DO NOT OPEN THIS EXAM UNTIL YOU ARE READY TO TAKE IT

PRINT your name here

Do not discuss the exam with, or accept help from, anyone. You must write and sign this statement:

“This examination represents my own work in accordance with University regulations.”

Rules
This examination is open-book and open-note:

• You may use the textbook, course notes, your own notes, any documents from the web that you printed prior to starting the exam, corrected problem sets and solutions, lab instructions, etc.
• You may use a calculator.
• You may not use anything else; specifically, you may not use a computer, phone or tablet (except that you can use the calculator program on one of these, and you can use your computer to view course notes if you did not print them).

Procedure
DO NOT OPEN THIS EXAM UNTIL YOU ARE READY TO TAKE IT
This is a 90-minute exam that you must complete in a single 90-minute period any time before it is due. Set aside a comfortable time when you will be awake, where you will not be disturbed, and where you have all your course material at hand. Then open the exam and do it. After 90 minutes, close it and turn it in as soon as possible. Make sure that all pages are firmly attached.
There are a total of 90 points for the questions; use the point values for each question to allocate your time appropriately (one point per minute). Note that you only need answer 5 of the 7 parts of Question 5.
Write your answers directly on these pages; if you need more space, attach extra pages (stapled) and make sure your name is on any extra pages you submit. Please write neatly -- we can't grade it if we can't read it. It's quite all right to be brief as long as you're clear. We have tried to leave plenty of room for answers; if you are writing or computing a lot, you may be off on the wrong track.
Good luck.

Submission
Due by 7:00 PM, Friday, October 30, in the box outside Room 419 of the Computer Science building. Alternatively, Yichen will be in the lobby of the Friend Center (outside the library) to receive exams from 9-10AM on Saturday October 21.
Please do not discuss the exam with anyone until after the submission deadline has passed.
Problem 1 (15 points) WarmUp

My iPhone 6 has 55.5 GB of storage space. At present 28.1 GB are being used by applications, music, videos, photos. The rest is available. A typical movie requires 700 MB of storage at standard definition (SD). At high definition (HD), storing the same movie would require twice as much storage.

Do your calculations on the next page and record your answers here.

a) How many HD movies could I add to my iPhone?

27.4 GB of memory available; 1.4 GB per HD movie
27.4/1.4 = 19.5714, so 19 movies

b) Explain what Moore’s law is and make a reasonable assumption about the effect of Moore’s law on the storage capacity of the iPhone of the future.

Moore’s Law essentially says that the capacity of devices doubles every 18 months. So, applying this to the iPhone means that in 18 months, a version of my iPhone would have 111 GB of storage space and in 36 months, it would have 222 GB, …

c) My usage of iPhone space grows by about 12% per year. Using your answer to part b), how many movies at SD will I be able to store on my iPhone in 6 years?

By the rule of 72, growth of 12% per year means that in 6 years, my usage of the iPhone will have double from 28.1 GB to 56.2 GB. Meanwhile, Moore’s law says that the capacity of my iPhone will have gone through 4 doublings since 6 years is 4 x 18 months. So, the capacity of my iPhone will be 16 (=2^4) times as large or 16x55.5 which is 888 GB. This will leave 831.8 (=888-56.2) GB free to store movies. An SD movie is 700 MB and so, my iPhone in 6 years will be able to store 831.8/.7 = 1188 SD movies.
2. (20 points, 2 each) Short Answers. Circle the right answer or write it in the space provided.

(a) Unix systems distinguish upper case letters from lower case letters in filenames: lab3.html, Lab3.html and LAB3.HTML are different names. How many different ways are there to write the filename lab3.html in mixtures of upper and lower case letters?

Each of the 7 letters l,a,b,h,t,m,l can be either upper case or lower case. So, there are 2 options for each of 7 letters or $2^7 = 128$ possibilities.

(b) A version of Unicode represents characters by 2 bytes. Of the possible byte pairs that can represent a character, 256 are taken up with ASCII codes. Assume that another 4000 are taken up with Kanji codes and another 2000 are used for characters from other alphabets. There is a recent proposal to dedicate some 100 of the remaining characters to representing emojis. If this were to happen, what fraction of the remaining locations in the Unicode table would be used for emojis?

If Unicode is 2 bytes (= 16 bits) long, then it can represent $2^{16} = 65,536$ different characters. Of these, 256 are used by ASCII and another 6000 (=4000 for Kanji and 2000 for other alphabets) are used. This leaves 59,280 locations ($=65,536 – (256 + 4000 + 2000)$) that can be used. If 100 of these are used for emojis, this represents $100/59280 = .0016$ or .16% of the available space.

(c) On the side of the Computer Science building, a handful of bricks are reoriented to send a message to the campus. How is the message encoded and what does it say?

The message on the side of the computer science building is the ASCII representation for P=NP?

(d) Give an example of a problem that requires exponential time for its solution.

The Towers of Hanoi problem where we move disks of increasing size from one rod to another rod with the availability of an intermediate rod requires exponential time under the restriction that no disks can be placed on a disk of smaller size.

(e) What does the operating system do if it runs out of RAM when operating?

If the operating system runs out of RAM, it uses the disk for additional RAM. This memory is referred to as virtual memory. The process of moving data back and forth to disk is referred to as paging.

(f) The Minivac 601 (pictured above) was designed in 1961 to be the first personal computer. It sold for $85 and had the equivalent of 5 bits of memory. Accounting for inflation, the cost of the Minivac is about the same as the cost of a laptop a Princeton freshman might buy. Based on Moore’s Law, would
we expect the Minivac to have grown into a machine with more, less or the same amount of memory as the laptop a Princeton student bought?

The Minivac had 5 bits of memory 54 years ago. That represents 36 doublings under Moore’s law and so the Minivac of the present would have \(5 \times 2^{36}\) bits of memory which is \(5 \times 2^{33}\) gigabytes of memory. This amounts to 40GB of memory which is 4-5 times as large as the memory in a typical student’s laptop. So, the Minivac projected to the present would have more memory than the laptop a Princeton student would buy.

(g) What is the hexadecimal representation for each of the (base 10) numbers 27, 156, 256?

27 is 1B, 156 is 9C and 256 is 100

(h) The toy machine we programmed had one accumulator in which all computation took place. In a larger machine, how many accumulators might there be and what are they called?

In a larger machine, the accumulator would be replaced by a group of registers. Typical machines have 16 registers.

(i) If your computer has 8GB of RAM, how many transistors are used to create that amount of RAM

a. Less than 1 billion
b. 1-10 billion
c. 10-100 billion
d. 100 billion – 1 trillion

8GB of RAM is about 8 billion bytes or 64 billion bits of memory. The bit of memory we saw in class required 6 transistors for its storage. So, 64 billion times 6 is 384 billion which means answer d. is correct.

(j) Computer pioneer Grace Hopper (1906-1992) said, “The instruction code should use symbols which are easily learned and identified with the operations by already existing mental associations: 'a' for add, etc.’ Replacing a sequence of binary numbers with a single letter to represent an operation [simplifies] the coding process and makes it much more intuitive for users.” What kind or level of programming language is Hopper describing? One or two words should be enough.

Grace Murray Hopper is describing assembly language programming as an improvement over machine language programming.
3. (15 points) Machines

Here is a program in the Toy assembly language, with reminders about what the instructions do.

```
GET                  get a number from keyboard into accumulator
STORE INP            store accumulator in location INP
Foo IFPOS Bar        if accumulator is positive, go to Bar
LOAD SUM             print contents of accumulator
PRINT                add the value of SUM to the accumulator
STOP                 store the accumulator in location SUM

Bar SUB 1            subtract 1 from the accumulator
SUB 1                subtract 1 from the accumulator
STORE INP            store accumulator in location INP
ADD SUM              subtract 1 from the accumulator
LOAD SUM             store the value of INP into the accumulator
GOTO Foo             load the value of SUM into the accumulator
INP SUM              go to instruction labeled Foo
0                    reserve a memory location called INP
                      reserve a memory location called SUM, set its
initial value to 0
```

(a) What does the program print when the input is 4? What if the input is 7?

If the input is 4, the program prints 2 which is the sum of 2+0, the numbers it counts down through. If the input is 7, the program prints 8 which is the sum of 5+3+1+(-1), the numbers it counts down through.

(b) Describe in words what the program does.

If the input is N, the program counts down by 2 from N and starts adding values that it sees beginning at N-2 and continuing until it sees a non-positive number. It will add the last number it sees (which will be 0 or -1) to the sum it is computing and then print out the sum.

(c) What is a von Neumann architecture? Give one advantage and one disadvantage of such an architecture.

A von Neumann architecture is a computer architecture where data and program are stored in the same memory. An advantage of this is the added flexibility that comes from being able to use RAM for either program or data as the operating system runs rather than designating one area for program memory and one for data storage. A disadvantage is that there is risk of program code overrunning data or vice versa.
4. (15 points) State machines

Design a state machine to operate a toll booth. The toll booth will permit cars and trucks to pass. The toll for cars is $1 and for trucks, it is $2. The booth only takes 1 dollar bills. You can assume that no other bills are ever inserted.

When a vehicle arrives, it generates an input of either C (for car) or T (for truck). When money is inserted, an input of M is generated. So, your input symbols are (C,T,M).

When the proper toll for the vehicle has been inserted, an output of O (for open toll gate) is generated and the state machine resets to its initial state.

While a car or truck is being processed, you can assume that no further cars or trucks arrive. So, once an input symbol of C or T has been read, there will be no input of C and T until either the gate is opened (so, O is output) or the machine enters its dead state.

For simplicity, you can add a dead state to your state machine. If the state machine ever receives inappropriate input, it outputs E (for error) and goes to the dead state. Once in the dead state, it’s response to all input is to generate output E and stay in the dead state. It can only escape the dead state by human intervention and this is not part of your state machine.

Note that the state machine only has the memory provided by its states and so we need to begin by distinguishing between the arrival of a car and the arrival of a truck. Once we have done so, we then proceed to accept toll money (once for a car and twice for a truck) at which point we open the gate. In the process of opening the gate, we return to the start state and wait for the next vehicle. Any incorrect input (e.g. money before there is a car or truck or a car or truck while we are waiting for money) sends the machine to its dead state where it awaits human intervention.
5. (25 points, 5 each) Miscellaneous
You need answer only 5 of these 7 questions!!

(a) When we discussed sort/merge and quicksort in class, we described them as using recursion or being divide-and-conquer algorithms. What do these words mean? Why is selection sort not a divide-and-conquer algorithm?

A divide-and-conquer algorithm is an algorithm where we solve a problem by dividing it in 2 and solving each half and then combining the two results to get a final solution. We then solve the problem of each half by dividing it in 2 and then combining the two results to get a solution. This division continues until we have trivial problems to solve. Quicksort and Sort/Merge are examples of divide and conquer (or recursive) algorithms. Selection sort is not because selection sort operates by comparing all pairs of numbers in the input. There is never a point where the input is subdivided into smaller pieces.

(b) While the doublings predicted by Moore’s Law have lasted longer than anticipated, they cannot go on forever. True or False? Explain your answer

True. Moore’s Law is facilitated by making wires thinner and thinner which allows circuits to operate in a smaller area and so switch faster. An effect of this is that fewer electrons flow in a smaller wire. As the number of electrons decrease, other effects come into play. Most importantly, we cannot subdivide an electron and so we cannot get beyond the stage where only single electrons can flow in the wires.

(c) For the function (NOT (A AND B)) OR (NOT C)

(i) Draw the circuitry (in terms of basic AND, OR, NOT gates) that this represents

(ii) Give a truth table for the function
(d) In this problem, we consider colors represented by the RGB triple described in class with one byte for each pixel. (N.B. I intended for this to be one byte for each color in each pixel which is the answer given below. Students who interpreted it as one byte per pixel were given credit for their work because of my mistake in setting the question.)

(i) How many memory is needed to store my screen which is 1280x1024?

1280x1024x3 is 3932160 bytes of memory to store my screen (or slightly less than 4MB).

(ii) Would it be possible for my screen to be of a size so that not every pixel could have a unique color? If so, what would be a screen resolution that would lead to duplicate pixel values? If not, explain why not.

24 bits of color representation amount to $2^{24} \approx 16$ million different colors. For the screen to have high enough resolution so that it was not possible for each pixel to have a unique color, it would have to be at least $4096x4096$ ($2^{12}x2^{12}$) in resolution.

(e) Why do students in COS 109 typically not run UNIX or Linux on their personal computers but companies like Microsoft and Apple use Linux to support their web sites?

Linux/UNIX are designed to operate well in environments where they are controlling large systems such as the Microsoft website or the iTunes store. But they do not present an elegant window manager and user interface that is attractive to COS 109 students.

(f) Experience shows that enrollment in MOOCs (massive open online courses) drops off rapidly; a typical course loses half its remaining students every week. Suppose a MOOC starts with one million online students..
(i) What percentage of the remaining students drop the course every day?
If half the students drop the course each day, then by the rule of 72, \( \frac{72}{7} = 10.28\% \) of the students drop the course each day.

(iii) How many students will be left by the end of a 12-week semester?
After 12 weeks, the number of students remaining will be \( \frac{1000000}{2^{12}} = 244 \).

(g) When we discussed memory management in the operating system, we talked about how the RAM used by an individual program could get fragmented as more memory was needed and memory was released. Give a simple example of this in action by having programs start and stop, request more memory and free memory. Your example does not need to go through more than about 5 stages.