**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: Tracy Michaels

Campus ID: tmichaels1

Panther #: 002430918

**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*

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)

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.

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.

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

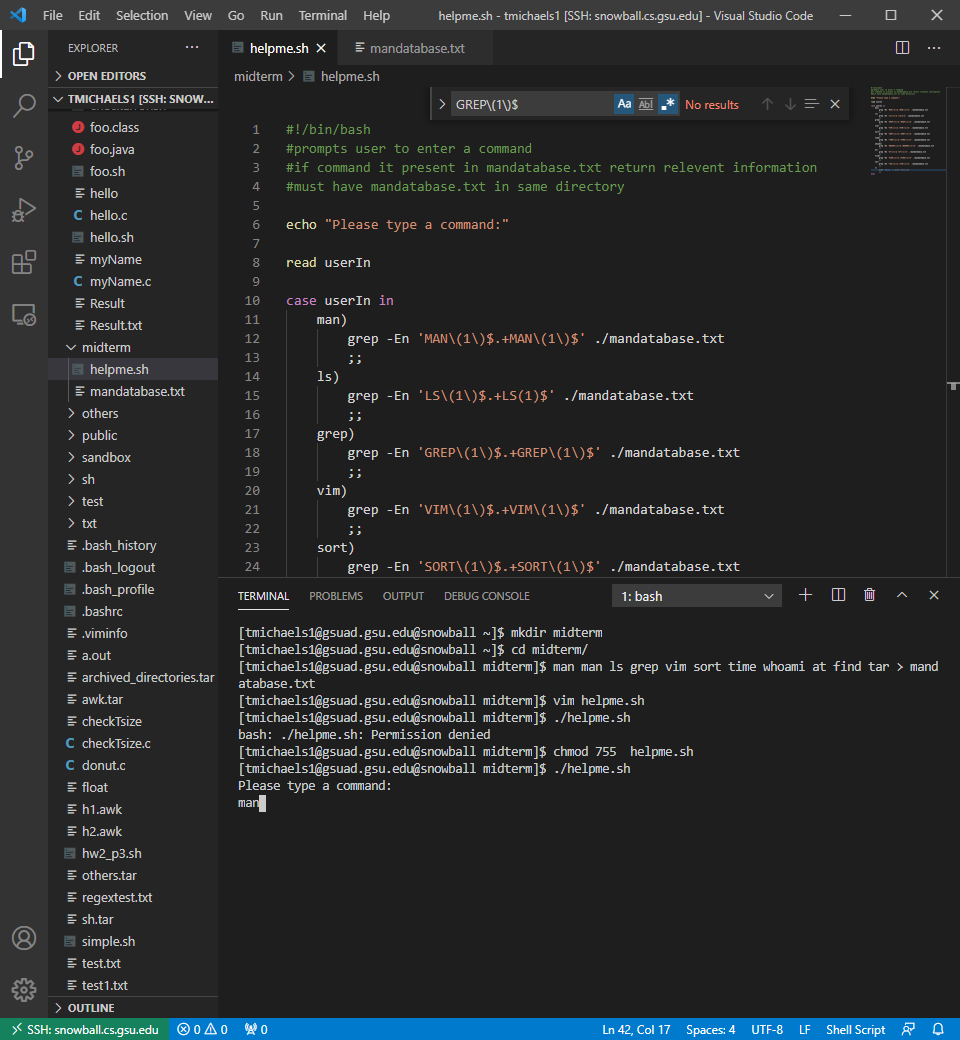
binary and vice-versa)

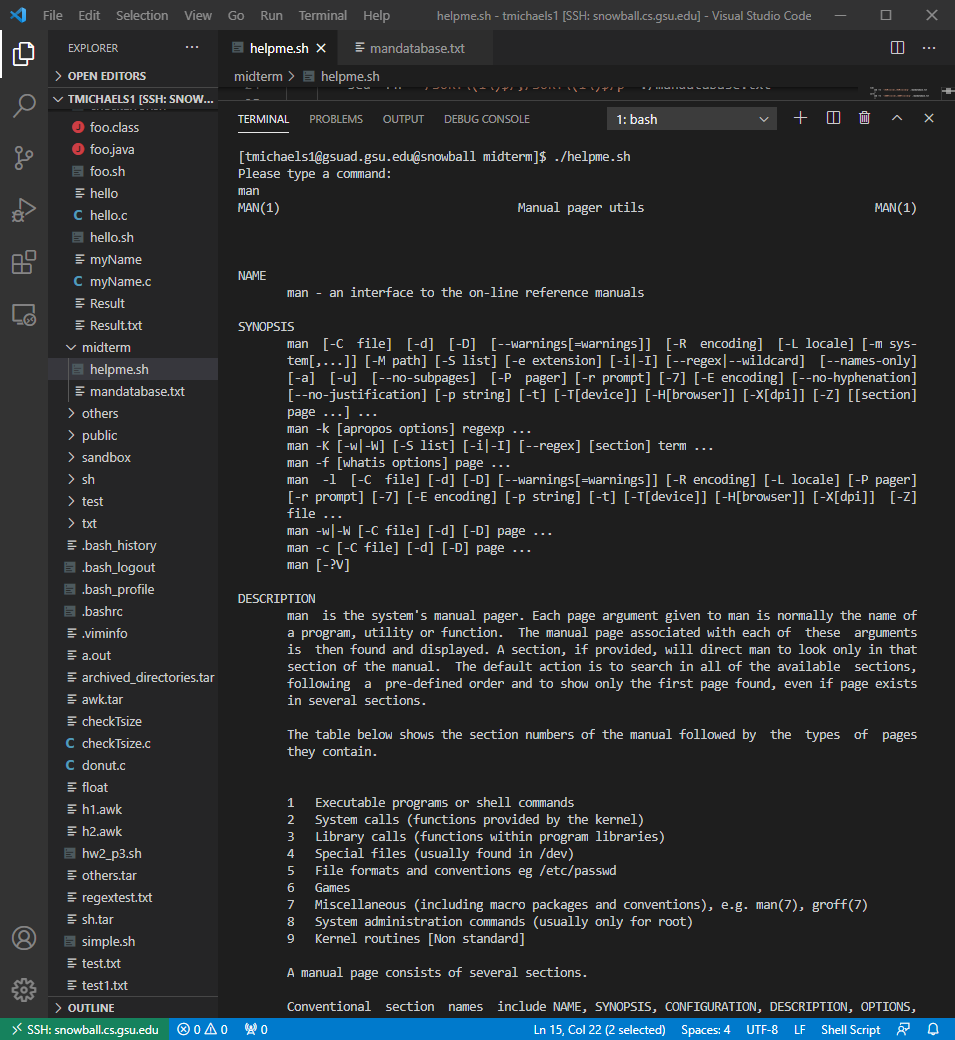
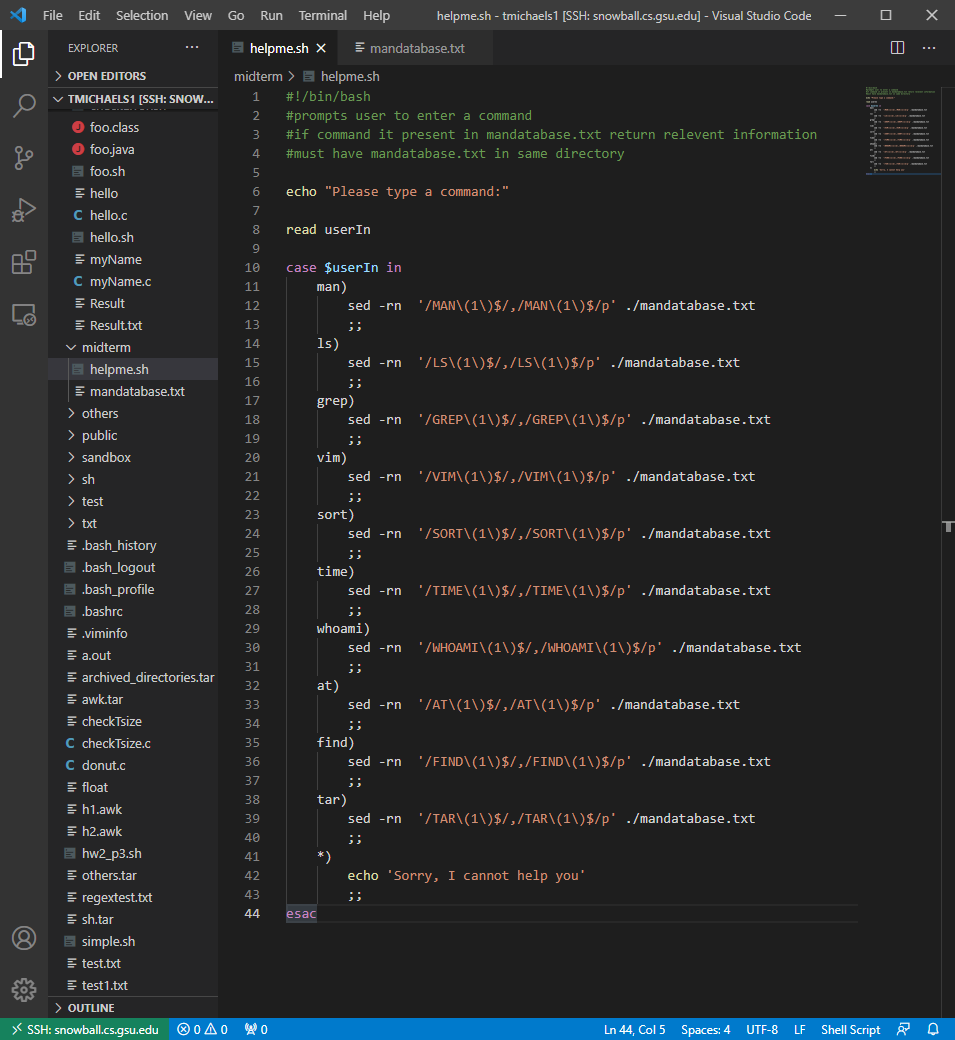
(10 bonus points for writing a shell script that will execute both the

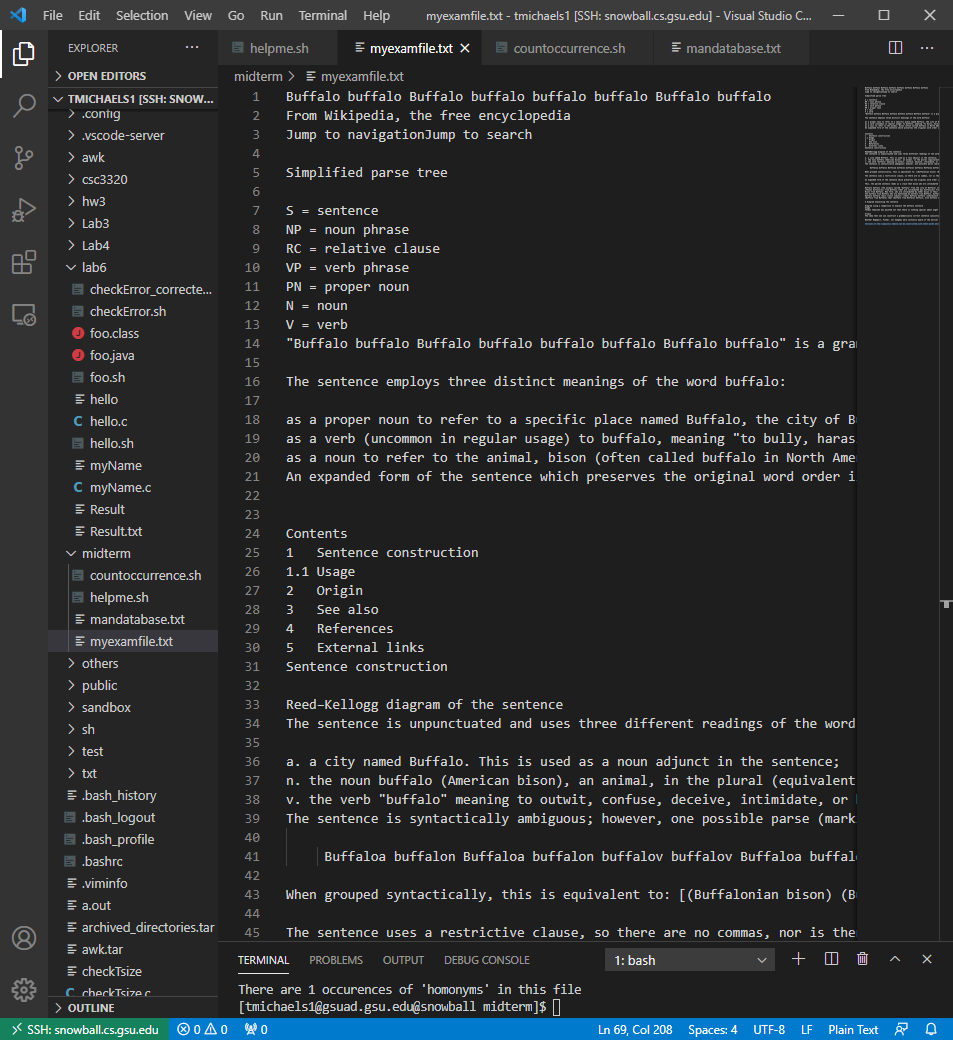
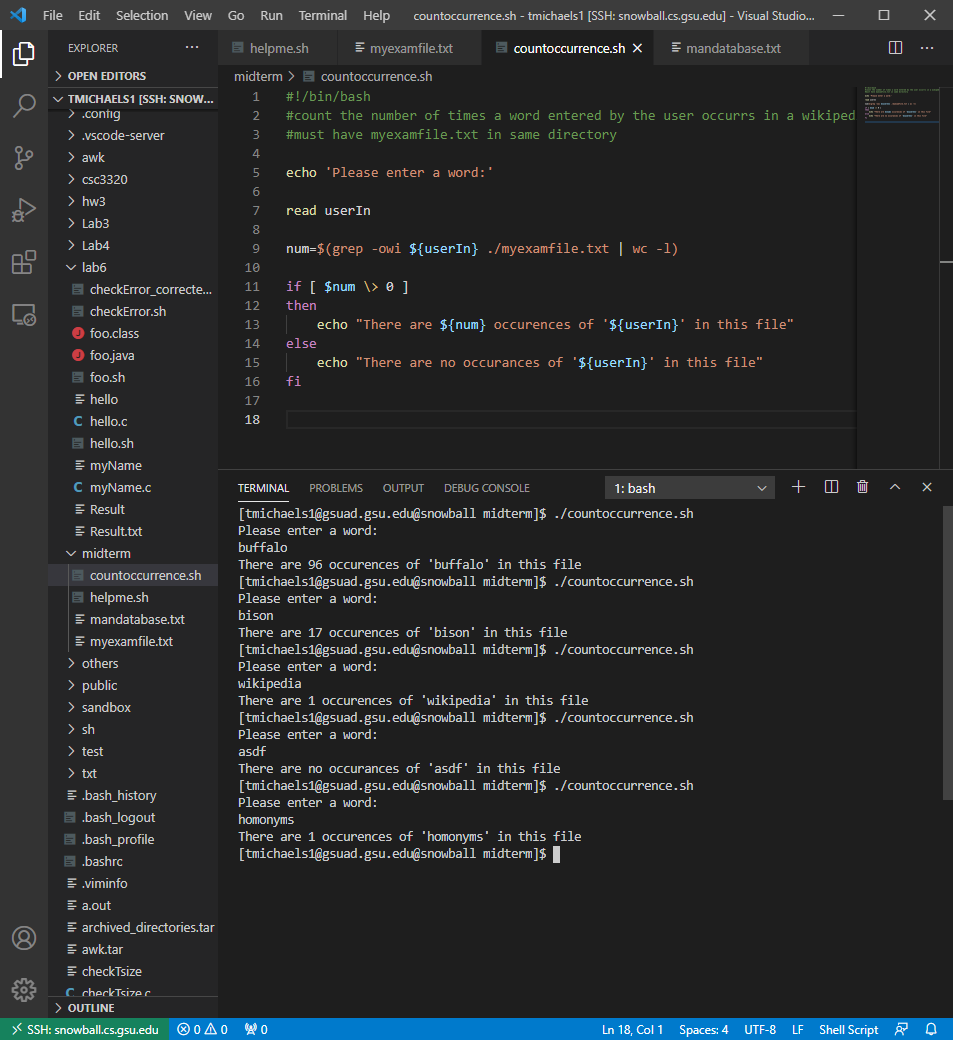
C scripts from above for a given integer number)

1. Used command man man ls grep vim sort time whoami at find tar > mandatabase.txt to get the information from each manuals and store them in a text

Created helpme.sh that prompts for a command and it that command is in the mandatabase.txt it will print the man pages for that command

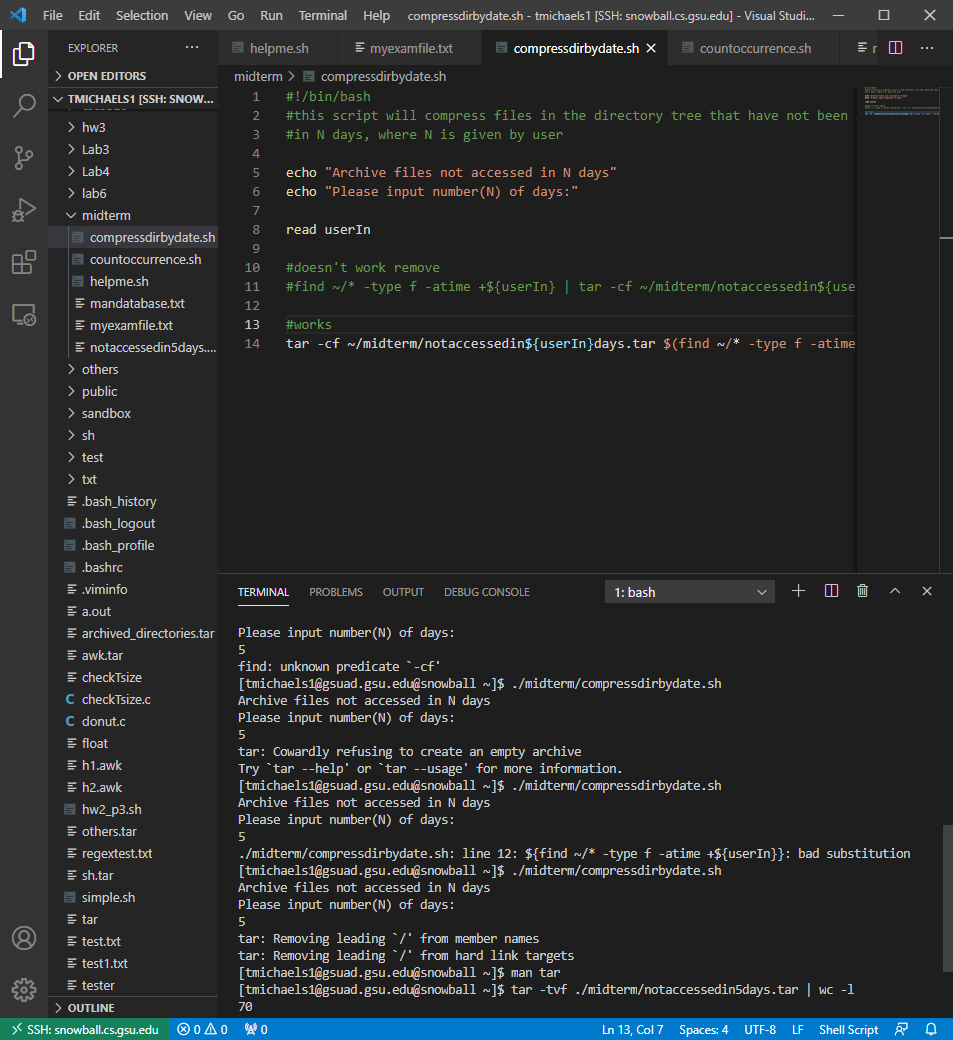


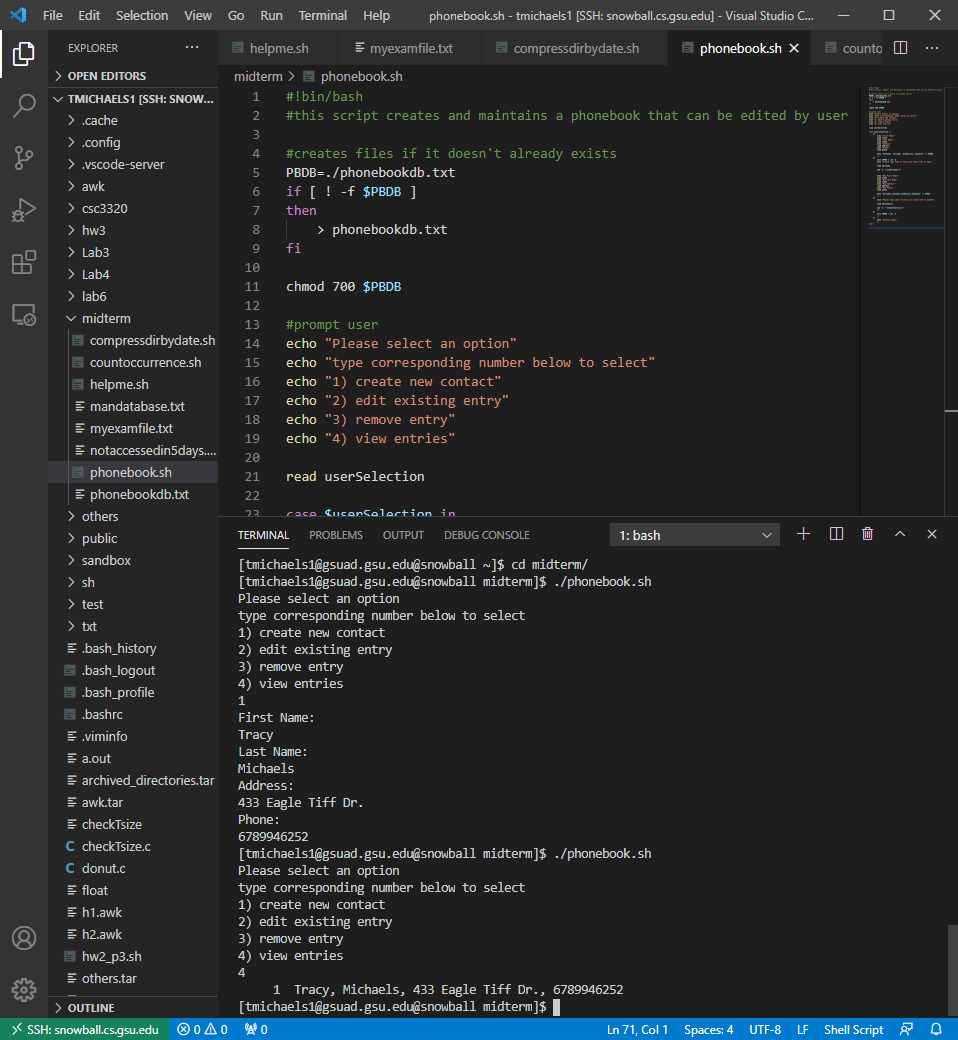


2)asks user for a word and searches myexamplyfile for that word and returns the number of times it occurs in that file

3) compresses all files that have not been accesed in the last N days which is supplied by the user

In the output in the screenshot I counted the number of files that were saved in the .tar file and compared that to the number of files that have not been accessed in the last 5 days and they match so I’m positive that it gotthem all.



4) 

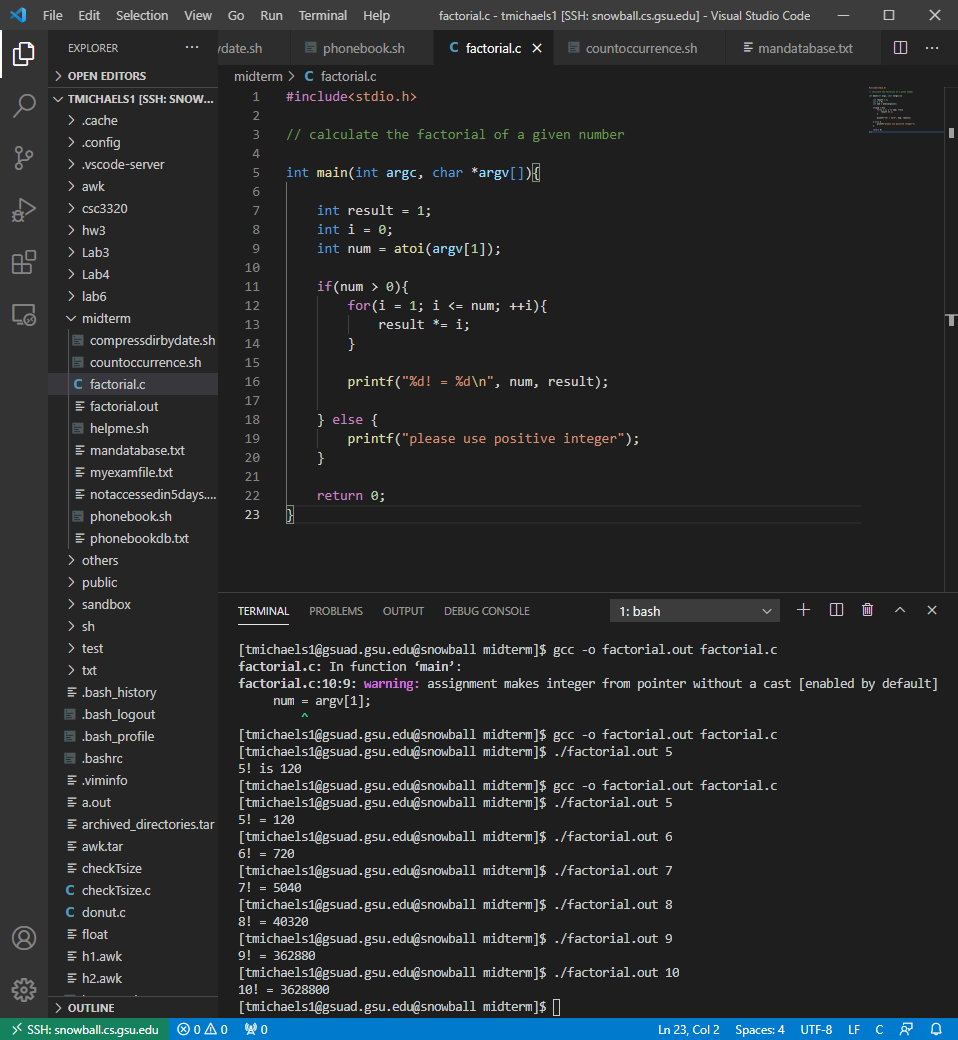
5)

A) to compile this code I used the command gcc -o factorial.out factorial.c

This creates a file called factorial.out

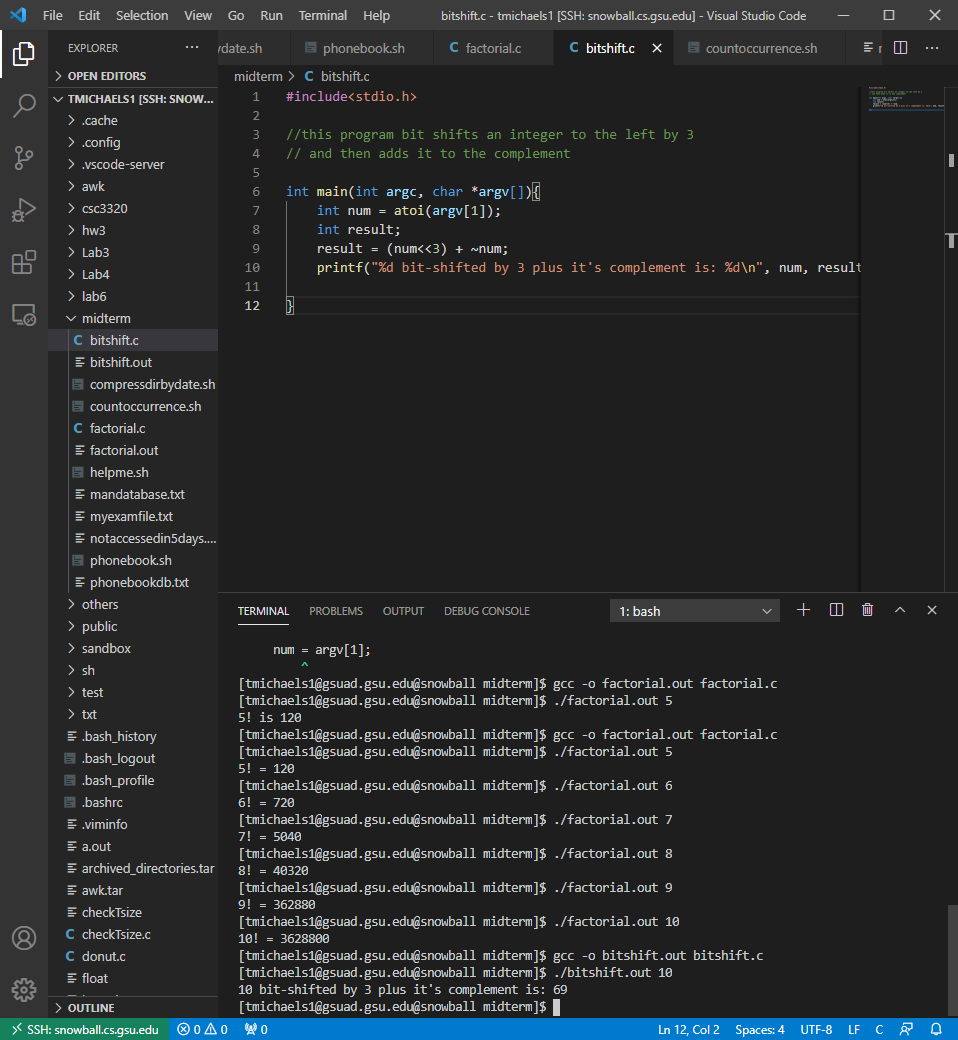
This program accepts a command line argument for the input so to run this program I typed ./factorial.out 5

Which gave the desired output of 5! = 120



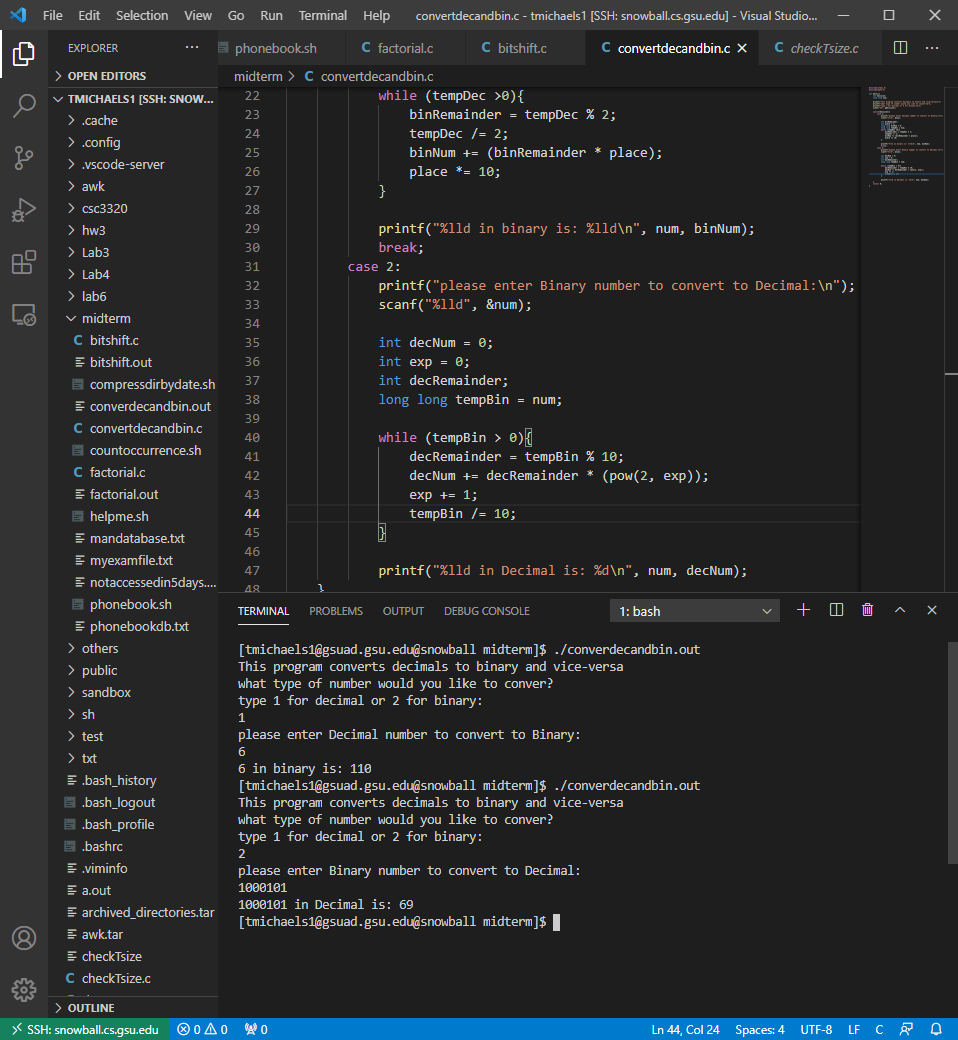
B) compiled this program using gcc -o bitshift.out bitshift.c

Bitshif.out accepts a command line argument of an integer to perform it’s operation on

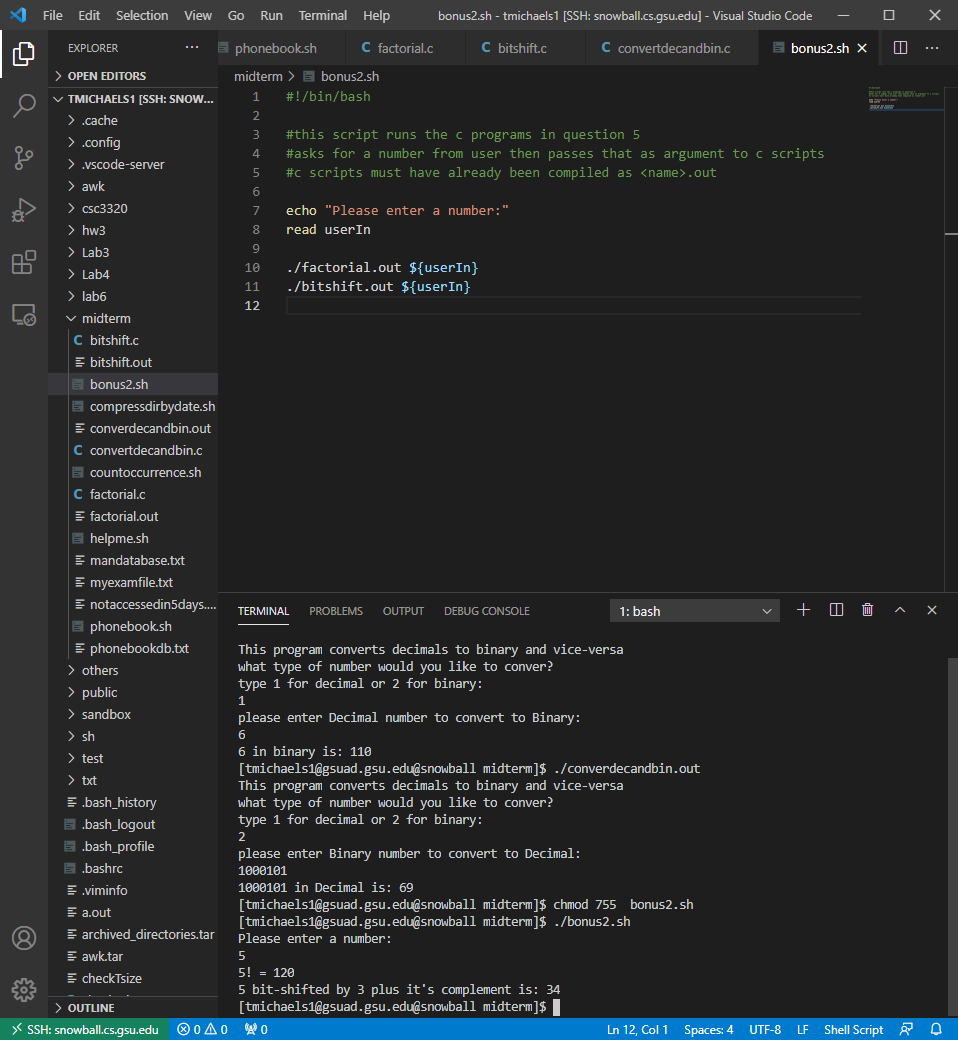


Bonus 1) compiled with gcc -o converdecandbin.out convertdecandbin.c -lm

Had to use -lm to link math



Bonus 2)



I compressed all files in the midterm directory and will upload that as well