CSC/MAT-220 **Name:** _____
Fall 2017
Final
Handout: 12/6, Due: 12/14 **Pledge:** _____

Each question topic and point value is recorded in the tables below. You may review these topics from any resource at your leisure. Once you decide to start an exam problem, you are on the clock and you must work without any external resources. Each problem can be done one at a time, but must be finished in a single sitting. Answer each question in the space provided, if you run out of room, then you may continue on the back of the page. It is your responsibility to plan out your time to ensure that you can finish all problems within the 7.0 hours allotted. All SML problems should be put in a single file titled *name_final.sml*. Use comments so that I may easily see where each problem begins and ends, and drop this file in your dropbox folder by the due date. By writing your name and signing the pledge you are stating that your work adheres to these terms and the Davidson honor code.

Scoring Table

Question	Points	Score
1	10	
2	10	
3	10	
4	10	
5	10	
6	10	
7	10	
8	15	
9	15	
Total:	100	

Topics Table

Question	Topic
1	Recurrence Relations
2	Functions
3	Cardinality and Infinity
4	Permutations
5	Probability
6	Graphs
7	Trees
8	SML Concepts
9	SML Programming

Time Table

Question	1	2	3	4	5	6	7	8	9
Time									

1. (10 points) Solve the following recurrence relations given the initial conditions.
   (a) $a_n = 4 - 2a_{n-1}$, $a_0 = 0$.

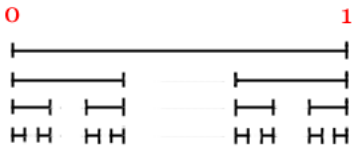
   (b) $a_n = 2a_{n-1} + 5a_{n-2}$, $a_0 = 2$ and $a_1 = 3$.

2. (a) (4 points) State the definition of a function, the domain and image of a function, and what it means when we write $f\colon\ A \to B$.

   (b) (6 points) Define $f\colon \mathbb{Z} \to \mathbb{Z}$ by $(a, b) \in f$ provided that $a + b = 5$. Prove that f is both one-to-one and onto.

3. (a) (2 points) State the definition of the cardinality of a set and the relation of two sets having the same cardinality.

(b) (8 points) We construct the famous Cantor Set as follows. Starting with the interval $[0, 1]$ remove the open middle third $(1/3, 2/3)$. Then, from what is remaining we remove the open middle thirds $(1/9, 2/9)$ and $(7/9, 8/9)$. The first three steps of this process are illustrated below.



Please answer each of the following, with brief justification.

- What is leftover after the 1st, 2nd, and 3rd steps?
- Are there more points leftover after the 1st, 2nd, or 3rd steps?

Provide a conjecture for each of the following, with brief justification.

- How many points will be left over after any finite number of steps?
- How many points will be left over if this process is repeated forever?

4. (a) (2 points) State the definition of a permutation and the symmetric group on n elements (S_n).

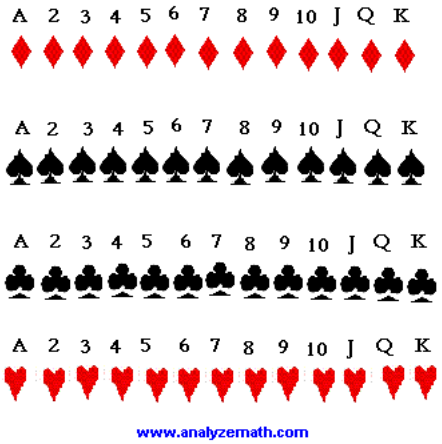
(b) (4 points) Prove that the cardinality of S_n is $n!$.

(c) (4 points) Consider the permutation $\pi = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 3 & 1 & 2 & 4 & 5 & 7 & 6 \end{bmatrix}$. Draw and graph of π and then write π as a collection of pairwise disjoint cycles and as a composition of transpositions. Find π^{-1}.

5. (a) (4 points) State the definition of a sample space, event, random variable, and expectation value.

(b) (6 points) Consider the sample space determined by randomly drawing two cards (one at a time without replacement) from a standard deck of 52, see figure to the right. Please answer each of the following.

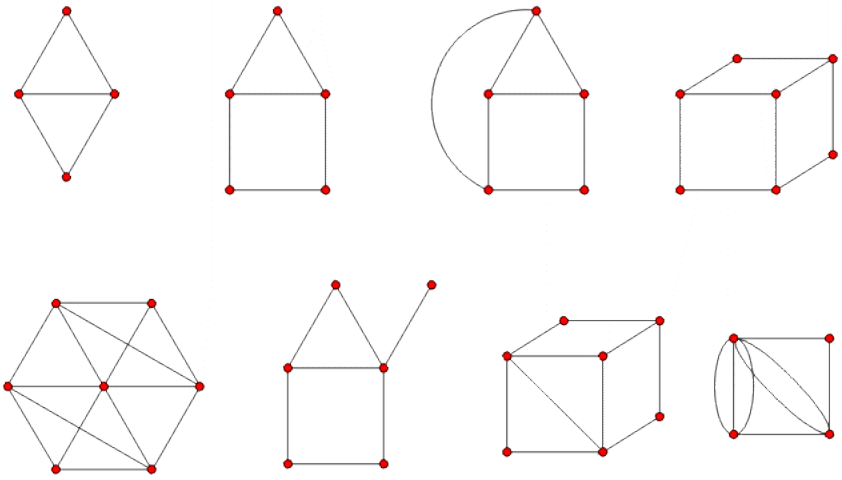
- How many outcomes are in this sample space?
- Let X be a random variable defined on the sample space, where X is 2 if both cards have the same rank, 1 if both cards have the same suit, and 0 otherwise. What is $P(X = 2)$, $P(X = 1)$, and $P(X = 0)$?
- What is the expectation value $E(X)$?



6. (a) (2 points) State the definition of a Simple Graph and a Multi Graph.

(b) (3 points) Draw Euler's Graph representation of the Seven Bridges of Königsberg. Explain why this graph does not have an Euler walk.

(c) (5 points) Put Yes/No next to each graph below, denoting whether or not it has an Euler walk.



7. (a) (3 points) State the definition of a tree, forest, and leaf.

(b) (3 points) Show that a tree can also be defined recursively. State the base case and recursive step.

(c) (4 points) Use proof by mathematical induction to show that a tree on $n \geq 2$ vertices has at least 2 leaves.

8. (a) (5 points) Describe what is happening in the following snippet of code in terms of bindings and pattern matching. What does the last expression evaluate to?

$$datatype\ nat\ =\ Zero \mid Succ\ of\ nat$$

$$\begin{aligned} &val\ rec\ depth\ :\ nat\ \rightarrow\ int\ = \\ &\quad fn\ Zero\ =>\ 0 \\ &\quad \mid Succ\ n\ =>\ 1 + depth\ n \end{aligned}$$

$$val\ test\ =\ depth(Succ(Succ\ Zero))$$

(b) (5 points) Describe what is wrong with the following function expressions.

- fn (x,y) => (x+y) / 2
- fn zero => true | x => false
- fn(x:real) => if x=0 then "zero" else "non-zero"

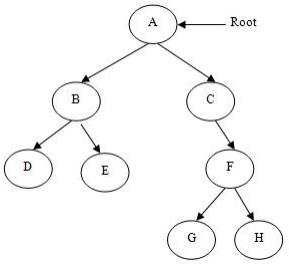
Also, why is it OK to leave out the parenthesis in the expression: *Math.sqrt 2.0*, but it is not OK to leave out the parenthesis in the expression: depth Succ Zero?

(c) (5 points) Give a detailed discussion involving what your learned this semester about SML and some of the benefits of functional programming.

9. In the space provided, include your scratch work to highlight your thought process when solving each problem. Then write the code in your *sml file*. You will be graded on your code compiling, performing the correct behavior, and clear evidence of preparation below. If you can't get your code to compile, then comment out the parts that don't work, you will be given partial credit based on your work below.

   (a) (5 points) Write a polymorphic recursive function called *revAppend* that takes on two lists $(l1, l2)$ and returns the reversal of $l1$ appended to $l2$. Don't use any built in functions of SML.
   Create the two lists $l1 = [1, 2, 3, 4]$ *and* $l2 = [2, 3, 4]$ *and provide the test of* $revAppend(l1, l2)$.

   (b) (5 points) Define the parametrized datastructure *'a tree* as we did in Lab 8. *Construct a tree of type char tree that matches the graph below.*



   (c) (5 points) Write a polymorphic recursive function that performs an inOrder traversal on a tree and searches for a given target value. If the target value is found, then the function returns true; otherwise the function returns false. Note that this is not a BST and should not be turned in to one. All I am asking is for you to slightly modify the traversal we did in Lab 8 to also perform a search, note that you will have to provide context in order for this to work.
   Provide the test on the tree from (b).