CSc 3320: Systems Programming

Fall 2021

Midterm 1: Total points = 100

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 program 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).
9. Scripts/Code without proper comments, indentation and titles (must have the name of the program, and name & email of the programmer on top the script).

Full Name: Aparna Mandapaka

Campus ID: amandapaka2

Panther #: 002553236

Questions 1-5 are 20pts each

1. (20 pts) 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. (10pts each) 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).

a. Write a shell script that will find the number of statements in the text. A statement is defined as the collection of text between two periods (full-stops).
b. Update the script to present a tabular list that shows the number of words and number of letters in each statement.

3. (20pts) Design a calculator using a shell script using regular expressions. The calculator, at the minimum, must be able to process addition, subtraction, multiplication, division and modulo operations. It must also have cancel and clear features.

4. (20pts) 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 you (the user).

5. (4 pts each) Give brief answers with examples, wherever relevant

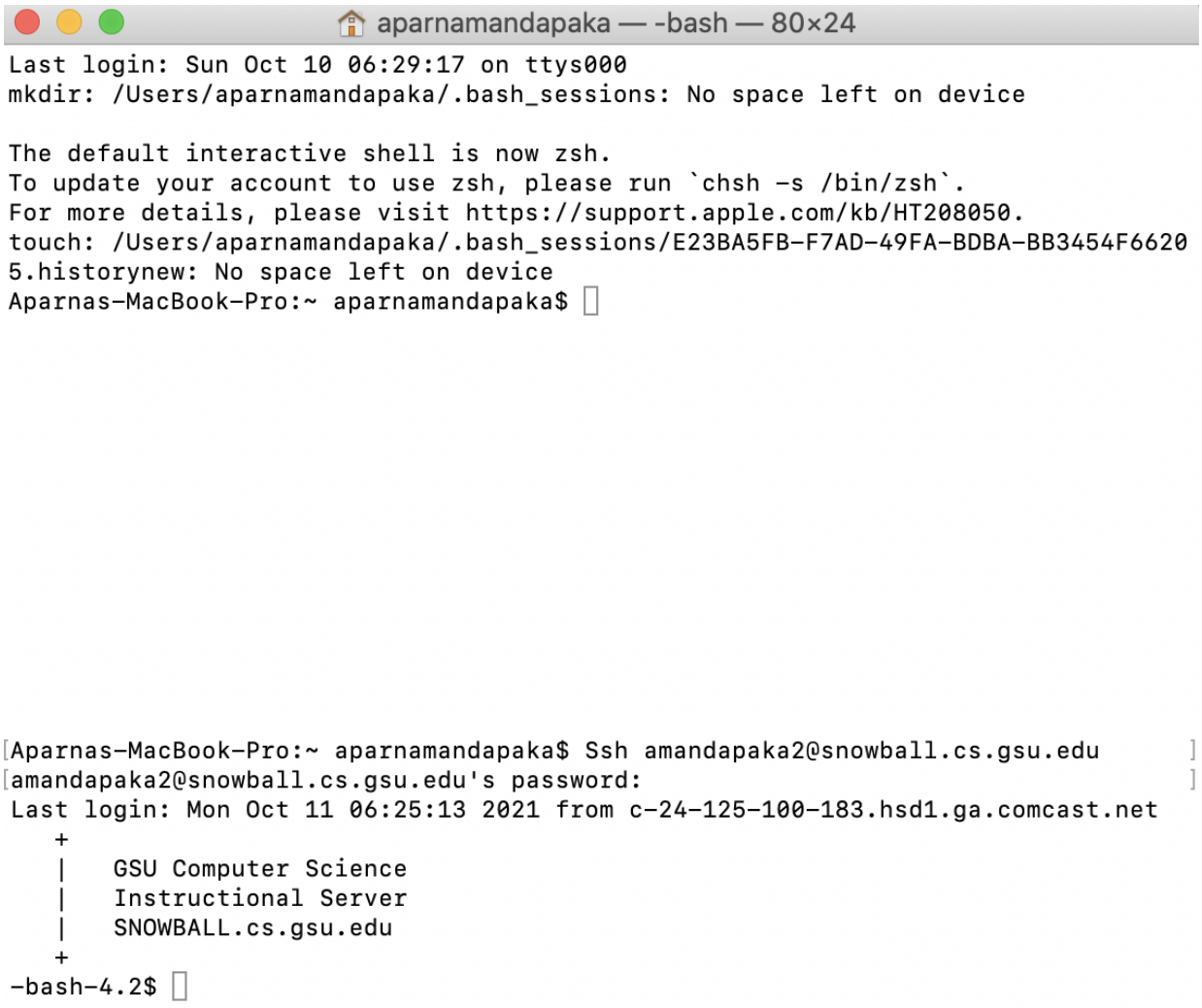
A. What is the use of a shell?

- Shell is a program that is used as an interface between the user and the UNIX operating system. One of its uses is that it is used to control the computer using the commands instead of the Graphical User interface. Shell basically takes the command from programs and it also helps to interact with the system
- Some examples of shells are Powershell, sh, csh and tcsh.

B. Is there any difference between the shell that you see on your PC versus that you see on the snowball server upon login. If yes, what are they? Provide screenshots for examples.

Yes, there is a difference the shell that we use on PC vs the one on the snowball serve. The Snowball server stops all the commands running in the system between the data source whereas the shell in PC's doesn't do that. To secure the snowball secure shell

is used for the connections.



C. What are the elements in a computer (software and hardware) that enable the understanding and interpretation of a C program?

As C is a compiled language and not an interpreted language. So that it is intercepted by a compiler of the C language and machine code generates and executes the output. The interpreted language is completely different from the compiled language.

D. The “printf()” C command is used for printing anything on the screen. In bash we use the command “echo ”. What is the difference (if any) in terms of how the computer interprets and executes these commands?

The printf() and echo are the in-built commands. Echo command always ends with the status zero, whereas printf() gives an exit of non-zero. The printf() is slower compared to echo.

E. What do these shell commands do? “ssh”, “scp” and “wget”. Describe briefly using an example that you have executed using the snowball server.

- ssh : is a snowball secure shell, it acts as a connection between two systems that is used to copy, manage or move files.
    - I have used the ssh serve to connect to the shell terminal to execute and write commands
- scp: means that the secure copy protocol, which means that securely transferring computer files between a local host and remote host
    - It is used a copy a file on a remote server to the computer. The scp is a tool used by the ssh network protocol.
    - I used this to transfer the files from and to snowball to my local machine in the power shell.
- wget: it’s a utility that helps us to download files from the web
    - I have used the wget command in the snowball serve to download the single file onto the ubuntu machine.

Answers

Question 1

Run the program by typing the following command

./helpme.sh

The 10 commands that I used are

sudo

cat

awk

ls

pwd

sed

echo

mkdir

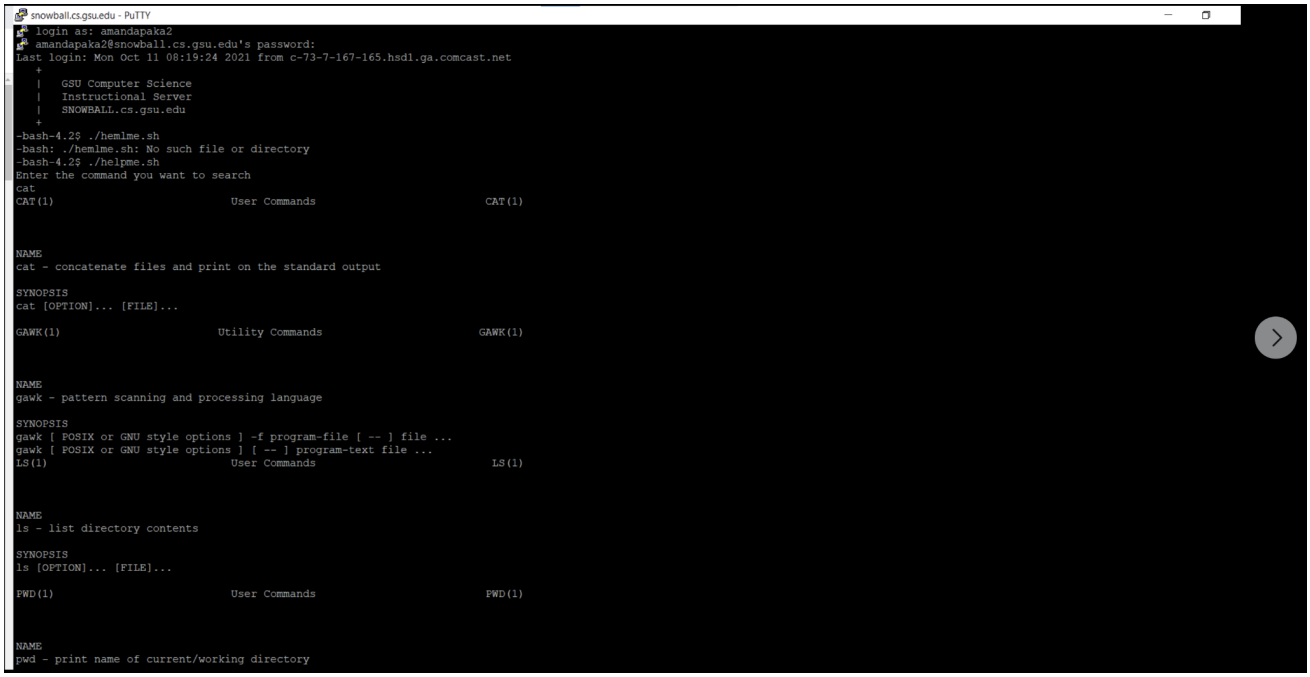
grep

chmod

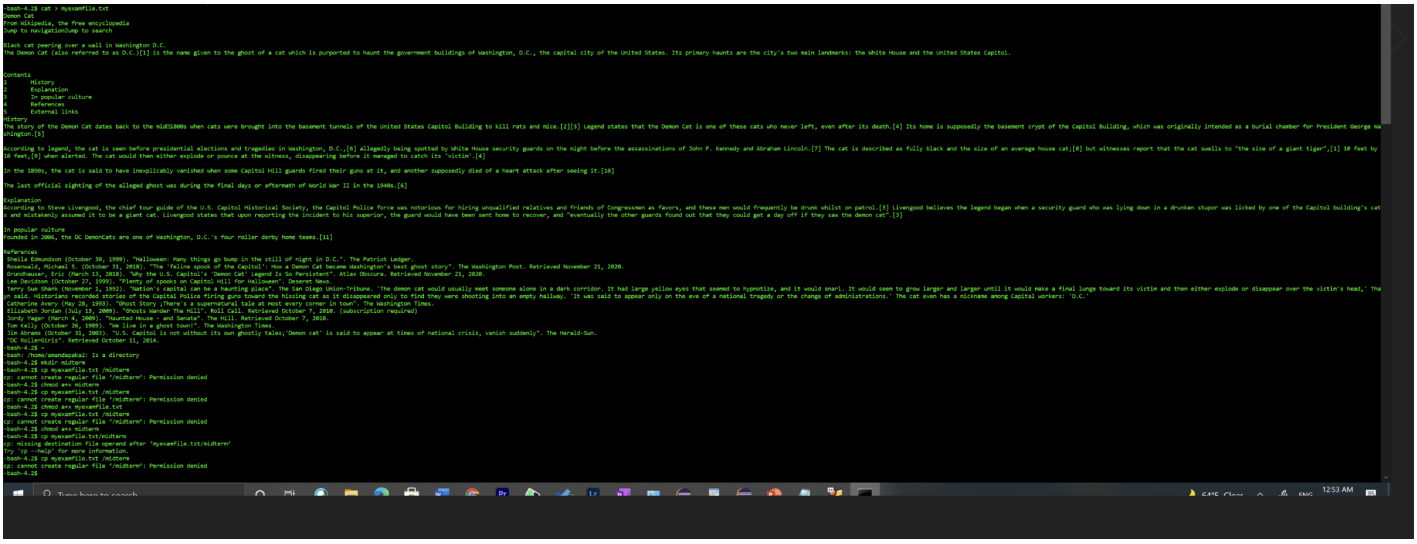
```
#!/bin/bash
echo Enter the command you want to search
read com \c
x=1
condition="$(grep -i ^$com\( mandatabase.txt | wc -l)"
if [ $condition -ge $x ]
then

n="0"
while read line; do
if [[ $line == ${com^^}"(* ]]
then
 echo "$line"
 n="1"
elif [[ "$n" -eq "1" && $line != *${com^^}"(1)" ]]
then
```

```
 echo"$line"
elif [[ "$n" -eq "1" && $line == *${com^^}"(1)" ]]
then
 echo "$line"
 n="0"
fi
done < mandatabase.txt
else
  echo "sorry, I cannot help you"
fi
```



Question 2


bash-4.2$ cat > myexamfile.txt
Demon Cat
From Wikipedia, the free encyclopedia
Jump to navigationJump to search
Black cat peering over a wall in Washington D.C.
The Demon Cat (also referred to as D.C.)[1] is the name given to the ghost of a cat which is purported to haunt the government buildings of Washington, D.C., the capital city of the United States. Its primary haunts are the city's two main landmarks: the White House and the United States Capitol.
Contents
1 History
2 Explanation
3 In popular culture
4 References
5 External links
History
The story of the Demon Cat dates back to the mid1800s when cats were brought into the basement tunnels of the United States Capitol Building to kill rats and mice.[2][3] Legend states that the Demon Cat is one of these cats who never left, even after its death.[4] Its home is supposedly the basement crypt of the Capitol Building, which was originally intended as a burial chamber for President George Washington.[5]
According to legend, the cat is seen before presidential elections and tragedies in Washington, D.C.,[6] allegedly being spotted by White House security guards on the night before the assassinations of John F. Kennedy and Abraham Lincoln.[7] The cat is described as fully black and the size of an average house cat;[8] but witnesses report that the cat swells to "the size of a giant tiger",[1] 10 feet by 10 feet,[9] when alerted. The cat would then either explode or pounce at the witness, disappearing before it managed to catch its 'victim'.[4]
In the 1890s, the cat is said to have inexplicably vanished when some Capitol Hill guards fired their guns at it, and another supposedly died of a heart attack after seeing it.[10]
The last official sighting of the alleged ghost was during the final days or aftermath of World War II in the 1940s.[6]
Explanation
According to Steve Livengood, the chief tour guide of the U.S. Capitol Historical Society, the Capitol Police force was notorious for hiring unqualified relatives and friends of Congressmen as favors, and these men would frequently be drunk whilst on patrol.[3] Livengood believes the legend began when a security guard who was lying down in a drunken stupor was licked by one of the Capitol building's cat
s and mistakenly assumed it to be a giant cat. Livengood states that upon reporting the incident to his superior, the guard would have been sent home to recover, and "eventually the other guards found out that they could get a day off if they saw the demon cat".[3]
In popular culture
Founded in 2006, the DC DemonCats are one of Washington, D.C.'s four roller derby home teams.[11]
References
Sheila Edmundson (October 30, 1999). "Halloween: Many things go bump in the still of night in D.C.". The Patriot Ledger.
Rosenwald, Michael S. (October 31, 2018). "The 'feline spook of the Capitol': How a Demon Cat became Washington's best ghost story". The Washington Post. Retrieved November 21, 2020.
Grundhauser, Eric (March 13, 2018). "Why the U.S. Capitol's 'Demon Cat' Legend Is So Persistent". Atlas Obscura. Retrieved November 21, 2020.
Lee Davidson (October 27, 1999). "Plenty of spooks on Capitol Hill for Halloween". Deseret News.
Terry Sue Shank (November 1, 1991). "Nation's capital can be a haunting place". The San Diego Union-Tribune. 'The demon cat would usually meet someone alone in a dark corridor. It had large yellow eyes that seemed to hypnotize, and it would snarl. It would seem to grow larger and larger until it would make a final lunge toward its victim and then either explode or disappear over the victim's head,' The
on said. Historians recorded stories of the Capitol Police firing guns toward the hissing cat as it disappeared only to find they were shooting into an empty hallway. 'It was said to appear only on the eve of a national tragedy or the change of administrations.' The cat even has a nickname among Capital workers: 'D.C.'
Catherine Avery (May 28, 1993). "Ghost Story ;There's a supernatural tale at most every corner in town". The Washington Times.
Elizabeth Jordan (July 13, 2009). "Ghosts Wander The Hill". Roll Call. Retrieved October 7, 2010. (subscription required)
Jordy Yager (March 4, 2008). "Haunted House - and Senate". The Hill. Retrieved October 7, 2010.
Tom Kelly (October 26, 1989). "We live in a ghost town!". The Washington Times.
Ian Abrams, (October 31, 2003). "U.S. Capitol is not without its own ghostly tales;'Demon cat' is said to appear at times of national crisis, vanish suddenly". The Herald-Sun.
"DC RollerGirls". Retrieved October 11, 2014.
bash-4.2$ ~
bash: /home/amandapeka2: Is a directory
bash-4.2$ mkdir midterm
bash-4.2$ cp myexamfile.txt /midterm
cp: cannot create regular file '/midterm': Permission denied
bash-4.2$ chmod a+x midterm
bash-4.2$ cp myexamfile.txt /midterm
cp: cannot create regular file '/midterm': Permission denied
bash-4.2$ chmod a+x myexamfile.txt
bash-4.2$ cp myexamfile.txt /midterm
cp: cannot create regular file '/midterm': Permission denied
bash-4.2$ chmod a+x midterm
bash-4.2$ cp myexamfile.txt/midterm
cp: missing destination file operand after 'myexamfile.txt/midterm'
Try `cp --help' for more information.
bash-4.2$ cp myexamfile.txt /midterm
cp: cannot create regular file '/midterm': Permission denied
bash-4.2$
12:53 AM

Question 2, part A

OpenSSH SSH client

```
#!/bin/bash


function find_statements() {

        i=0

        while IFS=. read -r -a line
        do
        IFS="."

                for stat in ${line[@]}
                do

                        statements[i]=$(echo $stat | sed 's/^ //g')

                        i=$((i+1))
                done
        done < $1

        echo "There are total ${#statements[@]} statements"
}


if(($#==1))
then

        file=$1
else
        # Else, prompt the user for a file
        echo -n "Text file: "
        read file
fi

# Check if the file exists
if [ -f $file ]
then

        find_statements $file
else

        echo "$file: file not exist"
        exit 1
fi
~
~
~
~
~
~
~
~
~
~
~
```

```
-bash-4.2$ vi question2.sh
-bash-4.2$ -bash-4.2$ ./question2.sh myexamfile.txt
There are total 98 statements
-bash-4.2$
```

Question 2 part B (updated the above script)

```
OpenSSH SSH client
#!/bin/bash

function fnd_statements() {
        # Initialize array index counter
        i=0

        while CLT=, read -r -a line
        do
        CLT="."

                for stat in ${line[@]}
                do

                        statements[i]=$(echo $stat | sed 's/^ //g')
                        # Increment counter
                        i=$((i+1))
                done
        done < $1

        # Display the table header
        echo -e "\t\tWords\tletters"
        # Initilize the array counter
        i=1


        for statement in ${statements[@]}
        do
                # Count number of words and letters
                words=$(echo $statement | wc -w)
                letters=$(echo $statement | tr -d ' ' | wc -c)
                # Display the words and letters count
                echo -e "Statement $i:\t  $words\t  $((letters -1))"

                i=$((i+1))
        done
}

# Check if argument is passed
if(($#==1))
then
        # Then, set file to $1
        file=$1
else
        # Else, prompt the user for a file
        echo -n "Text file: "
        read file
fi

# Check if the file exists
if [ -f $file ]
then
        # Then, call the function
        fnd_statements $file
else
        # Else, display error and exit
        echo "$file: file not exist"
        exit 1
fi
~
~
~
~
~
~
```

Output for above code

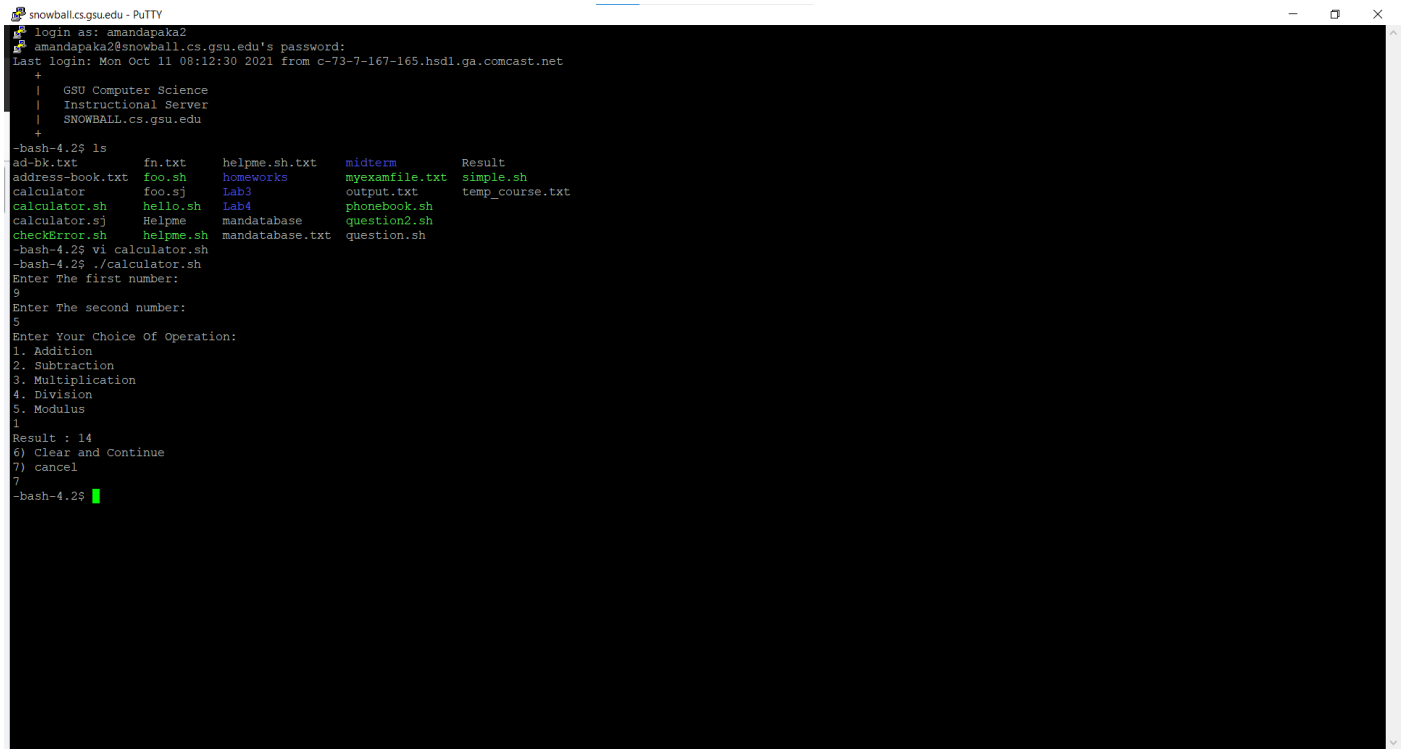
```
OpenSSH SSH client
-bash-4.2$ vi question2.sh
-bash-4.2$ -bash-4.2$ ./question2.sh myexamfile.txt
There are total 98 statements
-bash-4.2$ vi question2.sh
-bash-4.2$ -bash-4.2$ ./question2.sh myexamfile.txt
There are total 98 statements
-bash-4.2$ vi question2.sh
-bash-4.2$ ./question2.sh myexamfile.txt
                Words   letters
Statement 1:    2       8
Statement 2:    5       33
Statement 3:    5       28
Statement 4:    9       37
Statement 5:    1       1
Statement 6:    8       29
Statement 7:    1       1
Statement 8:    22      93
Statement 9:    1       1
Statement 10:   8       32
Statement 11:   17      83
Statement 12:   1       8
Statement 13:   2       9
Statement 14:   2       13
Statement 15:   4       18
Statement 16:   2       12
Statement 17:   3       15
Statement 18:   1       7
Statement 19:   30      136
Statement 20:   19      79
Statement 21:   24      129
Statement 22:   1       3
Statement 23:   15      83
Statement 24:   1       1
Statement 25:   18      91
Statement 26:   4       24
Statement 27:   37      156
Statement 28:   19      94
Statement 29:   1       3
Statement 30:   32      144
Statement 31:   1       4
Statement 32:   22      94
Statement 33:   1       3
Statement 34:   1       11
Statement 35:   11      49
Statement 36:   1       1
Statement 37:   29      166
Statement 38:   36      158
Statement 39:   39      176
Statement 40:   1       3
Statement 41:   3       16
Statement 42:   11      48
Statement 43:   1       1
Statement 44:   6       26
Statement 45:   1       4
Statement 46:   1       10
Statement 47:   5       31
Statement 48:   12      47
Statement 49:   1       1
Statement 50:   1       1
Statement 51:   3       16
Statement 52:   3       18
Statement 53:   3       16
Statement 54:   15      75
Statement 55:   3       17
Statement 56:   4       24
Statement 57:   5       30
Statement 58:   3       8
Statement 59:   1       1
Statement 60:   7       40
Statement 61:   2       12
Statement 62:   4       24
Statement 63:   5       27
Statement 64:   8       41
Statement 65:   2       11
Statement 66:   6       29
```


OpenSSH SSH client
Statement 66: 6 29
Statement 67: 7 36
Statement 68: 4 24
Statement 69: 12 55
Statement 70: 13 57
Statement 71: 30 135
Statement 72: 26 131
Statement 73: 18 76
Statement 74: 11 45
Statement 75: 1 1
Statement 76: 1 1
Statement 77: 5 26
Statement 78: 12 60
Statement 79: 3 18
Statement 80: 5 28
Statement 81: 4 21
Statement 82: 2 8
Statement 83: 4 22
Statement 84: 2 22
Statement 85: 5 23
Statement 86: 5 24
Statement 87: 2 7
Statement 88: 4 22
Statement 89: 5 24
Statement 90: 6 21
Statement 91: 3 18
Statement 92: 5 25
Statement 93: 1 2
Statement 94: 1 1
Statement 95: 20 101
Statement 96: 2 13
Statement 97: 2 15
Statement 98: 4 23
-bash-4.2$

Question 3

To execute the file, type in the following command

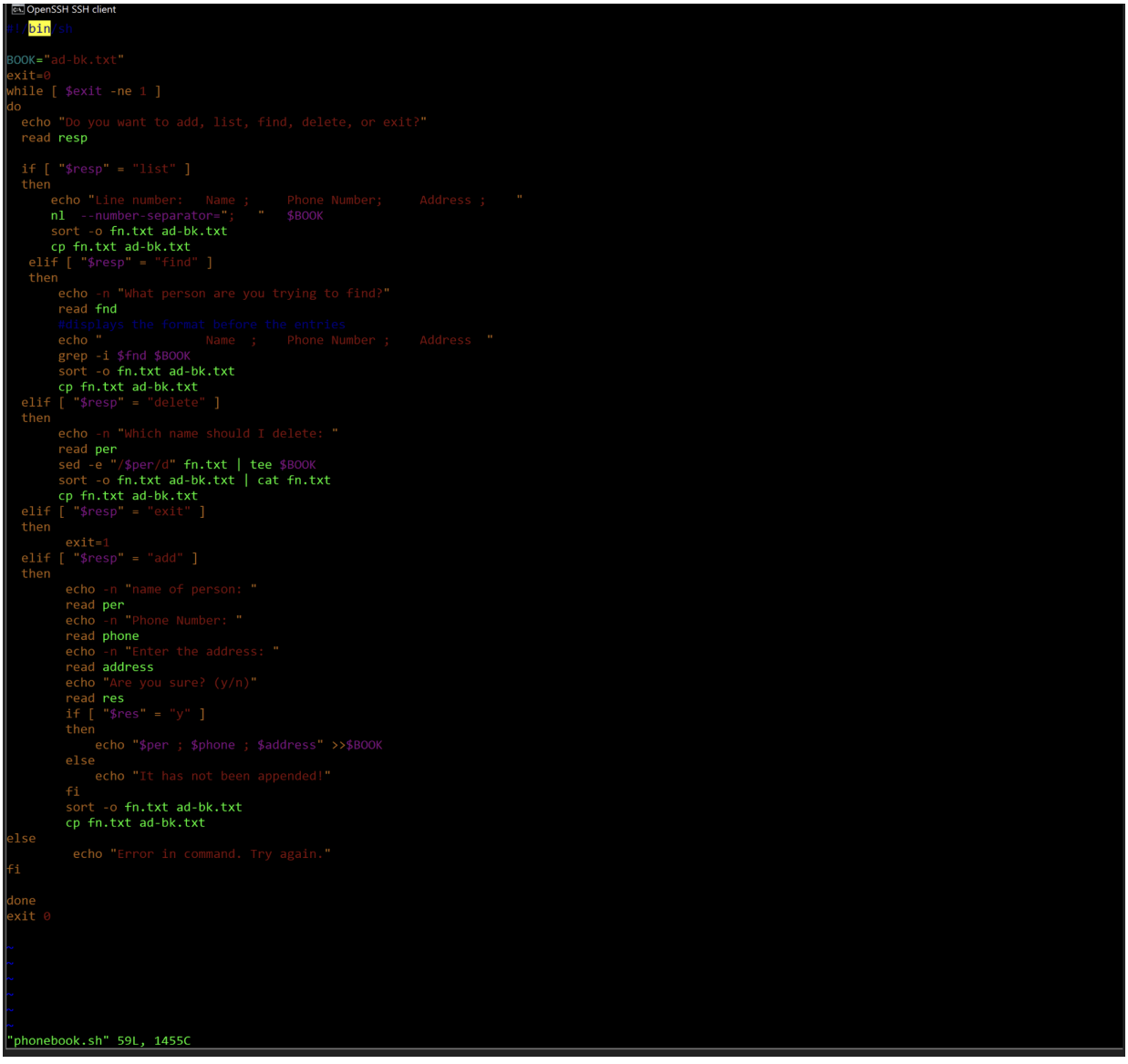
./calculator.sh



Question 4

To execute this type the following command

./phonebook.sh



```
OpenSSH SSH client
#!/bin/sh

BOOK="ad-bk.txt"
exit=0
while [ $exit -ne 1 ]
do
  echo "Do you want to add, list, find, delete, or exit?"
  read resp

  if [ "$resp" = "list" ]
  then
      echo "Line number:   Name ;     Phone Number;     Address ;    "
      nl  --number-separator=";   "   $BOOK
      sort -o fn.txt ad-bk.txt
      cp fn.txt ad-bk.txt
   elif [ "$resp" = "find" ]
   then
       echo -n "What person are you trying to find?"
       read fnd
       #displays the format before the entries
       echo "              Name  ;    Phone Number ;    Address  "
       grep -i $fnd $BOOK
       sort -o fn.txt ad-bk.txt
       cp fn.txt ad-bk.txt
  elif [ "$resp" = "delete" ]
  then
       echo -n "Which name should I delete: "
       read per
       sed -e "/$per/d" fn.txt | tee $BOOK
       sort -o fn.txt ad-bk.txt | cat fn.txt
       cp fn.txt ad-bk.txt
  elif [ "$resp" = "exit" ]
  then
        exit=1
  elif [ "$resp" = "add" ]
  then
       echo -n "name of person: "
       read per
       echo -n "Phone Number: "
       read phone
       echo -n "Enter the address: "
       read address
       echo "Are you sure? (y/n)"
       read res
       if [ "$res" = "y" ]
       then
           echo "$per ; $phone ; $address" >>$BOOK
       else
           echo "It has not been appended!"
       fi
       sort -o fn.txt ad-bk.txt
       cp fn.txt ad-bk.txt
else
        echo "Error in command. Try again."
fi

done
exit 0

~
~
~
~
~
~
"phonebook.sh" 59L, 1455C
```

Output

```
-bash-4.2$ ./phonebook.sh
Do you want to add, list, find, delete, or exit?
add
name of person: Kristen
Phone Number: 2028765409
Enter the address: 276 Oak Creek, Johns Creek, GA
Are you sure? (y/n)
y
Do you want to add, list, find, delete, or exit?
add
name of person: Fam
Phone Number: 9098907890
Enter the address: 456 Fram Street, Pearls, NV
Are you sure? (y/n)
n
It has not been appended!
Do you want to add, list, find, delete, or exit?
list
Line number:   Name ;     Phone Number;     Address ;
     1;   Helen ; 4048790876 ; 2345 welfare crossing, Forsyth, GA
     2;   Kristen ; 2028765409 ; 276 Oak Creek, Johns Creek, GA
     3;   Sam ; 2340987654 ; 1324 Goldmine Street, Alpharetta, GA
Do you want to add, list, find, delete, or exit?
find
What person are you trying to find?Sam
             Name  ;    Phone Number ;    Address
Sam ; 2340987654 ; 1324 Goldmine Street, Alpharetta, GA
Do you want to add, list, find, delete, or exit?
delete
Which name should I delete: Helen
Kristen ; 2028765409 ; 276 Oak Creek, Johns Creek, GA
Sam ; 2340987654 ; 1324 Goldmine Street, Alpharetta, GA
Helen ; 4048790876 ; 2345 welfare crossing, Forsyth, GA
Kristen ; 2028765409 ; 276 Oak Creek, Johns Creek, GA
Sam ; 2340987654 ; 1324 Goldmine Street, Alpharetta, GA
Do you want to add, list, find, delete, or exit?
list
Line number:   Name ;     Phone Number;     Address ;
     1;   Kristen ; 2028765409 ; 276 Oak Creek, Johns Creek, GA
     2;   Sam ; 2340987654 ; 1324 Goldmine Street, Alpharetta, GA
Do you want to add, list, find, delete, or exit?
exit
-bash-4.2$
```