Practice Exam Answer Key

True/False 1 point each (10 Questions)

1. True / False Symbolic Links create a link to a specific file anywhere else on the machine, ultimately acting as a shortcut, so that there is one single source of truth. (Q4Q2)
2. True / False `git` is a program for searching files by using regular expressions and glob patterns (Q5?)
3. True / False `.` in a directory is a reference to the directory itself (Q1!9)
4. True / False `fgrep` and `egrep` are aliases for `grep -F` and `grep -E` respectively. (Q2Q4)
5. True / False `fold` is a compression algorithm, used for reducing the size of a file. (Q2)
6. True / False Mathematical expressions are expanded by using wrapping the clauses in `{}` (Q1)
7. True / False `sftp` is a command used to log into a remote server and transfer files back and forth securely (Q4,q9)
8. True / False bash function parameters are positional parameters. (Q3)
9. True / False `case` statements must be closed with a `done` tag (Q3)
10. True / False `grep` is a command used to find and edit files by using regular expressions (Q2)

Multiple Choice 4 points each (10 questions)

`faculty_summer_2020.csv`

```
   firstName, lastName, email,  phone,  currentClass, className
   Shawna,  Climer, climersh@umsl.edu,  , 1250, Introduction to Computing
   Fadi,  Wedyan, wedyanf@umsl.edu,  ,  1250, Introduction to Computing
   Nazire,  Koc,  kocn@umsl.edu,  314-516-6356, 2250, Programming andDatastructures
   Steve, Riegerix, riegerixs@umsl.edu, , 2261, Object Oriented Programming
   Ankit,   Chaudhary,  chaudharya@umsl.edu,  314-516-4984, 2700, Computer Organization and Hardware
   Matthew, Lane, mjlny2@umsl.edu,  314-282-7563, 2750, System Programming and Tools
   Steve, Reigerix, riegerixs@umsl.edu, , 3010, Web Programming
   Gregory, Hommert,  gmhz7b@umsl.edu,  , 4220, Introduction to iOS Programming and Apps
   Galina,  Piatnitskaia, piatnitskaiag@umsl.edu, 314-516-5239, 4250, Programming Languages
   Mark,  Hauschild,  hauschildm@umsl.edu,  314-516-6426, 4732, Introduction to Cryptography for Computer Security
```

1. Which of the following `grep` commands will retrieve only users with phone numbers in the following data: (Module 2)
    1. `grep [0-9][0-9][0-9]-[0-9][0-9][0-9][0-9] faculty_summer_2020.csv`
    2. `grep [0-9]* faculty_summer_2020.csv`
    3. `grep (0-9)* faculty_summer_2020.csv`
    4. `grep {0-9}*3-{0-9}*4 faculty_summer_2020.csv`

2. The `i-list` is: (Module 4)
    1. A list of all users on the linux environment
    2. A multi-dimensional array matrix of all users and their associated groups
    3. A list of all input/output devices connected to the system
    4. A single dimensional array of i-nodes that the operating system searches through for commands and files.

3. Which of the following commands will **create** two separate directories, `dir_a` and `dir_b`, each with their own files `notes.md` and `assignments.md`: (Module 1)
    - `mkdir dir_{a,b} || touch dir_{a,b}/{notes.md,assignments.md}`
    - `mkdir dir_{a,b} && touch dir_{a,b}/{notes.md,assignments.md}`
    - `mkdir dir_{a,b} | touch dir_{a,b}/{notes.md,assignments.md}`
    - `touch dir_{a,b}/{notes.md,assignments.md}`

4. What is the output of the following Command: (Module 3) `greaterOrLessThan.sh`

```bash
#!/bin/bash

someNumber=5

if [[ $someNumber > 10 ]]
then
    echo $someNumber is greater than 10
else
    echo $someNumber is not greater than 10
fi
```

```
./greaterOrLessThan.sh
```

```
5 is greater than 10
```

```
5 is not greater than 10
```

```
someNumber is greater than 10
```

```
someNumber is not greater than 10
```

5. Which of the following commands will capture an interrupt (Module 3)

```
#!/bin/bash

trap  SIGINT

if [[ trap ]]
then
 echo Interrupt Caught!
fi

### Some more logic
```

```
#!/bin/bash

catch SIGINT "echo Interrupt Caught"

### Some more logic
```

```
#!/bin/bash

trap "echo Interrupt Caught!" SIGINT

### Some more logic
```

```
#!/bin/bash

catch SIGINT

if [[ catch ]]
then
 echo Interrupt Caught!
fi

### Some more logic
```

6. Which of the following is true about using the `-exec` flag with `find` (Module 4):
    1. `exec` executes a command with the files returned from find.
    2. `exec` deletes all files returned by the `find` command
    3. `exec` excludes all files matching the provided string.
    4. `exec` ensures that the `find` command does not traverse through other directories.
7. What affect will `chmod 070` have on a file with no permissions: (Module 1)
    - u: `r w x` g: `_ _ _` o: `r w x`
    - u: `_ _ x` g: `_ _ x` o: `_ _ x`
    - u: `_ _ _` g: `r w _` o: `_ _ _`
    - u: `_ _ _` g: `r w x` o: `_ _ _`
8. Which of the following will redirect the standard output of history to a file without overwriting is done with: (Module 1)
    - `history > historyFile.txt`
    - `history < historyFile.txt`
    - `history | historyFile.txt`
    - `history >> historyFile.txt`
9. If a user has no execute permissions for a particular directory, which of the following is true (Module 4):
    1. The user will not be able to access the files within the directory or any files in any subdirectories.
    2. The user will be able to access the files within the directory and subdirectory, but will not be able to view the files (i.e. the directory will appear empty)
    3. The user will only be able to access files if they specifically have a mode of `rwx`, otherwise, they will be inaccessible.
    4. File permissions only exist on ordinary files. It is not possible for directories to have file permissions.
10. What is the output of the following code (Module 3) `shiftyUntil.sh`

```bash
#!/bin/bash

inputVariables="$@"
continue=$1

until [[ $continue = "no" ]]
do
        echo You input: $continue
        shift
        continue=$1
done

echo You\'re Done!
```

```
./shiftyUntil.sh no ifs ands or buts
```

- You input: no
- You input: no You input: no You input: no .... (infinitely repeating)
- You're Done!
- You input: no You input: ifs You input: ands You input: or You input: buts

Short Answer 5 points each (8 questions)

1. Write a short bash script that takes in any number of command line arguments, and changes all input to lower case. (Module 3 && Module 1)

```
```

2. Write a short bash script that captures SIGINT and prints "Process cannot be interrupted" to the console. (Module 4)

```
```

```
```

3. Write a grep command that will search history for all previous `find` commands: ()

```
```

4. Write a find command that finds all files that don't have read, write, or execute permissions for the user, and then changes the mode to enable read access (Module 4)

```
```

5. Write a sed command that takes the output from `ls -l`, but displays only files that have less than 1000 bytes (Module 2)

6. Write a sed command that changes all coma delimited data to pipe delimited data (Module 2)

```

```

7. Write a brief shell script that will find all shell scripts, and then prompt the user to ask if they would like to delete the file (if yes, delete, if not, move on) (Module 3 && Module 4)

```

```

8. Write a brief shell script that will take in a specific file name, prompt the user whether they would like to gzip, bzip2, or xz compress the file. Depending on response, the script then ought to compress the provided file with the corresponding method (Module 4)

Long Answer 10 pts

1. Write a short bash script that takes in two arguments from a user, asks the user whether they would like to add, subtract, multiply, or divide. Each of these operations must be a function that returns data.