

20W-COMSCI35L-1 Final exam

GEORGE OWEN

TOTAL POINTS

81.25 / 100

QUESTION 1

1 Linux - directories 1 / 1

✓ + 0.5 pts . is correct

✓ + 0.5 pts .. is correct

+ 0 pts Click here to replace this description.

QUESTION 2

Linux - hard links & symbolic links 4 pts

2.1 link count, hard links 1 / 1

✓ + 1 pts Increases to 5.

+ 0 pts incorrect

2.2 link count, symlink 1 / 1

✓ + 1 pts Nothing happens to link count.

+ 0 pts Incorrect answer.

2.3 link cycle 1 / 1

✓ + 1 pts Fully correct; ln -s . [some_name] OR ln -s

bar s1

+ 0.5 pts Cycle is possible, but incorrect command.

+ 0 pts Incorrect.

2.4 remove bar 1 / 1

✓ + 1 pts Link s1 is left broken/dangling.

+ 0 pts Incorrect

QUESTION 3

3 Bash - sed script 2 / 2

✓ - 0 pts Correct

QUESTION 4

4 Bash - ls 2 / 2

✓ - 0 pts Correct

QUESTION 5

5 Bash - echo 4 / 4

✓ - 0 pts Correct

QUESTION 6

6 'AB' regex 8 / 8

✓ - 0 pts Correct

QUESTION 7

7 CSV - output correctly formatted lines 6 / 8

✓ + 1 pts Use grep and/or appropriate shell

commands

✓ + 0.5 pts Use ^ correctly

✓ + 0.5 pts Use \$ correctly

✓ + 1 pts Correct comma detection

id field, e.g. [0-9]+

✓ + 2 pts Correct

+ 1 pts Partially correct

+ 0 pts Incorrect (or barely correct)

location field, e.g. "[A-Za-z '&-]*"

+ 3 pts Correct

+ 2 pts 1 character off

✓ + 1 pts Partially correct

+ 0 pts Incorrect (or barely correct)

Misc. error (no -E for ERE, improper quotations, additional whitespace etc.)

+ 0 pts No error

- 1 pts Single error

- 2 pts Multiple errors

+ 0 pts No answer

QUESTION 8

8 Python - call add_fruits() 3 / 3

✓ - 0 pts Correct

QUESTION 9

9 Python - add class var melons 3 / 3

✓ - 0 pts Correct

QUESTION 10

10 Python - write remove_melons() 3 / 3

✓ - 0 pts Correct

QUESTION 11

11 C - problem in Point code 2 / 2

✓ + 1 pts mention about dangling pointer or pointer

returned by create_point() going out of scope

✓ + 1 pts Suggesting a fix for it.

+ 0 pts Click here to replace this description.

QUESTION 12

SSH 4 pts

12.1 attackers defeat this approach 1 / 1

✓ + 1 pts Correct

+ 0.5 pts May be possible, but missing some description

+ 0.5 pts Intercept - limited because still can't review/modify encrypted traffic.

+ 0 pts Incorrect

12.2 passive attackers 0.75 / 1

+ 1 pts 2 valid things they can easily learn and it is hard to learn encrypted file contents

✓ + 0.75 pts 2 out of 3 above

+ 0.5 pts Communication is encrypted

+ 0 pts Incorrect

12.3 using scp 2 / 2

✓ + 2 pts Clearly explain a valid reason. Either server specific files like /etc/passwd need to be copied securely with scp, or all other files are already synced to remote file server making scp unnecessary

+ 1.5 pts Uses justification that SCP is secure

+ 1 pts On the right track, but explanation is missing details. Or description not applicable to SEAS

environment

+ 0 pts Incorrect or Unclear

QUESTION 13

C - writing a grep 15 pts

13.1 implement valid_regex 1 / 1

✓ - 0 pts correct

13.2 implement add_char 1 / 2

✓ - 1 pts * has higher precedence than +

13.3 implement match 10 / 12

logic error group

✓ - 2 pts small logic error

QUESTION 14

14 C - naive getchar() 5 / 5

✓ + 2 pts call the read function correctly

✓ + 2 pts check the output of read correctly

✓ + 1 pts Correct overall logic

+ 0 pts Click here to replace this description.

QUESTION 15

Makefiles 10 pts

15.1 main.c 3.5 / 4

✓ - 0.5 pts one extra/less include

15.2 Makefile 4 / 6

+ 6 pts Correct

✓ + 1 pts addchar.o, subchar.o

✓ + 1 pts libcomputechar.a

✓ + 2 pts *.so (.c + .a)

+ 2 pts main (no .a)

+ 0 pts Incorrect

+ 5 pts No prereq; used computechar.h

+ 1 pts no correct rules, but general flow is ok

QUESTION 16

Version Control 10 pts

16.1 VC tool 1 / 1

+ 0 pts No answer

✓ + 1 pts Name a valid VC Tool

+ 0 pts Incorrect

16.2 VC workflow 3 / 5

✓ + 3 pts Discuss at least 3 points in argument

+ 1 pts Clarity

+ 1 pts Logical Workflow

+ 1 pts Explanation missing some steps/unclear

16.3 VC solution concerns 1 / 1

✓ + 1 pts State at least 1 valid concern based on workflow above

+ 0 pts Unclear concern

16.4 VC git commands 2 / 3

✓ + 1 pts Add and commit changes to server.py

+ 2 pts Commands match all workflow above

✓ + 1 pts Commands match partial workflow

+ 0 pts Incorrect

QUESTION 17

17 Git - git log commit output 2 / 2

✓ - 0 pts Correct

QUESTION 18

18 Git - topo order validation 0 / 4

✓ - 4 pts Incorrect

QUESTION 19

19 Writing - technology relevance 6 / 10

+ 10 pts Checked all boxes

✓ + 2 pts Mentioned 1 feature that can help tackle the given problems.

+ 4 pts Mentioned 2 or more features that can help tackle the given problems. Alternatively, covered 1 feature in great detail.

+ 1 pts Semi-clear writing

✓ + 2 pts Clarity of writing

✓ + 2 pts Discussed student's own presentation

+ 1 pts Semi-persuasive

+ 2 pts Persuasiveness

Name: George Owen
Student ID: 405-196-467

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 $8\frac{1}{2}'' \times 11''$) 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: George Owen

Date: 3/15/20

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?

These directories are ":" and ".." - ":" points to the ~~current~~ current directory, while ".." points to the parent of the current directory. These directories allow humans and programs to effectively navigate the file tree.

1 Linux - directories 1 / 1

✓ + **0.5 pts** . is correct

✓ + **0.5 pts** .. is correct

+ **0 pts** Click here to replace this description.

count of 6 for (b).

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

The link count will increase by 2, totaling 5

2.1 link count, hard links 1 / 1

✓ + 1 pts Increases to 5.

+ 0 pts incorrect

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?

The link count for A will not change, as 'bar' information is contained within inode B, which is unrelated to A.

Additionally, the link count for B will not change either, as symbolic links do not affect the link count

3. Is it possible to create a symbolic link inside the directory 'bar'

2.2 link count, symlink 1 / 1

✓ + 1 pts Nothing happens to link count.

+ 0 pts Incorrect answer.

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.

Yes!



the first command doesn't work if bar moves in the filesystem, while the second command does

`ln -s $(pwd)/bar bar/cycle`

or from inside the directory bar:

`ln -s . cycle`

`ln -s . bar/cycle`

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

2.3 link cycle 1 / 1

✓ + **1 pts** Fully correct; `ln -s . [some_name]` OR `ln -s bar s1`

+ **0.5 pts** Cycle is possible, but incorrect command.

+ **0 pts** Incorrect.

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

The symbolic link s1 becomes unusable, as it is simply a pointer to the original directory "bar"

"no such file or directory"

2.4 remove bar 1/1

✓ + 1 pts Link s1 is left broken/dangling.

+ 0 pts Incorrect

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?

's/2'2'3/f/g'

s/22}/f/g

This ~~script~~ uses the stream editor `sed` to write the contents of "`input.txt`" to `stdout`. In this output, all instances of the string "22}" are replaced with "f"

input.txt

Press 22} to pay respects

22}

22} 22} 22 22}}

output

Press f to pay respects

f

ff 22f}

3 Bash - sed script 2 / 2

✓ - 0 pts Correct

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

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

This command appends the names of all files present in the current working directory which start with a '.' (so ~~#~~ hidden files) to the file myfile. The names are appended after the existing contents of myfile, on a new line. Every subsequent file after the first is also on a new line (due to grep).

This command will not append the directories '.' and '..', as the -A option of ls ignores them.

Any errors during execution will be directed from stderr to stdout, which is being redirected to "myfile"

4 Bash - ls 2 / 2

✓ - 0 pts Correct

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:

1 2 3
./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:

1 2 3 4
./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
```

```
#!/bin/bash
echo "line 1: $1"
echo 'line 2: $2'
echo "line 3: $@"
echo "line 4: ca${4}ts"
```

rewritten for
legibility

5 Bash - echo 4 / 4

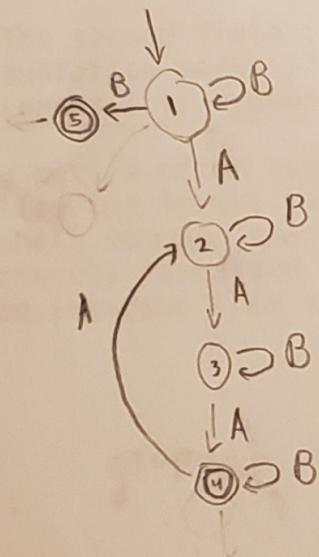
✓ - 0 pts Correct

6 (8 points). Write an extended regular expression that matches non-empty strings that consist of only characters A and B, in which the number of As is 3n, where n is a non-negative integer. You will get 4 points if your solution doesn't exclude the empty string but is otherwise correct. These are example matches, one example per line:

B
BAAA
AABBA
BBBBB
ABABAB
AAAAAA

$\wedge ((B + (AB^*AB^*AB^*)^*) | (B^*(AB^*AB^*AB^*)^+)) \$$

$\wedge (B + | (B^*(AB^*AB^*AB^*)^+)) \$$ ← shorter



$B (A)^*$ B & 0 or more As

$B^*(A)^+$ 0 or more Bs & 1 or more As

\wedge | $B^*(AB^*AB^*AB^*)^*$

everything, inc.
empty string

6 'AB' regex 8 / 8

✓ - 0 pts Correct

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"
```

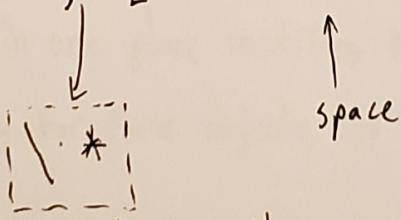
The first row contains the column names separated by commas. Each successive row contains a unique data entry.

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.

grep -E ^[0-9]+\,\\"[A-Za-z\-\&\']*\\"\$



append this after
 \, if there can optionally
 be a space separating the fields.
 This question doesn't specify that there
 is not a space present, and I cannot tell by looking

7 CSV - output correctly formatted lines 6 / 8

- ✓ + 1 pts Use grep and/or appropriate shell commands
- ✓ + 0.5 pts Use ^ correctly
- ✓ + 0.5 pts Use \$ correctly
- ✓ + 1 pts Correct comma detection

id field, e.g. [0-9]+

- ✓ + 2 pts Correct
- + 1 pts Partially correct
- + 0 pts Incorrect (or barely correct)

location field, e.g. \"[A-Za-z '&-\"]"

- + 3 pts Correct
- + 2 pts 1 character off
- ✓ + 1 pts Partially correct
- + 0 pts Incorrect (or barely correct)

Misc. error (no -E for ERE, improper quotations, additional whitespace etc.)

- + 0 pts No error
- 1 pts Single error
- 2 pts Multiple errors

- + 0 pts No answer

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 = []

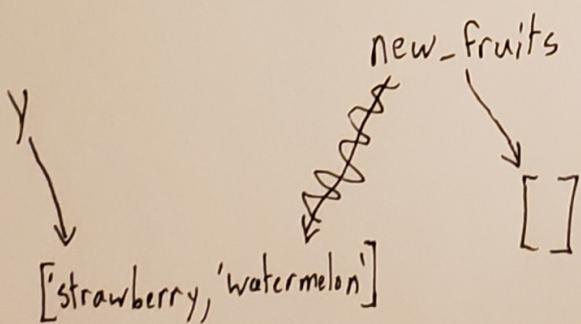
if __name__ == "__main__":
    x = ['apple', 'banana']
    y = ['strawberry', 'watermelon']

    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.

`y` is `['strawberry', 'watermelon']` after calling `add_fruits(y)`

The contents of `y` remain the same because python passes variables to function by object-reference. The contents of the list ~~exist at the~~ exist at only one location in memory, and modifying the contents in one place modifies them everywhere. However, variable names are simply pointers/labels for these objects, so modifying what a variable points to does not change



`new_fruits[:] = []`
empties the contents of
`x`

8 Python - call add_fruits() 3 / 3

✓ - 0 pts Correct

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

```
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 }

```
def find_melons(self):  
    melon_map = {}  
    for fruit in self.fruits:  
        if fruit in self.melons:  
            melon_map[fruit] = self.fruits.index(fruit)  
    return melon_map
```

9 Python - add class var melons 3 / 3

✓ - 0 pts Correct

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']`.

```
def remove_melons(self):
    new_fruits = []
    for fruit in self.fruits:
        if fruit not in self.melons:
            new_fruits.append(fruit)
    return new_fruits
```

10 Python - write remove_melons() 3 / 3

✓ - 0 pts Correct

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() remove this
Point p;
p.x = 0;
p.y = 0;
return &p;
```

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

Point this->p = {0, 0};
 Point *p = &this->p;

The `create_point` function ~~not~~ returns a reference to a local variable,
 meaning that `point` does not exist outside of the scope of the function.
 We fix this by inlining the ~~initialization~~ initialization.

11 C - problem in Point code 2 / 2

✓ + 1 pts mention about dangling pointer or pointer returned by `create_point()` going out of scope

✓ + 1 pts Suggesting a fix for it.

+ 0 pts Click here to replace this description.

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.

Since the attackers control the network, they can simply block the request from your computer to prevent you from getting or uploading the exam. They could also reroute your request so ~~they~~ you receive a different file.

12b (1 point). Suppose the attackers are passive i.e. they do not

12.1 attackers defeat this approach 1 / 1

✓ + 1 pts Correct

+ 0.5 pts May be possible, but missing some description

+ 0.5 pts Intercept - limited because still can't review/modify encrypted traffic.

+ 0 pts Incorrect

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?

They can see when you downloaded the exam, as well as when you submitted it.

It will be hard to see the exam contents
(impossible without the right key)

12.2 passive attackers 0.75 / 1

+ **1 pts** 2 valid things they can easily learn and it is hard to learn encrypted file contents

✓ + **0.75 pts** 2 out of 3 above

+ **0.5 pts** Communication is encrypted

+ **0 pts** Incorrect

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.

Not really. All ~~the~~ users and files are shared across the SEASnet servers already, so there is no need to use 'scp' to copy a file.

ex. Running: ~~scp from~~

scp target classone@lnxrv05.seas.ucla.edu:351/target
on lnxrv09~~.09~~. Still on lnxrv09, I can run:

cd 351

ls

> target

Files are synced!

12.3 Using scp 2 / 2

✓ + **2 pts** Clearly explain a valid reason. Either server specific files like /etc/passwd need to be copied securely with scp, or all other files are already synced to remote file server making scp unnecessary

+ **1.5 pts** Uses justification that SCP is secure

+ **1 pts** On the right track, but explanation is missing details. Or description not applicable to SEAS environment

+ **0 pts** Incorrect or Unclear

(continued from previous page)

13a (1 point). Implement the function 'valid_regex' that returns nonzero if its argument is a valid regular expression, zero otherwise.

```
int valid_regex(const char *regex)
/* Write your implementation here. */
```

```
{
    char *L = regex;
    while (*L != '\0')
    {
        L++;
    }
    L--;
    if (*L == '\\')
    {
        int okay = 1; // even is okay (every '\' has a corresponding '\\')
        // odd is not okay
        while (*(L - okay) == '\\') // count the number of consecutive '\' chars at the end
        {
            okay++;
        }
        if (okay % 2 != 0) { return 0; }
    }
    return 1;
}
```

Better solution

Valid : 'a\\', 'a\\\\'

invalid: 'a\\',
'a\\\\'

```
valid_regex(...)

{
    char *L = regex;
    int literal = 0;
    while (*L != '\0')
    {
        if (*L == '\\' && literal == 0) { literal = 1; }
        else { literal = 0; }
        L++;
    }
    return literal == 1 ? 0 : 1;
}
```

MUCH better way
to do the same thing. I figured this out after I wrote match()

13.1 implement valid_regex 1 / 1

✓ - 0 pts correct

(continued from previous page)

13b (3 points). Implement the function 'addchar' that adds a character to the dynamically allocated array. You only need to use realloc in this function.

```
char *addchar (char *line, size_t *nchars, char ch)
/* Write your implementation here. */
{
    line = (char *) realloc(line, ((*nchars)+1 * sizeof(char)));
    line[*nchars] = ch;
    (*nchars)++;
    return line;
}
```

13.2 implement add_char 1 / 2

✓ - 1 pts * has higher precedence than +

(continued from previous page)

13c (12 points). Implement the function 'match' that can match the line with the corresponding regex, and prints out the matching text inside the function. The function should always return NULL so that the line variable will get reset after matching. Please free all allocated memory inside the match function.

There are at least two approaches to implement this function. The first one would be storing the matching characters to a new dynamically allocated array and print out after iterating through the line. The other one would be having the start index and the end index, and then moving these indices to circle the range of text in the line that needs to be printed.

```
char *match (const char *regex, char *line)
/* Write your implementation here. */
```

```
{
    char *match(
    int length = 0;
    char *L = regex;
    int literal = 0;
    while (*L != '\0') {
        if (*L == '\\' && literal == 0) { literal = 1; }
        else { literal = 0; length++; }
        L++;
    }
    int start, matchLen = 0;
    for (start = 0; line[start + length - 1] != LF; start++)
    {
        matchLen = 0;
        literal = 0;
        int charNum = matchLen;
        while (matchLen < length) {
            if (regex[charNum] == '\\' && literal == 0) { literal = 1; }
            else if (regex[charNum] == ':' && literal == 0) { matchLen++; }
            else if (line[start + matchLen] == regex[charNum]) {
                literal = 0;
                matchLen++;
            }
            else { break; }
            charNum++;
        }
        // (cont. on back)
    }
}
```

```
    if (matchLen == length)
    {
        for (int i=0; i<matchLen; i++)
        {
            putchar(line[start]);
            start++;
        }
        putchar('\n');
        free(line);
        return NULL;
    }
}

Free(line);
return NULL;
```

13.3 implement match 10 / 12

logic error group

✓ - 2 pts small logic error

14 (5 points). Use the 'read' system call with buffer size 1 to write a naive implementation of getchar() with the following function signature taken from <stdio.h>:

```
#define EOF (-1)
/* Return a character read from stdin, or EOF if no character/error. */
int getchar(void)
{
    char c[1];
    int status = read(0, c, 1);
    if (status < 1)
    {
        return EOF;
    }
    return c[0];
}
```

14 C - naive getchar() 5 / 5

- ✓ + 2 pts call the read function correctly
- ✓ + 2 pts check the output of read correctly
- ✓ + 1 pts Correct overall logic

+ 0 pts Click here to replace this description.

(continued from previous page)

main.c (4 points):

It takes three arguments <arg1> <arg2> <arg3> where <arg1> is either "add" or "sub". Based on <arg1>, your main.c should dynamically link either libaddstr.so or libsubstr.so via dlopen(), dlsym(). Your main.c should output an integer value, which is the sum if <arg1> is "add" and the difference if <arg1> is "sub". <arg2> and <arg3> are two non-empty strings with the same lengths.

Your main.c should also handle errors correctly, including incorrect <arg>'s, dlopen() errors, dlsym() errors, and dlclose() errors. Exit with status 1 if any of the errors above occurred.

Examples:

./main add Com Sci # Outputs an integer.
 ./main sub Oh ok # Outputs an integer.
 ./main diff a b # Exits with status 1 because "diff" is not valid.

given more space, I would add
descriptive error messages at
each exit point

/* Put your complete implementation of main.c below. */

```
#include "computechar.h"
#include <dlfcn.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>

int main(int argc, char **argv) {
    if (argc != 4) { exit(1); }
    if (strcmp(argv[2], "add") || strcmp(argv[2], "sub")) { exit(1); }
    if (strlen(argv[2]) != strlen(argv[3])) { exit(1); }

    void *lib_handle;
    char *error;
    int (*operation)(char *, char *);

    if (!strcmp(argv[1], "add")) {
        lib_handle = dlopen("libaddstr.so", RTLD_LAZY);
        if (!lib_handle) { exit(1); }
        operation = dlsym(lib_handle, "addStr");
        error = dlerror();
    }
    else {
        lib_handle = dlopen("libsubstr.so", RTLD_LAZY);
        error = dlerror();
    }
    if (error) {
        printf("Error: %s\n", error);
        exit(1);
    }
    int result = operation(argv[2], argv[3]);
    printf("%d\n", result);
}
```

↓(on back)

```
↓  
if (!lib_handle) { exit(1); }  
operation = dlsym(lib_handle, "substr");  
error = dlerror();  
}  
if (error != NULL) { exit(1); }  
int output = operation(argv[2], argv[3]);  
int a = dlclose(lib_handle);  
if (a != 0) { exit(1); }  
printf("%d\n", output);  
return 0;  
}
```

15.1 main.c 3.5 / 4

✓ - 0.5 pts one extra/less include

(continued from previous page)

Makefile (6 points):

It should at least include rules to:

- (1). Compile addchar.c and subchar.c into an archive: libcomputechar.a.
- (2). Compile addstr.c and substr.c into dynamic libraries libaddstr.so and libsubstr.so.
- (3). Compile main.c into an executable 'main'.

You can assume all files are under the same directory.

Make sure each module includes the minimal set of include files.

Put your complete implementation of Makefile below.

default: libcomputechar.a libaddstr.so libsubstr.so main

addchar.o: addchar.c ~~computechar.h~~

 gcc -c addchar.c

substr.o: substr.c

 gcc -c substr.c

libcomputechar.a: addchar.o substr.o

 ar -cvq libcomputechar.a addchar.o substr.o

libaddstr.so: addstr.c ~~computechar.h~~ libcomputechar.a

 gcc addstr.c -O libaddstr.so -fPIC -shared -lcomputechar -L.

libsubstr.so: substr.c ~~computechar.h~~ libcomputechar.a

 gcc substr.c -O libsubstr.so -fPIC -shared -lcomputechar -L.

main: main.c computechar.h

 gcc main.c -O main -ldl -Wl,-rpath=\$(PWD)

clean:

 rm -f *.o *.so *.a main

15.2 Makefile 4 / 6

+ 6 pts Correct

✓ + 1 pts addchar.o, subchar.o

✓ + 1 pts libcomputechar.a

✓ + 2 pts *.so (.c + .a)

+ 2 pts main (no .a)

+ 0 pts Incorrect

+ 5 pts No prereq; used computechar.h

+ 1 pts no correct rules, but general flow is ok

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.

I would use git!

16b (5 points). Explain in at most half a page what kind of workflow

16.1 VC tool 1 / 1

✓ + 1 pts Name a valid VC Tool

+ 0 pts Incorrect

I would use git.

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?

My team will use a centralized remote repo which contains the "definitive" version of our project. Each engineer will clone a version of this repository and develop on their local version. When the engineer is ready, he can publish his additions to the remote repo by pushing to origin with "git push". In the event of a conflict (someone else pushed their changes, and your work is no longer compatible), engineers can use "git rebase" to resolve this. This method would not use branching, but branches could be added for different project versions.

16.2 VC workflow 3 / 5

- ✓ + 3 pts Discuss at least 3 points in argument
 - + 1 pts Clarity
 - + 1 pts Logical Workflow
 - + 1 pts Explanation missing some steps/unclear

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

The lack of feature branching means problems could arise if multiple engineers develop on the same feature at the same time. Since my team is small, this isn't a huge deal - I will just assign each engineer 3-4 distinct features.

16.3 VC solution concerns 1/1

✓ + 1 pts State at least 1 valid concern based on workflow above

+ 0 pts Unclear concern

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.  
git clone <url of remote repository>  
// coding here  
git add server.py  
git commit -m "Add new URL path to server.py"  
git push origin master  
(if this fails, first run  
  'git pull --rebase origin master'  
  then push again)
```

16.4 VC git commands 2 / 3

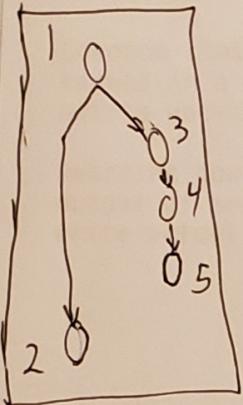
✓ + 1 pts Add and commit changes to server.py

+ 2 pts Commands match all workflow above

✓ + 1 pts Commands match partial workflow

+ 0 pts Incorrect

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.



Yes, this is possible, and happens when you merge.

In the graph to the left, commit 1 has parents 2 and 3, (1 is the result of a merge commit with commits 2 & 3).

In 'git log', the order these commits are listed in is:

1
2
3
4
5

As you can see, commit "3" is not the parent of commit "2", so this is possible.

17 Git - git log commit output 2 / 2

✓ - 0 pts Correct

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

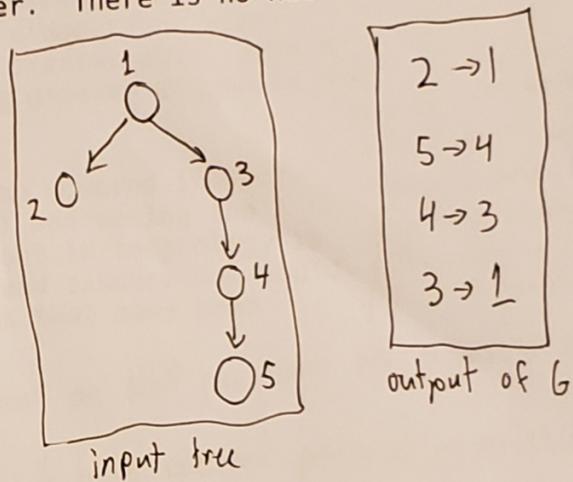
$h_{\text{parent}} \rightarrow h_{\text{child}}$

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.

Program G would output something like the right box for this tree.



This works for all valid topological orderings

We can process this data to create a slightly different dataset of the form $\{h_{\text{parents}}\} \rightarrow h_{\text{child}}$, where h_{parents} is a set of every parent of a certain node. Ex.

Next, iterate through the ordering output by P.

for every entry in P:

if the entry is a parent node, and it is not a child itself, it is ordered.

$\{2, 3\} \rightarrow 1$

$5 \rightarrow 4$

$4 \rightarrow 3$

processed data

valid:

2 5 4 3 |

5 4 3 2 |

Then, remove that directed edge from the dataset if the entry from the parent's set isn't in the data, or doesn't meet the above requirement, it's invalid (some conditions apply).

A valid ordering will process every entry in P and every edge in our processed dataset. That's how we can check !!

18 Git - topo order validation 0 / 4

✓ - 4 pts Incorrect

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.

a very cool program which
affects NLP models' predicti
success rates drastically
without changing meaning
↑ Has nothing to do with
e-learning.

My assignment 10 tech is irrelevant - it's about an NLP program textspoof which fools Google's BERT engine. However, I have relevant personal experience for this question. I work at Juni Learning, where I tutor children in CS online ~~using Zoom~~. My job is pretty much to do exactly what the professors here will have to do next quarter. I believe that Zoom can be an effective tool, but ~~it~~ is best used in small group settings. For discussions/labs, students can share their screens to get personal assistance, and the instructor can share his screen to teach new content. However, assisting students is only realistic in small groups (I'd say <20 people), and this method relies on all instructors/TAs having the requisite knowledge of how to use Zoom (so far, none of my ~~the~~ teachers except for those in this class have had this). Zoom is better than nothing, but might be better suited for a smaller university, ~~or~~ or simply smaller class sizes.

19 Writing - technology relevance 6 / 10

+ 10 pts Checked all boxes

✓ + 2 pts Mentioned 1 feature that can help tackle the given problems.

+ 4 pts Mentioned 2 or more features that can help tackle the given problems. Alternatively, covered 1 feature in great detail.

+ 1 pts Semi-clear writing

✓ + 2 pts Clarity of writing

✓ + 2 pts Discussed student's own presentation

+ 1 pts Semi-persuasive

+ 2 pts Persuasiveness

+ 0 pts No answer