

NAME: \_\_\_\_\_

COLLABORATOR(S): \_\_\_\_\_

1. What are three guiding principles of the Unix design philosophy?

5/3/0

2. Explain how the following pipeline meets those principles:

**cat sample-db.csv | grep PA | head -5**

5/3/0

3. What are the primary purpose for standard input, output, and error for different programs?

5/3/0

4. Consider the following command line with redirects

**grep PA < sample-db.csv 2> oops > sample-db.PA.csv**

5/3/1/0

a) What is the output file?

b) What is the input file?

c) What is the error file?

\_\_\_/20

5. Why is it necessary to have both standard error and standard out, with respect to the Unix design philosophy?

5/3/0

6. For the following command, where does the final output of the head command get written? That is, to what output file does the first 20 lines get written? Explain.

```
cat datafile | head -20 2> foo | cat > bar | cat > baz
```

8/5/1/0

7. Consider a situation where you want to include error output on a pipeline. You can do that using a redirect `>`, fill in the redirect below such that standard error is piped to `cat`'s standard input to produce the output below.

```
#>head -3 BAD_FILENAME  | cat BeatArmy.txt - GoNavy.txt
```

Go Navy!

head: BAD\_FILENAME: No such file or directory

Beat Army!

*(Hint: Note that `2>` redirects the file descriptor `2`, which is standard error, to a file, but `>` can also redirect to a file descriptor number by placing an `&` in front of the descriptor number.*

3/1/0

8. What does the following commands write to the output file `file`?

a) `cat < /dev/null > file`

3/1/0

b) `cat < file > /dev/null`

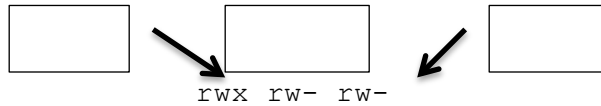
3/1/0

c) `head -c 20 /dev/urandom > file`

3/1/0

\_\_\_/25

9. Label the permission string below with the user, group, and all.



3/2/1/0

10. Convert the following permission states into an octet:

a) User: read and write; Group: read; Global: read

3/1/0

b) User: read and execute; Group execute; Global: none

3/1/0

11. Using the man pages (and **trying it yourself!**) describe the resulting action of the following chmod commands:

a) chmod a+x file

3/1/0

b) chmod u-x file

3/1/0

c) chmod og+r file

3/1/0

d) chmod -w file

3/1/0

e) chmod a-rw file

3/1/0

12. Consider a file with the following terminal output:

```
aviv@saddleback: tmp $ ls -l a.out
-rwxr-x--- 1 aviv scs 13358 Nov 19 08:09 a.out
aviv@saddleback: tmp $ groups
scs sudo mids ic221
```

Is user **aviv** allowed to change the group of **a.out** to **mids**? Explain why or why not?

3/1/0

Is user **aviv** allowed to change the owner to **m189999**? Explain why or why not?

3/1/0

13. Run the `~aviv/bin/ic221-up` command and then change into the `~/ic221/hw/02` directory. Change the permissions of the program **runme** such that it can be executed. What is the command you used and what is the output of execution?

command

output

4/2/1/0

14. List the directory `~aviv/ic221-hw/hw02/` where you will find binary called **secret**. Explain how you can still execute this program despite it not having group or global execute permissions.

5/3/1/0

15. What is the output of executing the `secret` program described above?

3/1/0

16. In what configuration file are `userid`'s and `usernames` defined?

2/0

17. In what configuration file are `groupid`'s and `groupnames` defined?

2/0

18. What are not all group members listed in the group configuration file?

2/0

19. In which of the configurations files is the users **default group** saved?

2/0

\_\_\_/20