Final Exam

- 3 hours long
- Some multiple choice
- Some fill in the blank
- Some written answer
- Includes terminology questions and applied questions

Topics:

- History – the figures covered in the worksheet given in class

- Algorithms (iterative and recursive)
    - algorithm design to solve a problem
        - Flowchart representation
        - Pseudocode representation
    - Efficiency (BigO)
    - Sorting (Selection sort (lab) + Bubble sort (assignment)
- Number Representation
    - binary or hex conversion to/from decimal of a value in the 3 different formats:
        - Unsigned, signed magnitude, 2's complement
    - Addition/subtraction of binary and hexadecimal
    - Big versus Little endian

- Architecture
    - Von Neuman architecture
    - Memory hierarchy
    - CPU components (ALU, general registers, program counter, instruction register)
    - Circuit design and using truth tables

- Assembly language
    - You will always be given the SimpleMachine ASM instruction sheet
        - Do not memorize, learn how to use it as a tool
    - Be able to translate C to assembly
    - Be able to translate assembly <-> machine code
    - Understand the difference between static allocation (memory is allocated at compile time) and dynamic allocation (memory is allocated when the program is running)
    - Be able to determine whether an assembly instruction is reading-from or writing-to memory or not accessing memory at all.
- Graphs:
    - Traversals, terminology (from slides)

- Databases:
    - Given a set of tables, write a query to answer a described question (as in your assignment)

Some suggested practice for some of the topics listed above.

Algorithm Design (iterative and recursive)

Redo the examples from lecture and lab.

More Practice questions:
Design algorithms for the following problems and represent them with flowcharts and pseudocode…

1. Design an algorithm that will search a list for a given number. The algorithm should takes a list of numbers, the length and the number you are looking for. If the number is found, the index it was found at should be returned, otherwise it will return a -1. Design both an iterative and recursive solution. Compare the runtime efficiency of the two.
   Examples:
   in list [1, 3, 5, 2] looking for 3 would return 1 since 3 is at index 1 of the list
   in list [1, 3, 5, 2] looking for 7 would return -1 since 7 is not found in the list

2. Design an algorithm that takes a list of numbers and the length of that list as input. The algorithm should count the number of values in that list that are odd. The algorithm should return the number of odd values in the list. Design both an iterative and recursive solution. Compare the runtime efficiency of the two.
   NOTE: You can assume there is function called **odd** that take a number and returns true if the number is odd and false if it isn't. You may call this function in your algorithm.

3. Design an algorithm that takes a number between 1 and 6 as input and rolls a dice over and over until the number rolled is the same as the input number. The algorithm should count and return the number of dice rolls it took to roll the input number.
   NOTE: You can assume there is a function called rollDice that doesn't take any input but returns a random number between 1 and 6. You may call this function in your algorithm.

4. Design an algorithm that takes a list of numbers and the length of that list as input. The algorithm should find and return the biggest number in the list. Design both an iterative and recursive solution. Compare the runtime efficiency of the two.
   NOTE: You can assume the list has at least one number in it.

5. Design a recursive algorithm that takes a binary tree and doubles every value in the tree.

6. Design a recursive algorithm that takes a binary tree and a number (n) and counts the number of nodes in the tree with values greater than n.

Number Representation

Redo question from the assignment and lab and lecture

Make your own questions by picking a number and converting it, (hex, binary). You can check your answers with online converters: https://www.rapidtables.com/convert/number/hex-to-decimal.html

Make your own hex or binary addition/subtraction questions. Check your answer by converting the numbers to decimal and checking to see if the result match using the converter above.
For example, adding 16 bit 2's complement numbers:
0xFFA1 and 0x2 (0xFFA1 + 0x0002 = 0xFFA3 is the same as: $-95_{10} + 0002_{10} = -93_{10}$)

Circuit Design and truth tables:

Redo lab examples and lecture examples (nothing as big as slides 29/30 of lecture 10)
Recall precedence from Lecture 10 slides.

More practice questions:
Draw a logic circuit and truth table for (A OR B) AND C
Draw a logic circuit and truth table for A OR B AND C OR NOT (D)
Draw a logic circuit and truth table for A AND B OR NOT(A AND C)
Draw a logic circuit and truth table for NOT(A OR B) AND (C OR D) AND NOT©

Solutions for the circuit diagrams can be found at the link below – try before clicking the link!
Note: the solutions use NAND and NOR gates, which can be replaced with AND feeding into a NOT and an OR feeding into a NOT respectively.
http://sandbox.mc.edu/~bennet/cs110/boolalg/gate.html

Assembly

Redo lab, lecture and assignment exercises. Check the correctness of your solutions by running them through the simulator.

More practice questions:
For the following questions, write the assembly instructions for the code within `foo`. You can assume memory allocation is done for you by the compiler and at runtime.
To check you answers, place your instructions in the correct format including lines for memory allocation and initial values for variables in a .s file and your program through the simulator.

Question 1:

```
int a[4];

void foo() {
  a[3] = a[0] + 2;
}
```

Question 2:

```
int* a;

void foo() {
  a = malloc(4*sizeof(int));
  a[3] = a[0] + 2;
}
```

Question 3:

```
int i;
int a[4];

void foo() {
  i++;
  a[i] = i * 4;
}
```

Question 4:

```
int i;
int j;
int a[10];
int*b;

void foo() {
  b = malloc(4*sizeof(int));
  a[i] = b[i];
  i = a[j] + i;
}
```

Question 5:

```
int i;
int a[10];
int*b;

void foo() {
  b = malloc(4*sizeof(int));
  a[b[i]] = a[i+1];
}
```

SQL

Redo lab, lecture and assignment exercises. Check the correctness of your solutions by running them sqlite3 either on a lab machine on your own computer.

More practice questions:

For the following questions, use the tables on the following page to answer the following questions (sql files attached for you to check correctness of your solutions)

1. Write a query that outputs the full name (first and last) and numerical grade in descending order, of all the students who took CSC 106 that semester, and an got A+. Your output should match this one:

```
first_name  last_name   grade
----------  ----------  ------
Jenifer     Lopez       92
Jack        Johnson     95
```

2. Write query that outputs the V number, name, major and the course number took of those students who have taken a CSC course. You should sort by major then by V number. Your output should match this:

```
Vnum        first_name  last_name   major                 course_num
----------  ----------  ----------  --------------------  ----------
V00787      Oprah       Winfrey     Biology               106
V00787      Oprah       Winfrey     Biology               115
V00135      Marshall    Mathers     Computer Science      225
V00135      Marshall    Mathers     Computer Science      230
V00258      Justin      Timberlake  Computer Science      225
V00258      Justin      Timberlake  Computer Science      230
V00451      Pamela      Anderson    Computer Science      225
V00451      Pamela      Anderson    Computer Science      230
V00922      Jenifer     Lopez       Computer Science      106
V00922      Jenifer     Lopez       Computer Science      115
V00987      Justin      Bieber      Computer Science      106
V00987      Justin      Bieber      Computer Science      110
V00254      Jack        Johnson     Psychology            106
V00587      Michelle    Obama       Psychology            106
```

3. Write a query that outputs the number of each letter grade achieved by only Computer Science students. Your output should match this one:

```
letter_grade     num_students
---------------  ---------------
A                5
A+               3
A-               3
B                2
B+               2
B-               1
C+               1
```

4. Write a query that outputs a list of the students that paid over $2000 in tuition that semester. The list must include their Vnumber, full name (first and last) and the total amount each paid in tuition. The list should be sorted from the biggest amount to the least. Your output should match this one:

```
Vnum      first_name  last_name   total_paid
--------  ----------  ----------  ------------
V00922    Jenifer     Lopez       2442.78
V00987    Justin      Bieber      2442.78
V00135    Marshall    Mathers     3105.72
```

grades table

```
Vnum             course_code      course_n  grade         letter_grade
---------------  ---------------  --------  ------------  ------------
V00987           CSC              106       88            A
V00987           CSC              110       74            B
V00987           MATH             100       68            C+
V00987           MATH             122       82            A-
V00135           CSC              225       85            A
V00135           CSC              230       76            B
V00135           SENG             265       79            B+
V00135           MATH             211       86            A
V00135           MATH             101       92            A+
V00258           CSC              225       92            A+
V00258           CSC              230       81            A-
V00451           CSC              225       72            B-
V00451           CSC              230       87            A
V00922           CSC              106       92            A+
V00922           CSC              115       86            A
V00922           MATH             211       83            A-
V00922           MATH             101       78            B+
V00254           CSC              106       95            A+
V00254           MATH             100       92            A+
V00587           CSC              106       48            F
V00587           MATH             100       56            D
V00787           CSC              106       72            B-
V00787           CSC              115       76            B
V00787           MATH             101       79            B+
```

tuition table

```
course_code      price_per_unit
---------------  ---------------
MATH             372.3
CSC              441.96
SENG             441.96
```

grade_point table

```
letter_grade     point
---------------  ---------------
A+               9
A                8
A-               7
B+               6
B                5
B-               4
C+               3
C                2
D                1
F                0
```

courses table

```
course_code      course_num       unit
---------------  ---------------  --------
CSC              105              1.5
CSC              106              1.5
CSC              110              1.5
CSC              115              1.5
CSC              225              1.5
CSC              230              1.5
MATH             100              1.5
MATH             101              1.5
MATH             122              1.5
MATH             211              1.5
SENG             265              1.5
```

students table

```
Vnum             first_name            last_name             major
---------------  --------------------  --------------------  -------------------------
V00987           Justin                Bieber                Computer Science
V00135           Marshall              Mathers               Computer Science
V00258           Justin                Timberlake            Computer Science
V00451           Pamela                Anderson              Computer Science
V00922           Jenifer               Lopez                 Computer Science
V00254           Jack                  Johnson               Psychology
V00587           Michelle              Obama                 Psychology
V00787           Oprah                 Winfrey               Biology
```

ASM Specification:

Name	Semantics	Assembly	Machine
load immediate	`r[d] ← v`	`ld $v, rd`	`0d-- vvvvvvvv`
load base+offset	`r[d] ← m[(o=p*4)+r[s]]`	`ld 0(rs), rd`	`1psd`
load indexed	`r[d] ← m[r[s]+4*r[i]]`	`ld (rs,ri,4), rd`	`2sid`
store base+offset	`m[(o=p*4)+r[d]] ← r[s]`	`st rs, 0(rd)`	`3spd`
store indexed	`m[r[d]+4*r[i]] ← r[s]`	`st rs, (rd,ri,4)`	`4sdi`
register move	`r[d] ← r[s]`	`mov rs, rd`	`60sd`
add	`r[d] ← r[d] + r[s]`	`add rs, rd`	`61sd`
and	`r[d] ← r[d] & r[s]`	`and rs, rd`	`62sd`
inc	`r[d] ← r[d] + 1`	`inc rd`	`63-d`
inc address	`r[d] ← r[d] + 4`	`inca rd`	`64-d`
dec	`r[d] ← r[d] - 1`	`dec rd`	`65-d`
dec address	`r[d] ← r[d] - 4`	`deca rd`	`66-d`
not	`r[d] ← ~ r[d]`	`not rd`	`67-d`
shift left	`r[d] ← r[d] << s`	`shl $s, rd`	`71ss`
shift right	`r[d] ← r[d] >> -s`	`shr $s, rd`	
halt	*halt machine*	`halt`	`F0--`
nop	*do nothing*	`nop`	`FF--`

Template for a recursive function:

```
recursiveFunction(data)
    if (smallestPossibleProblem? data)
        the simple answer
        return ...
    else
        first part of data ...
        recursiveFunction(smallerProblem(data))
        return ...
```