DISCLAIMER!!!!!!!!!!!!
This is a sample final used previously. You can use it to see the kind of questions you might find. Absence of items on this final (e.g., the adder circuit) doesn't mean they won't be on this year's final. Also, each semester is a little bit different, so not every one of the questions on this sample would be appropriate or fair this year.

Note, if there is any chance that your notes and work might be confused for the answer, circle your answers. If it's ambiguous, it's wrong.

(6)   (5)
A _____   N    _____
(6)   (10)
B _____   O    _____
(4)   (5)
C _____   P    _____
(4)   (5)
D _____   Q    _____
(4)   (5)
E _____   R    _____
(4)   (15)
F _____   S    _____
(4)   (20)
G _____   T    _____
(4)   (20)
H _____   U    _____
(6)   (2)
I _____   V    _____
(4)   (9)
J _____   W    _____
(10)   (12)
K _____   X    _____
(10)   (10)
L _____   Y    _____
(5)   (12)
M _____   Z    _____

------------------------------------------

HONOR CODE
I have not received or given any improper help on this exam.

(your signature)
A(6) Two programs were defined as being equivalent if, given the same input they always produced the same output.

Assume the core of program One consists of

```java
int x = inputField.getInt();
x = 2 * x * x + 1;
x = x / 2 + 1;
outputField.setInt(x);
```

Assume the core of program Two consists of

```java
int x = inputField.getInt();
x = x * x + 1;
outputField.setInt(x);
```

Are these two programs equivalent (ignoring implementation problems like integers getting too large to be stored in a register, etc)?

a) They are equivalent.

b) They not are equivalent.

c) The second program doesn't work for negative numbers.

d) The first program can't handle input of zero.

B(6) Given the Java program fragment:

```java
int x, k, a, c = 0;
a = inputField.getInt();
k = 0;
while (k < 10)
{
    x = inputField.getInt();
    if (x == a)
    {
        c = c + 1;
    }
    k = k + 1;
}
outputField.setInt(c);
```

The program above accepts 11 integers entered by the user of the program. It outputs a single value contained in c. What is the significance of the value of c in relation to the numbers read in?

a) It is the number of times the logical expression in the `if` was true.

b) It is the number of times the loop body was executed.

c) It is the number of times the first number read in matches later numbers.

d) It is both a) and c) but not b).
C(4) A program is left running on a workstation collecting passwords (as described when computer security was discussed). This kind of program was described as
   a) a Three-headed Greek dog.
   b) a Trojan horse
   c) a virus detector
   d) a security audit tool

D(4) If one were foolish enough to try to actually writing a program "halts(p, d)" to solve the halting problem, a naive start might be to look for the absence of which one Java statement to assure that it stops?
   a) if
   b) while
   c) return
   d) int

E(4) For large industrial programming efforts, the number of lines of code produced per day (on average) is quite different from productivity on a small or class program. For large projects the figure is
   a) less than 50 lines a day.
   b) more than a) but less than 100 lines per day.
   c) more than a) or b) but less than 150 lines per day.
   d) more than 150 lines per day.

F(4) In the early days of computing with primitive operating systems a system was called a "batch" system because: (pick one)
   a) It was a constructive blend of hardware and software.
   b) Many programs were place on one tape and all had to be processed before getting any back.
   c) The collection of programs put on a tape were merged together to cooperatively produce a better program. (multi-programming)
   d) None of the above.

G(4) If we were to define sum(N) recursively for positive integers by:
   1. sum(N) = N + sum(N-1);  2. sum(0) = 0;

Which one of the following statements is FALSE?
   a) 2. is the base or halting case.
   b) An algorithm for adding the numbers from 1 to N is described.
   c) It is restricted to positive numbers so the base case can be reached.
   d) A more descriptive name for the function sum(N) is factorial.
H(4) Which of the following statements best indicates what is necessary to avoid writing an infinite while loop?
a) The body of the loop must contain an if statement.
b) At least one function/method must be called in the body of the loop.
c) Something in the body of the loop must cause the value of the logical expression following the "while" to change.
d) The statement k = k - 1; must always be present.

I(6) Arrays make it easier to solve certain programming problems. Others don't need arrays at all. Check off the following to indicate whether or not arrays are need.

(need)(not need) a) Find the larger of two numbers.
(need)(not need) b) Sort a list of names.
(need)(not need) c) Create a "Used Car" database.
(need)(not need) d) Decide whether a number is odd or even.
(need)(not need) e) Count the number of data points read.

J(4) We proved in class that a solution to the halting problem can not exist. Is this an isolated result?
(Mark each line below as True/False)
a) Fortunately, this seems to be the only program that can't be solved.
b) Another example is programs that check the syntax of other programs.
c) Scientists think there are other problems, but none have been identified.

K(10) Below is part of a Java program. Show what the array "data" will contain after this code is executed by filling in the boxes below.

```java
int size = 10;
int data[] = new int[size];
int k = 0;
int m = 3;
int x = 0;

while (k < size)
{
    data[k] = x;
    if (x > m)
    {
        x = 0;
    }
    else
    {
        x = x + 1;
    }
    k = k + 1;
}
```
L(10) Assume that the program above continues with the code that follows and that the numbers you just wrote into data are still there. Show what will be displayed in the StringField "output" after this code runs.

    k = 10;
    int max = data[0];
    int min = data[0];
    while (k > 0)
    {
        k = k - 1;
        if (data[k] < min)
        {
            min = data[k];
        }
        if (data[k] > max)
        {
            max = data[k];
        }
    }
    output.setString("Stats: max = " + max + ", min = " + min);
For the next two questions, assume that we have a database on used cars with the following entries. (These would be stored in 5 arrays.) Assume that you are dealing with a program of the type discussed in the text and covered in the lectures.

Ford    1997    blue    rv      22500
Chevy   1995    green   sedan   7000
Ford    1995    white   truck   7450
Chevy   1996    blue    sedan   1800
Chevy   1995    white   truck   7500
Toyota  1997    red     truck   23995
Ford    1995    green   truck   8590

M(5) Assume that the following query was typed into this program:

|        ||        || red     || sedan   ||         |
|--------|--------|---------|---------|----------|

where a blank TextField implies a wild-card entry.

What output, if any (write NONE if appropriate) is produced?

--------------------------------------------------------------------
|                  |                  |                  |                  |                  |
--------------------------------------------------------------------

N(5) Now assume we've EXTENDED the system so that by putting a less-than(<) in front of a query word, it would mean take entries where is field is less than specified. For example, <10000 would mean select the record only if the price is less than 10000.

|        || <1996  || truck   ||         |
|--------|--------|---------|----------|

What output, if any (write NONE if appropriate) is produced?

--------------------------------------------------------------------
|                  |                  |                  |                  |                  |
--------------------------------------------------------------------
0.1.(6) Say that we run a program for several different amounts of data (represented by N) and get the following timing results:

<table>
<thead>
<tr>
<th>N</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>1s</td>
</tr>
<tr>
<td>2000</td>
<td>8s</td>
</tr>
<tr>
<td>3000</td>
<td>27s</td>
</tr>
<tr>
<td>4000</td>
<td>64s</td>
</tr>
</tbody>
</table>

Give the formula that describes the execution time behavior.

\[ t = A \times \text{__________} \]

2.(4) Compute the value of A: ___________

P(5) Assume we have another program which we have run for several moderate test values of N (say 50) and it finishes in a second or two. Now we want to put the program into production with its full load of 500 cases.

We notice that program keeps churning away, giving no signs of completing. After eight hours we go home. The next morning it's still grinding away. We stop the program to consider what's gone wrong. What would you suggest has happened?

(Mark all that seem plausible at this point.)

a) The program uses an exponential algorithm.

b) The logarithmic algorithm being used has finally caught up with us.

c) It's an example of being too tractable.

d) There's a bug in the program and it has gotten into an infinite loop.

Q(5) Say that we run a program for several different amounts of data (represented by N) and get the following timing results:

<table>
<thead>
<tr>
<th>N</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>3s</td>
</tr>
<tr>
<td>4000</td>
<td>4s</td>
</tr>
<tr>
<td>8000</td>
<td>5s</td>
</tr>
<tr>
<td>16000</td>
<td>6s</td>
</tr>
</tbody>
</table>

Circle one or more of the following problems that might have produced the kind of data shown above.

a) N searches for keys in a sorted list (e.g. telephone book)

b) sorting a list of N addresses

c) solving the "traveling salesman" for N cities

d) doing the "towers of Hanoi" for N disks

e) searching the text of an N word book for the number of "and"s.
R(5) Assume you are writing a program to play checkers. Checkers is a board game like chess in which players alternate moving pieces around on the board and attempting to capture opposing pieces. Assume that you are going to solve this by considering all possible moves by one side, then considering all possible responses by the other player, back, and forth, etc. We'll use simple, average figures to make this simple:

Assume that for each turn a player has, on average, 10 possible moves. Assume that there will be 44 total moves, 22 for each side before one side wins. How many total moves must the computer consider to play a perfect game (one in which all possible moves and responses have been considered)?

Show how you would calculate the answer. The exact numeric answer is not important. A single numeric answer will give you no credit unless you hit it exactly. However, you can get full credit by showing what numbers you plan to manipulate. A very brief explanation can also help you if you mess up some other detail.
S(15) Given the following Rules:

---

Name  Syntax Rules
---

R1: \( <i>j \rightarrow w \), where \( w = //a\ sequence\ of\ one\ or\ more\ letters//\)

R2: \( <e>i \rightarrow <i>j \)

R3: \( <s>i \rightarrow <i>j = <e>k; \)

R4: \( <e>i \rightarrow ( <e>j + <e>k ) \)

R5: \( <e>i \rightarrow ( <e>j * <e>k ) \)

---

Use the rules to show the syntactic derivation of \( S = (( X \ast Y) + Z ); \)
Do not worry about semantic rules for this problem. Steps one and two have been carried out for you.

<table>
<thead>
<tr>
<th>Derivation</th>
<th>Syntax Rule Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) ( &lt;s&gt;1 )</td>
<td>R3: ( &lt;s&gt;1 \rightarrow &lt;i&gt;2 = &lt;e&gt;3; )</td>
</tr>
<tr>
<td>(2) ( &lt;i&gt;2 = &lt;e&gt;3; )</td>
<td>R1: ( &lt;i&gt;2 \rightarrow S )</td>
</tr>
</tbody>
</table>
T(20) Do a trace for the program shown below left using the column headings shown below. Assume that the user enters the numbers 3, 2, 5, and 1 when input is requested. Then answer the questions below. The grade will be based mostly on your answers to the questions, not the trace.

<table>
<thead>
<tr>
<th>st#</th>
<th>CF</th>
<th>AX</th>
<th>p</th>
<th>c</th>
<th>st#</th>
<th>CF</th>
<th>AX</th>
<th>p</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>in</td>
<td>ax</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>copy</td>
<td>n, ax</td>
<td></td>
<td></td>
<td>2</td>
<td>copy</td>
<td>ax, #c0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>copy</td>
<td>c, ax</td>
<td></td>
<td></td>
<td>4</td>
<td>copy</td>
<td>ax, #c1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>copy</td>
<td>p, ax</td>
<td></td>
<td></td>
<td>6</td>
<td>#L0</td>
<td>in</td>
<td>ax</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>mul</td>
<td>ax, p</td>
<td></td>
<td></td>
<td>8</td>
<td>copy</td>
<td>p, ax</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>copy</td>
<td>ax, c</td>
<td></td>
<td></td>
<td>10</td>
<td>add</td>
<td>ax, #c1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>copy</td>
<td>c, ax</td>
<td></td>
<td></td>
<td>12</td>
<td>cmp</td>
<td>ax, n</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>jb</td>
<td>#L0</td>
<td></td>
<td></td>
<td>14</td>
<td>copy</td>
<td>ax, p</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>out</td>
<td>ax</td>
<td></td>
<td></td>
<td>16</td>
<td>halt</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>#c0</td>
<td>0</td>
<td></td>
<td></td>
<td>18</td>
<td>#c1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>p</td>
<td>0</td>
<td></td>
<td></td>
<td>20</td>
<td>n</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>c</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CONTINUE IN RIGHT COLUMN ABOVE
(3) This program uses 1 as a sentinel (value on which to stop). (True/False?)

(3) The looping programmed here depends on a count read in at the beginning which tells how many times we repeat the loop. (True/False?)

(4) What are the line numbers for the nine statements executed exactly once, regardless of the data? ____________________________

(10) The output produced by the program is ____________________
U(20) Use the syntactic and semantic rules shown below to generate the
code for

\[ R = (S \times T); \]

Use the methods shown in lecture and the textbook. The first four
steps have been completed. Finish the job on the back of the previous
page.

<table>
<thead>
<tr>
<th>Syntax Rules</th>
<th>Semantic Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1: (&lt;i&gt;j \rightarrow W )</td>
<td>( M(&lt;i&gt;j) = W )</td>
</tr>
<tr>
<td>R2: (&lt;e&gt;i \rightarrow &lt;i&gt;j )</td>
<td>( M(&lt;e&gt;i) = M(&lt;i&gt;j) )</td>
</tr>
<tr>
<td>( \text{code}(&lt;e&gt;i) = ) //nothing//</td>
<td></td>
</tr>
<tr>
<td>R3: (&lt;s&gt;k \rightarrow &lt;i&gt;j=&lt;e&gt;i; )</td>
<td>( \text{code}(&lt;s&gt;k) = \text{code}(&lt;e&gt;i) )</td>
</tr>
<tr>
<td>COPY AX, M(&lt;e&gt;i)</td>
<td></td>
</tr>
<tr>
<td>COPY M(&lt;i&gt;j), AX</td>
<td></td>
</tr>
<tr>
<td>R5: (&lt;e&gt;i \rightarrow (&lt;e&gt;j \times &lt;e&gt;k) )</td>
<td>( M(&lt;e&gt;i) = \text{CN1} )</td>
</tr>
<tr>
<td>( \text{code}(&lt;e&gt;i) = \text{code}(&lt;e&gt;j) )</td>
<td></td>
</tr>
<tr>
<td>( \text{code}(&lt;e&gt;k) )</td>
<td></td>
</tr>
<tr>
<td>COPY AX, M(&lt;e&gt;j)</td>
<td></td>
</tr>
<tr>
<td>MUL AX, M(&lt;e&gt;k)</td>
<td></td>
</tr>
<tr>
<td>COPY M(&lt;e&gt;i), AX</td>
<td></td>
</tr>
</tbody>
</table>

Derivation | Rules
-----------|----------------|
(1) \(<s>1 \) | R3: \(<s>1 \rightarrow <i>2 =<e>3 \) \( \text{code}(<s>1) = \text{code}(<e>3) \) |
| \( \text{COPY AX, M(<e>3) \) |
| \( \text{COPY M(<i>2), AX \) |
| MEANING \( \text{code}(<s>1) = \text{code}(<e>3) \) |
| \( \text{COPY AX, M(<e>3) \) |
| \( \text{COPY M(<i>2), AX \) |
(2) \(<i>2 = <e>3; \) | R1: \(<i>2 \rightarrow R \) \( M(<i>2) = R \) |
| MEANING \( \text{code}(<s>1) = \text{code}(<e>3) \) |
| \( \text{COPY AX, M(<e>3) \) |
| \( \text{COPY R, AX \) |
(3) \( R = \langle e \rangle 3 \); \hspace{1cm} R5: \langle e \rangle 3 \rightarrow (\langle e \rangle 4 * \langle e \rangle 5) \hspace{1cm} M(\langle e \rangle 3) = CN1
\hspace{1cm} code(\langle e \rangle 3) = code(\langle e \rangle 4)
\hspace{1cm} code(\langle e \rangle 5)
\hspace{1cm} COPY AX, M(\langle e \rangle 4)
\hspace{1cm} MUL AX, M(\langle e \rangle 5)
\hspace{1cm} COPY M(\langle e \rangle 3), AX

MEANING
\hspace{1cm} code(\langle s \rangle 1) = code(\langle e \rangle 4)
\hspace{1cm} code(\langle e \rangle 5)
\hspace{1cm} COPY AX, M(\langle e \rangle 4)
\hspace{1cm} MUL AX, M(\langle e \rangle 5)
\hspace{1cm} COPY CN1, AX
\hspace{1cm} COPY AX, CN1
\hspace{1cm} COPY R, AX

(4) \( R = (\langle e \rangle 4 * \langle e \rangle 5) \); R2: \langle e \rangle 4 \rightarrow \langle i \rangle 6 \hspace{1cm} M(\langle e \rangle 4) = M(\langle i \rangle 6)
\hspace{1cm} code(\langle e \rangle 4) = //nothing//

MEANING
\hspace{1cm} code(\langle s \rangle 1) = //nothing//
\hspace{1cm} code(\langle e \rangle 5)
\hspace{1cm} COPY AX, M(\langle i \rangle 6)
\hspace{1cm} MUL AX, M(\langle e \rangle 5)
\hspace{1cm} COPY CN1, AX
\hspace{1cm} COPY AX, CN1
\hspace{1cm} COPY R, AX
V(2) Add the following binary numbers and give their sum in binary:

\[
\begin{array}{cccccccc}
0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\
+ & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\
\hline
& & & & & & & & 1 & 0 & 1 & 1 & 0 & 1 & 1
\end{array}
\]

W 1.(6) For the truth table shown below, draw an equivalent circuit to the right showing switches in either the open or closed position. Show the appropriate battery and light. Label the switches correctly.

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

2. (3) Give the equivalent logical (Boolean) expression:

X(12) True/False?

(1) A computer program that aids in plant identification by using rules given to it by a expert botanists is a good example of an expert system.

(2) Computers are not nearly as good as humans when it comes to activities like translation of computer languages.

(3) Semantic networks are used in artificial intelligence programs to organize information about an object.

(4) Artificial Intelligence avoids the hard problems and focuses on the easy ones such as fast sorting.

(5) Computer game playing is considered to be a branch of artificial intelligence.

(6) Based on recent tournaments, the best human chess players are almost as good as the best computer chess programs.
Y(10) True/False?

(1) Light travels about one foot in a nanosecond.
(2) Relays produce a lot of heat for the function they perform.
(3) The mean time to failure is greater for transistors than for relays.
(4) Vacuum tube technology is still used in many computer displays.
(5) Integrated circuits can have over one million transistors on one chip.

Z(12) True/False?

(1) Many of the modern user interface ideas originally came from Xerox.
(2) Deadlocks can occur when computer resources are shared.
(3) Most modern operating systems no longer support multi-programming.
(4) Virtual memory is a way of simulating extra memory by using disks.
(5) Cache is used to keep frequently used information closer to the CPU.
(6) The earliest operating systems tried to provide a good graphical user interface (now called a GUI).