UCLA Computer Science 35L final exam - Winter 2020

Name:_________________________________

Student ID:___________________________

100 points total. Open book, open notes, open computer. Answer all questions yourself, without assistance from other students or outsiders. Although this exam is designed to take 180 minutes, under the circumstances it is simply a takehome exam.

Print this exam, write your answers on it, scan the completed exam, and upload your scans to CCLE Gradescope by 15:00 Monday (Los Angeles time). If you do not have easy access to a scanner, carefully photograph the sheets of paper with your cell phone and upload the photographs. Save your filled-out exam and do not give or show it to anybody other than an instructor or TA; we will send you instructions later about what to do with your filled-out exam.

If you lack access to a printer, read the exam on your laptop’s screen, write your answers on blank sheets of paper (preferably 8½″×11″) with one page per question, and upload the scanned sheets of paper. Please answer every question on a new sheet of paper. At the end of the exam, you should have scanned and uploaded as many photographs as there are questions. If you do not answer a question, scan a blank sheet of paper as the answer.

As previously announced, the exam is open book and open notes, but due to circumstances we are also making it open computer. You can use your laptop to use a search engine for answers, and to run programs designed to help you answer questions. However, do not use your computer or any other method to communicate with other students or outsiders, or anything like that. Communicate only via CCLE and Gradescope to obtain your exam and upload your scanned results, or via Zoom or email with the instructor or TAs.

IMPORTANT Before submitting the exam, certify that you have read and abided by the above rules by signing and dating here:

Signature:_________________________________ Date:__________

1 (1 point). Every directory in Linux is initialized with at least two other directories inside it. What are these two directories and why are they there?

2 (4 points). Consider an inode that we’ll call inode A, with a link count of 3. This inode holds information about the regular file ‘foo’. Consider another inode B also with a link count of 3 that holds information about the directory ‘bar’. Answer the following questions about inode A as if they were executed sequentially; e.g., if the link count increases from 3 to 6 in (1a), use the new link count of 6 for (1b).

2a. What happens to the link count for A when we create two new hard links (h1 and h2) to foo?

2b. What happens to the link count for A when we create a symbolic link s1 to ‘bar’ and then a symbolic link s2 to s1?

2c. Is it possible to create a symbolic link inside the directory ‘bar’ that causes a cycle? If so, write a command to create a cycle; if not, explain why not.

2d. What happens to link s1 when the directory ‘bar’ is removed?

3 (2 points). Consider the following Bash script:

```
#!/bin/bash
a=2
sed 's/2'${a}'}/f/g' < input.txt
```

Assuming the script is executable and that input.txt exists and is non-empty, what does this script do?

4 (2 points). Given an existing non-empty text file, myfile, what does the following command do?

```
ls -A | grep '^[.].*' >> myfile 2>&1
```

5 (4 points). Assume a Bash script named foo exists and is executable. Also assume foo has the following structure:

```
  #! /bin/bash
  echo ______
  echo ______
  echo ______
  echo "line 4: ca${4}ts"
```

where the arguments of the first three echo commands have been left blank. Assume the following shell command is run in the directory where foo exists:

```
  ./foo I love 'eating rro'
```

The resulting stdout output is as follows:

```
  line 1: I
  line 2: $2
  line 3: I love eating rro
  line 4: cats
```

Complete the echo statements to have the script produce the given stdout output. Make sure that the script can run with arbitrary arguments. For example:

```
  ./foo John Smith loves rro
```

should produce the following stdout output:

```
  line 1: John
  line 2: $2
  line 3: John Smith loves rro
  line 4: carrots
```

7 (8 points). Suppose you receive a comma-separated values (CSV) file “locations.csv”, containing a list of locations and their coordinates. Here are the first 5 lines in the file:

```
  id,location
  1,"In-N-Out"
  2,"Salt & Straw"
  3,"Rocco's Tavern"
  4,"Tatsu Ramen"
```

The first row contains the column names separated by commas. Each successive row contains a unique data entry.

A correctly formatted data line follows these rules:

* Exactly two fields separated by commas, i.e., each line has exactly one comma.

* The first field (i.e., id) is a non-empty string of ASCII digits.

* The second field (i.e., location) starts and ends with double quotation marks ("), and may use uppercase/lowercase ASCII letters, spaces ( ), hyphen-minuses (-), ampersands (&), or apostrophes (').

You notice after looking through the file that some entries don’t follow the format listed above, and you’d like to remove them. Write a shell command that reads the file from stdin and prints only correctly formatted lines. Do not include the header line (“id,location”) in the output of correctly formatted lines.

Problems 8 through 10 use the Python 3 class below:

```
  class FruitSalad:
    def __init__(self, fruits):
      self.fruits = fruits

    def add_fruits(self, new_fruits):
      self.fruits += new_fruits
      new_fruits = []

  if __name__ == "__main__":
    x = ['apple', 'banana']
    y = ['strawberry', 'watermelon']

    salad = FruitSalad(x)
    salad.add_fruits(y)
```

8 (3 points). What is the value of the list y after calling salad.add_fruits()? Explain your reasoning in a couple sentences.

9 (3 points). Suppose we add a class variable melons to FruitSalad:

```
  class FruitSalad:
    melons = ['watermelon', 'honeydew', 'cantaloupe']
    ...
```

Write an instance method find_melons() that returns the index of every
melon from melons that is found in self.fruits. The return value
should be a Python dictionary mapping the melon (a string) to the
index. For example, running find_melons() on the prior code should
return { 'watermelon': 3 }

10 (3 points). Write an instance method remove_melons() that returns a copy of self.fruits with all the melons removed. The original self.fruits should not be modified. You can use your implementation of find_melons(). For example, running remove_melons() on the prior code should return ['apple', 'banana', 'strawberry'].

11 (2 points). What's the problem with the following C code?
Fix the problem in a way that best matches the evident intent.

```
  #include <stdio.h>

  typedef struct {
      int x;
      int y;
  } Point;

  Point *create_point() {
      Point p;
      p.x = 0;
      p.y = 0;
      return &p;
  }

  int main() {
      Point *p = create_point();
      printf("(x, y) = (%d, %d)\n", p->x, p->y);
      return 0;
  }
```

12. Suppose we are worried that attackers have taken control of the SEASnet network (but not the SEASnet servers or anything else) and that the attackers want to prevent this final exam from being conducted fairly. We decide to conduct this exam via ssh in order to defeat the attackers, as follows. Students use ssh to get a copy of the exam from /home/eggert/cs35L/final.pdf, print it out, take the exam, scan the result into a file named 123-456-789.pdf (where 123-456-789 is their student ID), and then upload the file into /home/eggert/cs35L/submit/123-456-789.pdf and the professor and TAs take it from there.

12a (1 point). Describe a simple way the attackers can defeat this approach and prevent the exam from being conducted fairly anyway.

12b (1 point). Suppose the attackers are passive, i.e., they do not alter how the SEASnet network behaves, but can still get copies of all the data sent across the network. What information can they easily learn about the exam and how it was conducted? What information will it be hard for them to discover?

12c (2 points). Changing the subject slightly, does it ever make sense to use SSH's 'scp' command to copy a file directly from one SEASnet GNU/Linux host to another? If so, explain why; if not, explain why not.

16 (10 points total). Imagine you are on a team of 3 software engineers. You are working together to create a new web application which will require 3 different python code files: server.py, config.py, data.py.

Your team plans for 10 distinct ‘features’ that will eventually need to be created for the application to be usable.

16a (1 point). This project will require version control. Say in one line what tool(s) would you use to enforce version-control.

16b (5 points). Explain in at most half a page what kind of workflow each software-engineer should follow. Below are potential discussion points, and try to use at least 3 points in your argument. (Your 3 points don’t need to come from this list)

 * Should engineers use branches, and if so - then how?
 * If you plan to merge commits, how should merge conflicts be handled?
 * If you plan to merge commits, will your team prefer ‘git merge’ or ‘git rebase’?
 * Will you use a ‘centralized’ remote repo or not?
 * How should engineers make sure their work is in sync?

16c (1 point). Explain in 2-3 sentences any concerns about your solution.

16d (3 points). Assume you are in the middle of the same project. You were given the task to create the feature ‘Add new URL path to server.py’. What git commands would you use to emulate the workflow you described earlier? You can fill in your answer in a format like below:

```
        # Any pre-coding Git commands here

        // You make and test your changes to server.py.  (You don’t need
        // to write anything else here).

        # Any post-coding Git commands here.
```

17 (2 points). When you run the command ‘git log’ on one branch, e.g., ‘git log master’, is it possible that one of the commits is not the parent of the commit that comes right before it in the output? If it’s not possible, explain why. If it’s possible, give a scenario where this can happen.

18 (4 points). Assume you have a program G that prints out a directed edge for every pair of parent and child commit hashes in a Git repository, where an edge will look like

```
  h_parent -> h_child
```

where h_parent is the hash of the parent commit, h_child is the hash of the child commit.

Someone claims that they have a program P that outputs all the commit hashes in a topological order in which child commits are printed before ancestral commits.

Describe how can you use the output of program G to verify that the output of program P is a valid topological order. There is no need to write actual code.

19 (10 points). Consider the following text taken from:
Guzdial M. Beware of Hurting Our Weakest Students when Moving Classes Online. Blog@CACM. 2020-03-10.

> The biggest cost of moving our classes online will likely be the decreased learning and lower grades, particularly of our weakest students. Fully on-line classes can lead to less learning than face-to-face classes (see 2017 review paper here). Online learning has a differential impact on students. A 2013 paper studying 40,000 students found that students who come in with lower grades suffer the most in performance when moving online (see paper here). A 2018 New York Times article (see link here) describes about how online classes hurt students who most need help. Justin Reich (MIT) has written a great Twitter thread about all the evidence showing that moving classes online will hurt the most vulnerable students -- start here. His recommendation is to simply cancel classes until June or September. That would be better than creating greater disparity and inequity in our classes.

Discuss the relevance of the technology that you covered in your Assignment 10 presentation as a possible way of addressing the problems that Guzdial raises. If your Assignment 10 technology is largely irrelevant, state why it’s irrelevant and discuss how you’d address the problems by using the Zoom features that have been employed during the past week at UCLA.