CSc 3320: Systems Programming

Spring 2021

Midterm 1: Total points = 100

Assigned: 26th Feb 2021: 12.01 PM

Submission Deadline: 2nd Mar 2021: 12.01 PM

(No extensions. If your submission is not received by this time then it will NOT be accepted.)

Submission instructions:

1. Create a Google doc for your submission.
2. Start your responses from page 2 of the document and copy these instructions on page 1.
3. Fill in your name, campus ID and panther # in the fields provided. If this information is missing TWO POINTS WILL BE DEDUCTED.
4. Keep this page 1 intact. If this *submissions instructions* page is missing in your submission TWO POINTS WILL BE DEDUCTED.
5. Start your responses to each QUESTION on a new page.
6. If you are being asked to write code copy the code into a separate txt file and submit that as well. The code should be executable. E.g. if asked for a C script then provide myfile.c so that we can execute that script. In your answer to the specific question, provide the steps on how to execute your file (like a ReadMe).
7. If you are being asked to test code or run specific commands or scripts, provide the evidence of your outputs through a screenshot and/or screen video-recordings and copy the same into the document.
8. Upon completion, download a .PDF version of the google doc document and submit the same along with all the supplementary files (videos, pictures, scripts etc).

Full Name: Hawa Sylla

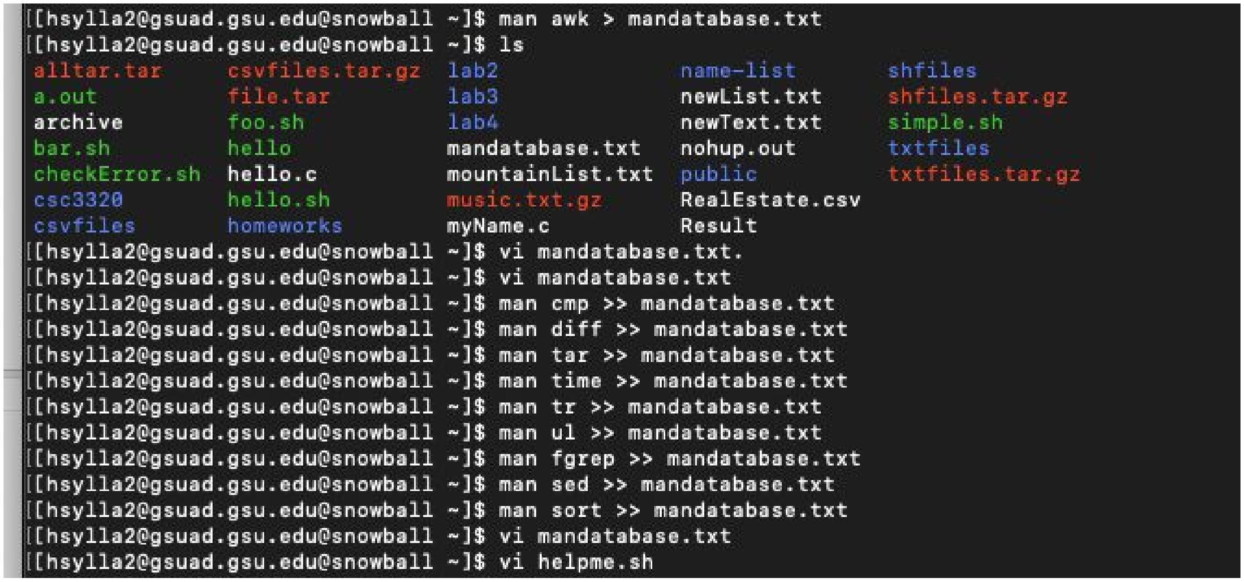
Campus ID: hsylla2

Panther #: 002-31-2868

Questions 1-5 are 20pts each

1. Pick any of your 10 favourite unix commands. For each command run the *man* command and copy the text that is printed into a mandatabase.txt. Write a shell script *helpme.sh* that will ask the user to type in a command and then print the manual's text associated with that corresponding command. If the command the user types is not in the database then the script must print

   sorry, I cannot help you



```
[[hsylla2@gsuad.gsu.edu@snowball ~]$ man awk > mandatabase.txt
[[hsylla2@gsuad.gsu.edu@snowball ~]$ ls
 alltar.tar     csvfiles.tar.gz  lab2              name-list       shfiles
 a.out          file.tar         lab3              newList.txt     shfiles.tar.gz
 archive        foo.sh           lab4              newText.txt     simple.sh
 bar.sh         hello            mandatabase.txt   nohup.out       txtfiles
 checkError.sh  hello.c          mountainList.txt  public          txtfiles.tar.gz
 csc3320        hello.sh         music.txt.gz      RealEstate.csv
 csvfiles       homeworks        myName.c          Result
[[hsylla2@gsuad.gsu.edu@snowball ~]$ vi mandatabase.txt.
[[hsylla2@gsuad.gsu.edu@snowball ~]$ vi mandatabase.txt
[[hsylla2@gsuad.gsu.edu@snowball ~]$ man cmp >> mandatabase.txt
[[hsylla2@gsuad.gsu.edu@snowball ~]$ man diff >> mandatabase.txt
[[hsylla2@gsuad.gsu.edu@snowball ~]$ man tar >> mandatabase.txt
[[hsylla2@gsuad.gsu.edu@snowball ~]$ man time >> mandatabase.txt
[[hsylla2@gsuad.gsu.edu@snowball ~]$ man tr >> mandatabase.txt
[[hsylla2@gsuad.gsu.edu@snowball ~]$ man ul >> mandatabase.txt
[[hsylla2@gsuad.gsu.edu@snowball ~]$ man fgrep >> mandatabase.txt
[[hsylla2@gsuad.gsu.edu@snowball ~]$ man sed >> mandatabase.txt
[[hsylla2@gsuad.gsu.edu@snowball ~]$ man sort >> mandatabase.txt
[[hsylla2@gsuad.gsu.edu@snowball ~]$ vi mandatabase.txt
[[hsylla2@gsuad.gsu.edu@snowball ~]$ vi helpme.sh
```

Read me

./helpme.sh

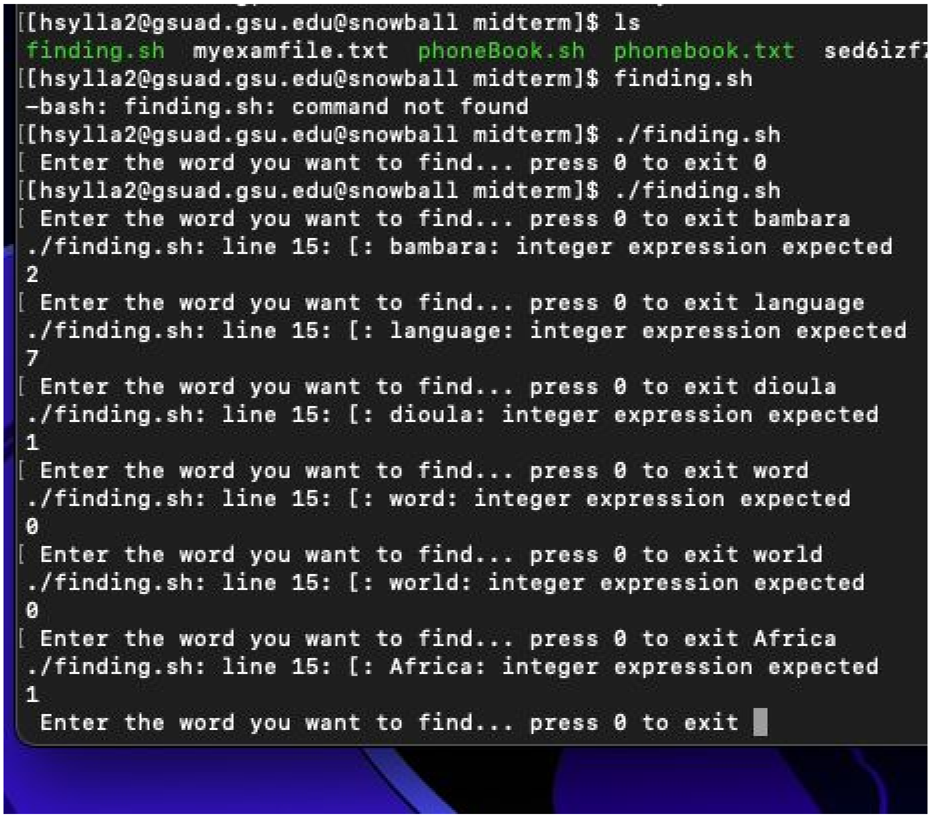
The system will prompt : "type in command you want to find. Press 1 to begin and see available commands, or press 2 to exit."

Once the user presses 1, they are given a list of commands to choose from. they can then type out the command by inputting one of the ones on the list, or exit , if the user types in a valid input, they will get the manual entry for the command as an output. otherwise they will receive error.

Screen Recording of Demo for helpme.sh Provided with other files.

2. On your computer open your favourite Wikipedia page. Copy the text from that page into a text file **myexamfile.txt** and then copy that file to a directory named **midterm** (use mkdir to create the directory if it doesn’t exist) in your snowball server home directory (use any FTP tool such as Putty or Filezilla to copy the file from your computer to the remote snowball server machine: see Lab 6).

Write a shell script that will find the number of occurrences of a particular keyword typed by the user. Present evidence of your testing with at least 5 trials (different keywords each time)



```
[[hsylla2@gsuad.gsu.edu@snowball midterm]$ ls
 finding.sh  myexamfile.txt  phoneBook.sh  phonebook.txt  sed6izf
[[hsylla2@gsuad.gsu.edu@snowball midterm]$ finding.sh
 -bash: finding.sh: command not found
[[hsylla2@gsuad.gsu.edu@snowball midterm]$ ./finding.sh
[ Enter the word you want to find... press 0 to exit 0
[[hsylla2@gsuad.gsu.edu@snowball midterm]$ ./finding.sh
[ Enter the word you want to find... press 0 to exit bambara
 ./finding.sh: line 15: [: bambara: integer expression expected
 2
[ Enter the word you want to find... press 0 to exit language
 ./finding.sh: line 15: [: language: integer expression expected
 7
[ Enter the word you want to find... press 0 to exit dioula
 ./finding.sh: line 15: [: dioula: integer expression expected
 1
[ Enter the word you want to find... press 0 to exit word
 ./finding.sh: line 15: [: word: integer expression expected
 0
[ Enter the word you want to find... press 0 to exit world
 ./finding.sh: line 15: [: world: integer expression expected
 0
[ Enter the word you want to find... press 0 to exit Africa
 ./finding.sh: line 15: [: Africa: integer expression expected
 1
  Enter the word you want to find... press 0 to exit
```

ReadMe ./finding.sh

to find the count of word, input the word you want to find in myexamfile.txt.

if you are done with the util, press 0 to exit.

3. Write a shell script to find files in a directory hierarchy (e.g. your home directory) that have not been accessed for N days and compress them. Here N is a parameter and the user will be asked for that input as the first step of the script execution.

   Read Me ./archive.sh

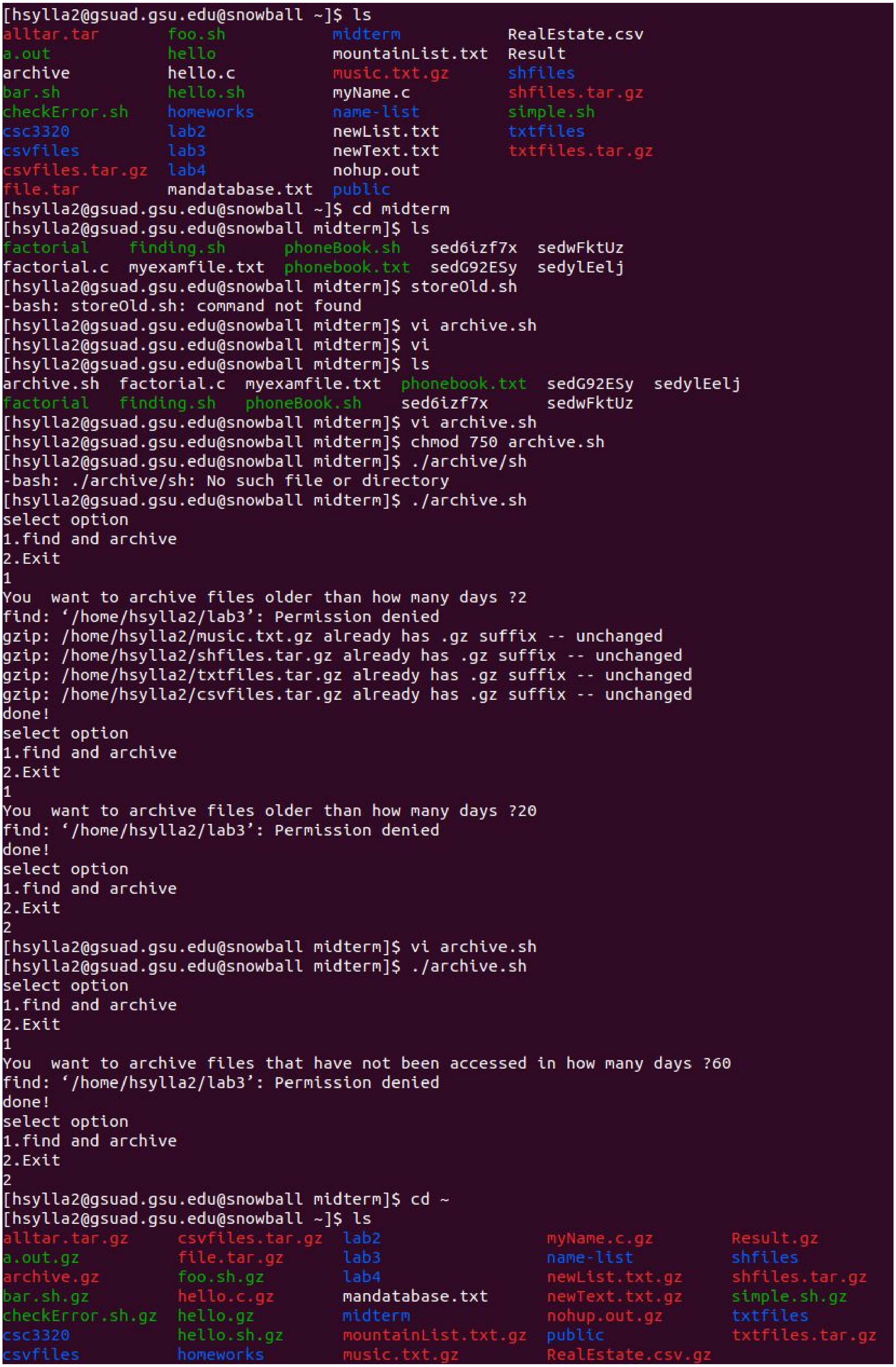
The system will prompt you to find and archive, or exit the utility.

If you choose to find and archive, The system will then ask you, "You want to archive files that have not been accessed in many days ?"

After you input the number, (eg. 3, for files that have not been accessed in three days," It will compress all the files that fit that criteria.

Once it is finished, it will prompt 'Done!"

Example Screenshot on following page:



```
[hsylla2@gsuad.gsu.edu@snowball ~]$ ls
alltar.tar       foo.sh           midterm           RealEstate.csv
a.out            hello            mountainList.txt  Result
archive          hello.c          music.txt.gz      shfiles
bar.sh           hello.sh         myName.c          shfiles.tar.gz
checkError.sh    homeworks        name-list         simple.sh
csc3320          lab2             newList.txt       txtfiles
csvfiles         lab3             newText.txt       txtfiles.tar.gz
csvfiles.tar.gz  lab4             nohup.out
file.tar         mandatabase.txt  public
[hsylla2@gsuad.gsu.edu@snowball ~]$ cd midterm
[hsylla2@gsuad.gsu.edu@snowball midterm]$ ls
factorial    finding.sh      phoneBook.sh   sed6izf7x  sedwFktUz
factorial.c  myexamfile.txt  phonebook.txt  sedG92ESy  sedylEelj
[hsylla2@gsuad.gsu.edu@snowball midterm]$ storeOld.sh
-bash: storeOld.sh: command not found
[hsylla2@gsuad.gsu.edu@snowball midterm]$ vi archive.sh
[hsylla2@gsuad.gsu.edu@snowball midterm]$ vi
[hsylla2@gsuad.gsu.edu@snowball midterm]$ ls
archive.sh  factorial.c  myexamfile.txt  phonebook.txt  sedG92ESy  sedylEelj
factorial   finding.sh   phoneBook.sh    sed6izf7x      sedwFktUz
[hsylla2@gsuad.gsu.edu@snowball midterm]$ vi archive.sh
[hsylla2@gsuad.gsu.edu@snowball midterm]$ chmod 750 archive.sh
[hsylla2@gsuad.gsu.edu@snowball midterm]$ ./archive/sh
-bash: ./archive/sh: No such file or directory
[hsylla2@gsuad.gsu.edu@snowball midterm]$ ./archive.sh
select option
1.find and archive
2.Exit
1
You  want to archive files older than how many days ?2
find: ‘/home/hsylla2/lab3’: Permission denied
gzip: /home/hsylla2/music.txt.gz already has .gz suffix -- unchanged
gzip: /home/hsylla2/shfiles.tar.gz already has .gz suffix -- unchanged
gzip: /home/hsylla2/txtfiles.tar.gz already has .gz suffix -- unchanged
gzip: /home/hsylla2/csvfiles.tar.gz already has .gz suffix -- unchanged
done!
select option
1.find and archive
2.Exit
1
You  want to archive files older than how many days ?20
find: ‘/home/hsylla2/lab3’: Permission denied
done!
select option
1.find and archive
2.Exit
2
[hsylla2@gsuad.gsu.edu@snowball midterm]$ vi archive.sh
[hsylla2@gsuad.gsu.edu@snowball midterm]$ ./archive.sh
select option
1.find and archive
2.Exit
1
You  want to archive files that have not been accessed in how many days ?60
find: ‘/home/hsylla2/lab3’: Permission denied
done!
select option
1.find and archive
2.Exit
2
[hsylla2@gsuad.gsu.edu@snowball midterm]$ cd ~
[hsylla2@gsuad.gsu.edu@snowball ~]$ ls
alltar.tar.gz     csvfiles.tar.gz  lab2                 myName.c.gz        Result.gz
a.out.gz          file.tar.gz      lab3                 name-list          shfiles
archive.gz        foo.sh.gz        lab4                 newList.txt.gz     shfiles.tar.gz
bar.sh.gz         hello.c.gz       mandatabase.txt      newText.txt.gz     simple.sh.gz
checkError.sh.gz  hello.gz         midterm              nohup.out.gz       txtfiles
csc3320           hello.sh.gz      mountainList.txt.gz  public             txtfiles.tar.gz
csvfiles          homeworks        music.txt.gz         RealEstate.csv.gz
```

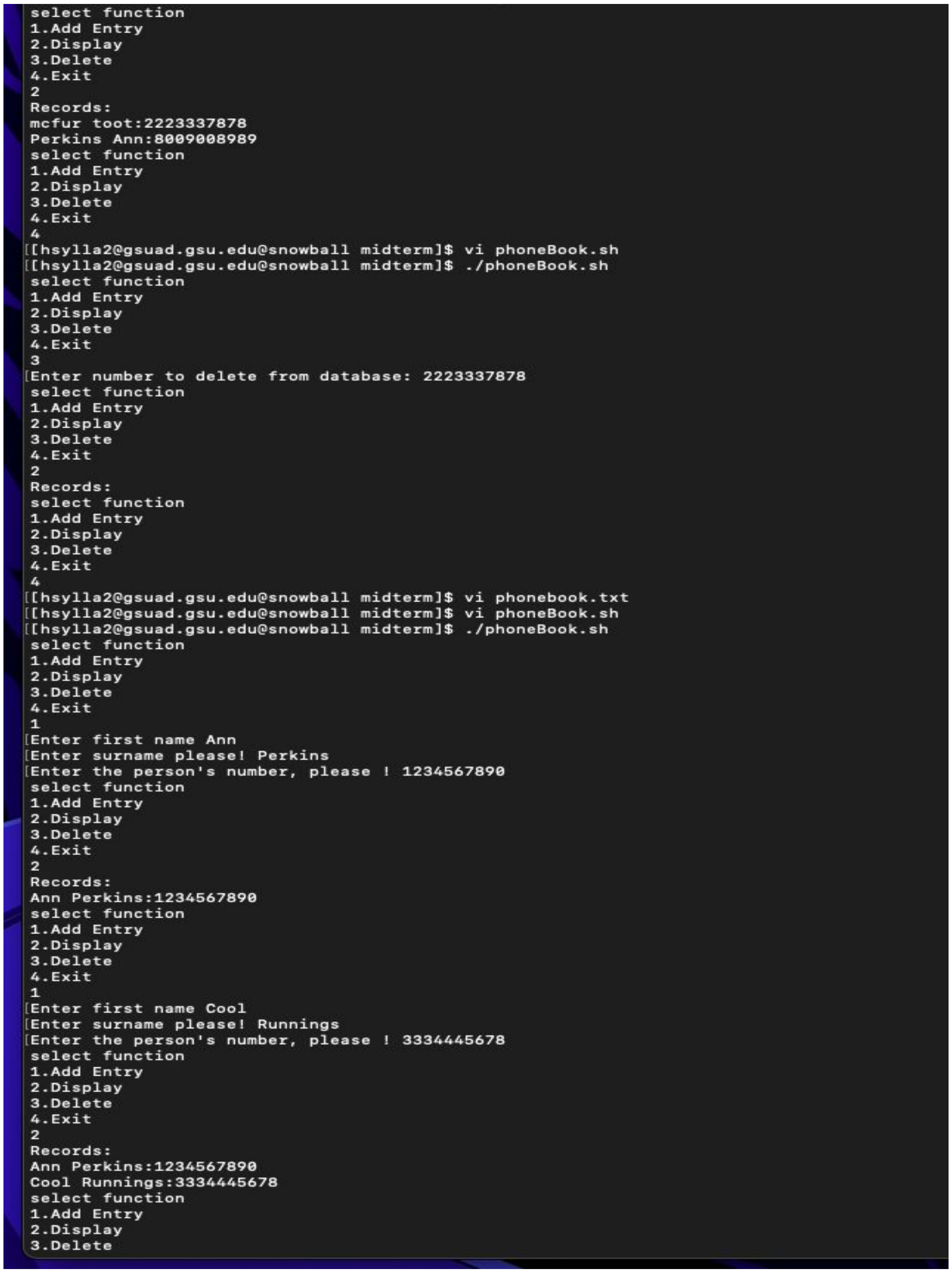
4. Build a phone-book utility that allows you to access and modify an alphabetical list of names, addresses and telephone numbers. Use utilities such as awk and sed, to maintain and edit the file of phone-book information. The user (in this case, you) must be able

to read, edit, and delete the phone book contents. The permissions for the phone book database must be such that it is inaccessible to anybody other than the user.

To Use: ./phoneBook.sh

select function from 4 options, type the corresponding option

to add entry, enter the first name, last name, and the phone number. To delete entry, enter the corresponding number. When the user is finished, press 4 to exit the utility.



```
 select function
 1.Add Entry
 2.Display
 3.Delete
 4.Exit
 2
 Records:
 mcfur toot:2223337878
 Perkins Ann:8009008989
 select function
 1.Add Entry
 2.Display
 3.Delete
 4.Exit
 4
[[hsylla2@gsuad.gsu.edu@snowball midterm]$ vi phoneBook.sh
[[hsylla2@gsuad.gsu.edu@snowball midterm]$ ./phoneBook.sh
 select function
 1.Add Entry
 2.Display
 3.Delete
 4.Exit
 3
[Enter number to delete from database: 2223337878
 select function
 1.Add Entry
 2.Display
 3.Delete
 4.Exit
 2
 Records:
 select function
 1.Add Entry
 2.Display
 3.Delete
 4.Exit
 4
[[hsylla2@gsuad.gsu.edu@snowball midterm]$ vi phonebook.txt
[[hsylla2@gsuad.gsu.edu@snowball midterm]$ vi phoneBook.sh
[[hsylla2@gsuad.gsu.edu@snowball midterm]$ ./phoneBook.sh
 select function
 1.Add Entry
 2.Display
 3.Delete
 4.Exit
 1
[Enter first name Ann
[Enter surname please! Perkins
[Enter the person's number, please ! 1234567890
 select function
 1.Add Entry
 2.Display
 3.Delete
 4.Exit
 2
 Records:
 Ann Perkins:1234567890
 select function
 1.Add Entry
 2.Display
 3.Delete
 4.Exit
 1
[Enter first name Cool
[Enter surname please! Runnings
[Enter the person's number, please ! 3334445678
 select function
 1.Add Entry
 2.Display
 3.Delete
 4.Exit
 2
 Records:
 Ann Perkins:1234567890
 Cool Runnings:3334445678
 select function
 1.Add Entry
 2.Display
 3.Delete
```

:

5.

A. Write a C script that will compute the factorial of a given number (positive integer).

B. Write a C script to find the new integer value of an original integer when it is bit-shifted left by 3 bits and added to its complement (one's complement of the original integer).

(Note: You can manually type in the binary representation of the original integer)

(10 bonus points for writing the C script to convert the integer to binary and vice-versa)

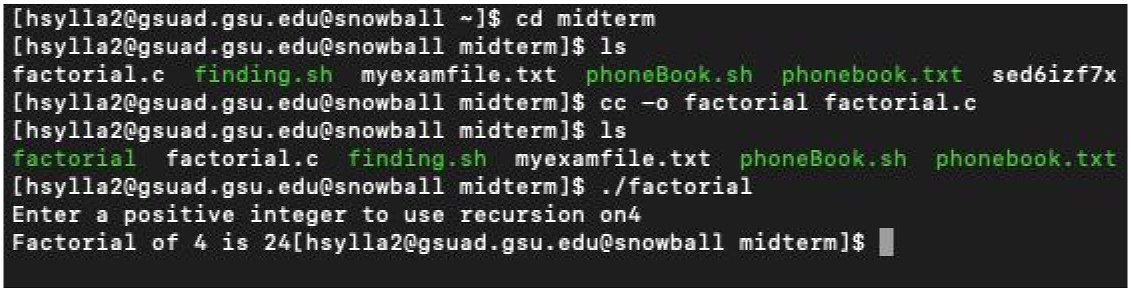
(10 bonus points for writing a shell script that will execute both the C scripts from above for a given integer number)

A. **Read Me**

compile factorial.c ($cc -o factorial factorial.c)

execute the compiled program: factorial. (./factorial)

The system will prompt you for a positive integer to use recursion on. Enter a positive integer. The system will output the answer.



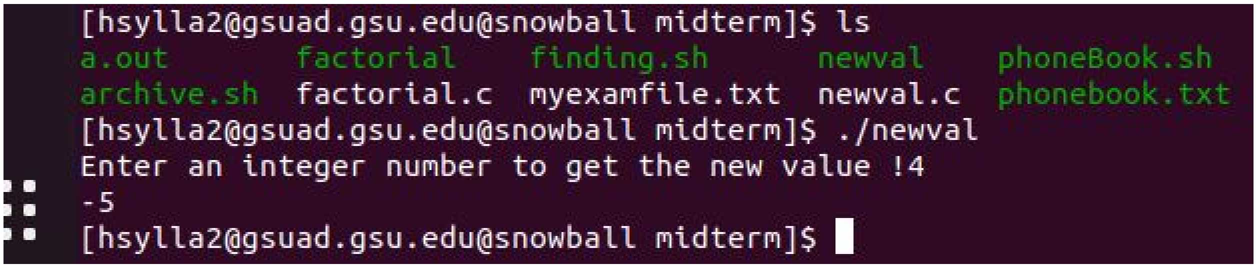
```
[hsylla2@gsuad.gsu.edu@snowball ~]$ cd midterm
[hsylla2@gsuad.gsu.edu@snowball midterm]$ ls
factorial.c  finding.sh  myexamfile.txt  phoneBook.sh  phonebook.txt  sed6izf7x
[hsylla2@gsuad.gsu.edu@snowball midterm]$ cc -o factorial factorial.c
[hsylla2@gsuad.gsu.edu@snowball midterm]$ ls
factorial  factorial.c  finding.sh  myexamfile.txt  phoneBook.sh  phonebook.txt
[hsylla2@gsuad.gsu.edu@snowball midterm]$ ./factorial
Enter a positive integer to use recursion on4
Factorial of 4 is 24[hsylla2@gsuad.gsu.edu@snowball midterm]$ 
```

B. **Read Me**

compile newval.c ($cc -o newval newval.c)

execute the compiled program: newval. (./newval)

The system will prompt you for a positive integer to enter. Enter a positive integer. The system will output the answer.



```
[hsylla2@gsuad.gsu.edu@snowball midterm]$ ls
a.out        factorial    finding.sh      newval    phoneBook.sh
archive.sh   factorial.c  myexamfile.txt  newval.c  phonebook.txt
[hsylla2@gsuad.gsu.edu@snowball midterm]$ ./newval
Enter an integer number to get the new value !4
-5
[hsylla2@gsuad.gsu.edu@snowball midterm]$ 
```