (10%) **Redo** Q3 above in Python programming language, i.e. for the structured code, please do this in Python 2 or Python 3 and have your Python code perform the operation of Q2, and computes and prints out the row sum and also the column sums for all three rows and three columns of the matrix in Q2 (c). If you are not familiar with Python, or Python by any means does not support this matrix operation, then do this question in MATLAB, in C#, in Java, in Javascript, or any other high level language that you are familiar with.

**Matrix processing (software) of Markov chain concept in ECE 315 (probability and statistics). (42%)**

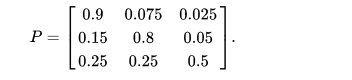
Markov chain is a well known math concept for engineers (especially ECE). In matrix form, a Markov chain M of dimension 2 is a 2x2 matrix with all the entries nonnegative and the row sums equal to 1. In a different book or literature, a Markov chain N of dimension 2 can be a 2x2 matrix with all the entries nonnegative, but the column sums equal to 1, but row sums may not equal to 1. In other words, a Markov chain N in this sense is equal to the transpose of Markov chain M mentioned earlier.

An example of 2x2 Markov chain with row sum = 1 comes as below from Yates and Goodman’s probability book as follows (2nd edition, page 449)



An example of a 3x3 Markov chain with row sums = 1 (and column sums may not equal 1) looks as below from Wikipedia:

<https://en.wikipedia.org/wiki/Markov_chain>



Note some other books or literature use the transpose of matrix P (called P’ or P t in some books) for 2x2 matrix and 3x3 matrix as Markov chain. In that case, the Markov chain has column sums equal to 1, but row sums may not equal 1.

* 1. (10%) **Write** a C program so that it verifies

1. (4%) It verifies that the 2x2 matrix P in (12.6) and the 3x3 matrix P from Wikipedia are Markov chain.
2. (6%) Enhance the program form part (a) so that it can tell if this is a Markov chain with row sum = 1 (and column sum may equal 1 or not) or a Markov chain with column sum = 1 (and row sum may equal 1 or not).
   1. (10%) Redo Q5 with Python, MATLAB, C# etc. of a language other than C language. Your program must be able to verify if the two matrices P of 2x2 or 3x3 or their transposes are Markov chain with row sum = 1or column sum = 1.
   2. **(12%) Test cases** for Markov Chains. You have learned the concept of test cases in chapter 8 of Sommerville, 10th edition.

In Q5, you had done the *normal* test cases where either P or P’ (transpose of P) is Markov chain with row sums = 1 (or column sums = 1).

Now provide some special test cases for **3x3 matrix** and verify your code will generate correct output like “this is a Markov chain with row sum = 1, but column not = 1”, “this is a Markov chain with column sum = 1, but row sum not = 1”, “this is NOT a Markov chain since entry (2, 2) is negative”, “ this is a NOT a Markov chain since row sum and column sum both are not equal to 1”, etc.

1. Test case A (boundary case): provide some test case with 0 entry(ies).
2. Test case B: provide a symmetric 3x3 Markov chain where all row sums = all column sums = 1)
3. Test case C: some entry is negative
4. Test case D: some row sum is NOT 1, also, some column sum is NOT 1.

**Verify** your C code from Q5 pass all 4 test cases (or enhance / correct that to make that work).

* 1. (10%) Coding standard. Remember my brief coverage of coding standard (not in Sommerville’s book, at least not explicitly).

**Enhance** your C code of Q7, or Q5, or Q3 (one of them is enough) to provide these:

1. The programming header in the front shows the author (which is you).
2. The programming header shows the date of today.
3. The programming header shows the purpose of this program
4. Most lines of code are commented, so that it is easy to understand the code without explanation.

You can do this question with MATLAB code, Python code etc. that you had done earlier. Clearly indicate which question you are referring to.

**Program Management (related to project management) (12%)**

* 1. (12%) You are the program manager of a company with some work contracting out. You are requesting proposals (called RFP) from other companies.

In the RFP, you put down these bullets:

* Duration of the work: (not to exceed) 9 months
* Flat fee: (not to exceed) $120,000.
* Salaries: $6,000 per engineer-month, $8,000 per manager-month.

You have a baseline that they should not propose more than $120,000, but you can accept *revised* proposal for up to $150,000.

**Proposals Received:**

*Company A:*

Month 1 – 3: Requirements by 2 engineers

Month 4- 5: Design and Testing by 2 engineers

Month 6: Documentation by 1 engineer

Month 7: Delivery by 1 manager

Deliver by the end of month 7.

*Company B:*

Month 1-2: Requirements spec by 1 engineer

Month 3-5: Design by 2 engineers

Month 6-8: Testing by 2 engineers

Deliver by the end of month 8.

Month 1 – 8: a manager manages the project

*Company C:*

Month 1-2: Requirements spec by 1 engineer

Month 3-5: Design by 2 engineers

Month 6-8: Implementation by 2 engineers

Month 9-11: Testing by 1 engineer.

Deliver by the end of month 11.

Month 1 – 11: a manager manages the project

* 1. Compute the man-months of each of these three companies.
  2. Compute the duration of these 3 proposals.
  3. Compute the proposed cost of these 3 proposals.
  4. Rank the 3 proposals by different criteria. Can you put some criteria?
  5. Do you think any (or all) of these 3 proposals meet the RFP criteria and also general software development practice? Why?
  6. What defect do you see in each of these 3 proposals and how do you want to suggest them to revise? If a proposal is OK, state so.

**The very common killer software: Google map and related system / software engineering: Requirements and Design (36%)**

You have learned requirements in chapter 4 and design in chapter 6 of Sommerville. Of course, you can do this even without learning from Sommerville officially since these concepts are easy to understand and straightforward.

* 1. **(18%) Google map** *requirements:* system, and software

Every one is familiar with Google map (or Map quest) on your laptop or on your IPhone, Android etc. There are system requirements such as communication satellite, communication links and of course Google map software etc. There are Google map software requirements such as a map database to map to the streets, someway to show detailed step by step instructions, someway to announce in voice of where you are etc. . Nobody outside like us knows how Google started the Google map. So please make n educated guess.

1. (3%) **Write** down / guess 3 (or more) Google map system requirements.
2. (6%) Explain briefly why these are required for the system.
3. (3%) Write down / guess 3 (or more) Google map software requirements.
4. (6%) Explain briefly why these are required for the software.
   1. (18%+) Google map *software* design. Assume you are the Google map software designer yourself. Write in a few paragraphs (in English), in pseudo code, in flow charts, in UML, which every way you feel most comfortable on the *top level design* of Google map that you feel, that you can present to Larry Page, Sergei Brin, or some top executives of Google.

This one will convey your design ideas in hopefully half a page to a page that should relate to the basic system requirements and software requirements you explained in Q10.