# 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: Kevin Gallardo

Campus ID: kgallardowepster1

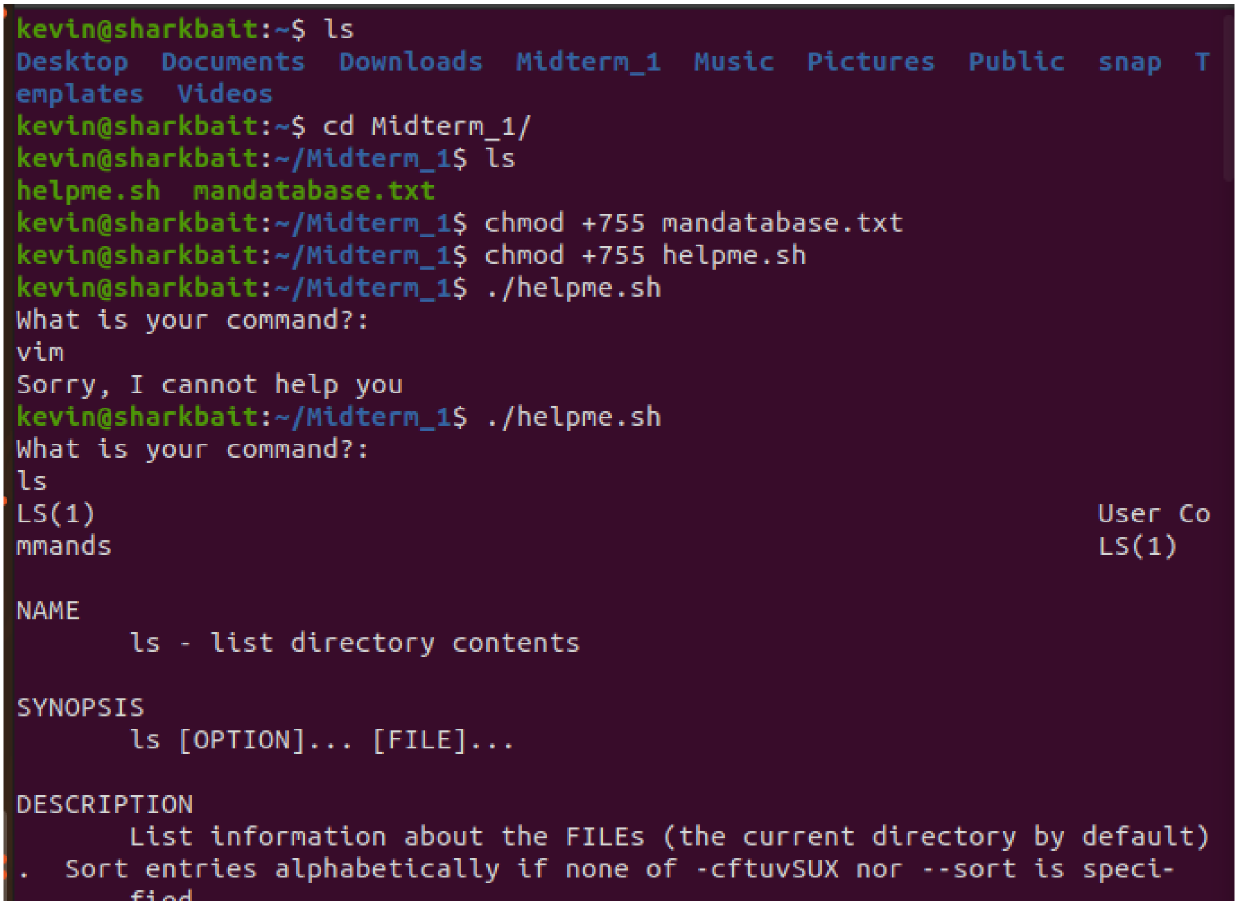
Panther #: 002-49-7849

**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*

*The commands in mandatabase.txt(text file should be in same directory as shell script):*
*ls*
*find*
*pwd*
*touch*
*cat*
*rm*
*cp*
*rmdir*
*mv*
*Mkdir*

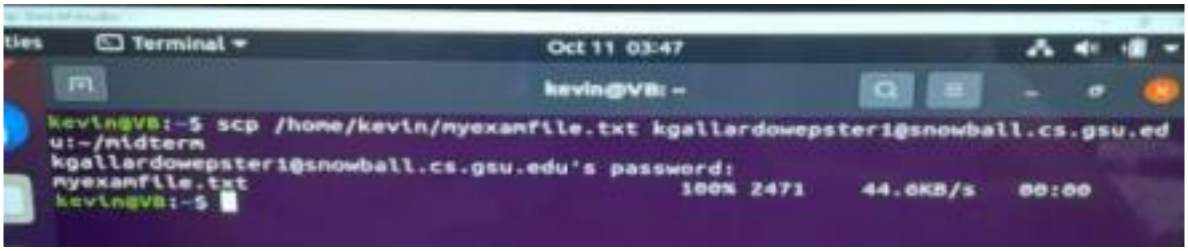
*Step 1) chmod +755 madatabase.txt*
*step 2) chmod +755 helpme.sh*
*Step 3) ./helpme.sh*
*step 4) vim*
*Step 5) ls*



```
kevin@sharkbait:~$ ls
Desktop  Documents  Downloads  Midterm_1  Music  Pictures  Public  snap  T
emplates  Videos
kevin@sharkbait:~$ cd Midterm_1/
kevin@sharkbait:~/Midterm_1$ ls
helpme.sh  mandatabase.txt
kevin@sharkbait:~/Midterm_1$ chmod +755 mandatabase.txt
kevin@sharkbait:~/Midterm_1$ chmod +755 helpme.sh
kevin@sharkbait:~/Midterm_1$ ./helpme.sh
What is your command?:
vim
Sorry, I cannot help you
kevin@sharkbait:~/Midterm_1$ ./helpme.sh
What is your command?:
ls
LS(1)                                                             User Co
mmands                                                            LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List information about the FILEs (the current directory by default)
.  Sort entries alphabetically if none of -cftuvSUX nor --sort is speci-
       fied
```

*For question one, I used the touch command to make the txt file.*
*After that, I used the manual command for each command redirected and appended the standard output to the txt file. Shell script checks user input against each case to see if there is a match. If there is a match it will output the corresponding manual from the man database, since each manual starts with the format COMMAND(1) (in uppercase letters, at the beginning of line) and ends with COMMAND(1)*
*# (in uppercase letters, at the end of the line).*

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).



```
kevin@VB:~$ scp /home/kevin/myexamfile.txt kgallardowepster1@snowball.cs.gsu.ed
u:~/midterm
kgallardowepster1@snowball.cs.gsu.edu's password:
myexamfile.txt                                100% 2471     44.6KB/s   00:00
kevin@VB:~$ 
```

Followed the instructions from above.

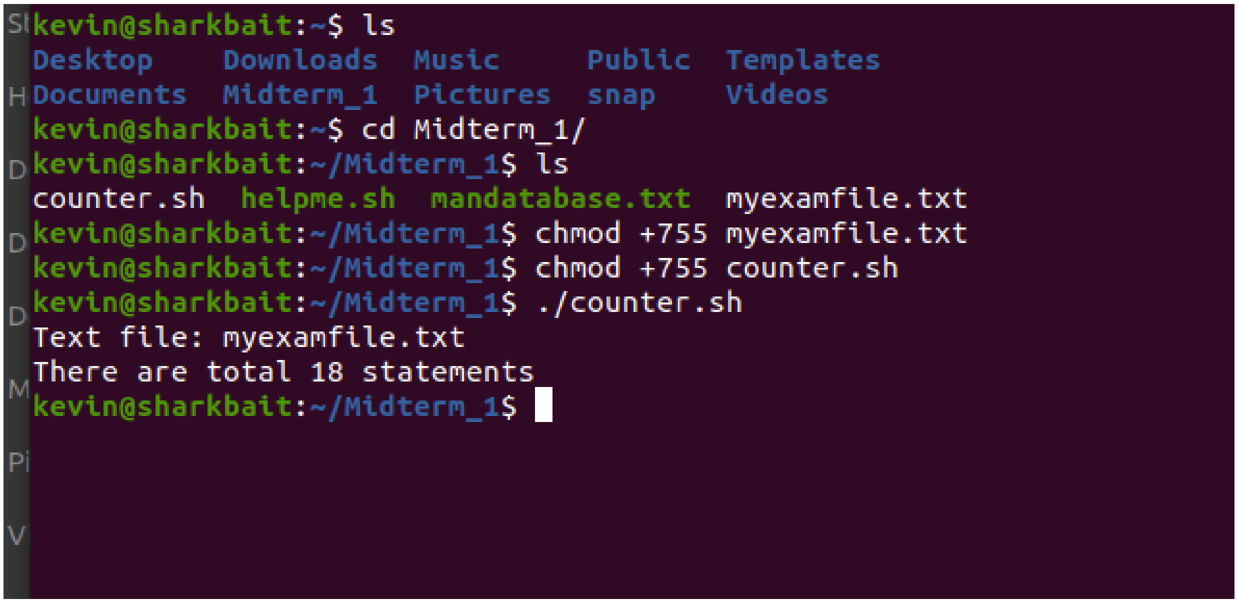
a. Write a shell script that will find the number of statements in the text. A statement is defined as the collection of text between two periods (full-stops).

Step 1) chmod +755 myexamfile.txt

Step 2) chmod +755 counter.sh

Step 3) ./counter.sh

Step 4) myexamfile.txt



```
kevin@sharkbait:~$ ls
Desktop    Downloads  Music     Public  Templates
Documents  Midterm_1  Pictures  snap    Videos
kevin@sharkbait:~$ cd Midterm_1/
kevin@sharkbait:~/Midterm_1$ ls
counter.sh  helpme.sh  mandatabase.txt  myexamfile.txt
kevin@sharkbait:~/Midterm_1$ chmod +755 myexamfile.txt
kevin@sharkbait:~/Midterm_1$ chmod +755 counter.sh
kevin@sharkbait:~/Midterm_1$ ./counter.sh
Text file: myexamfile.txt
There are total 18 statements
kevin@sharkbait:~/Midterm_1$ 
```

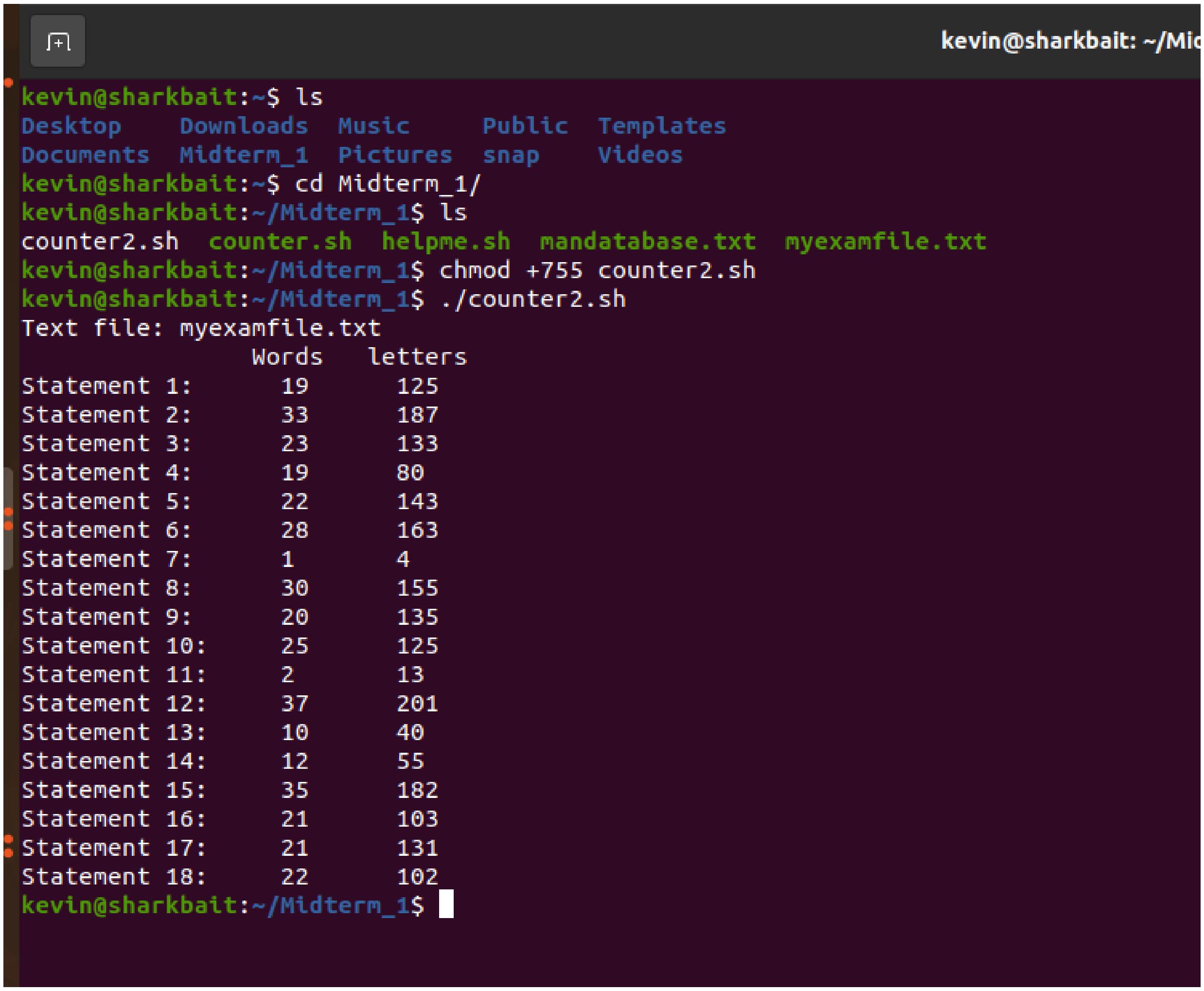
Approach: Created a function that reads lines and made the statements count as one if it was between two periods.

b. Update the script to present a tabular list that shows the number of words and number of letters in each statement.

Step 1) chmod +755 couter2.sh

Step 2) ./counter.sh

Step 3) myexamfile.txt

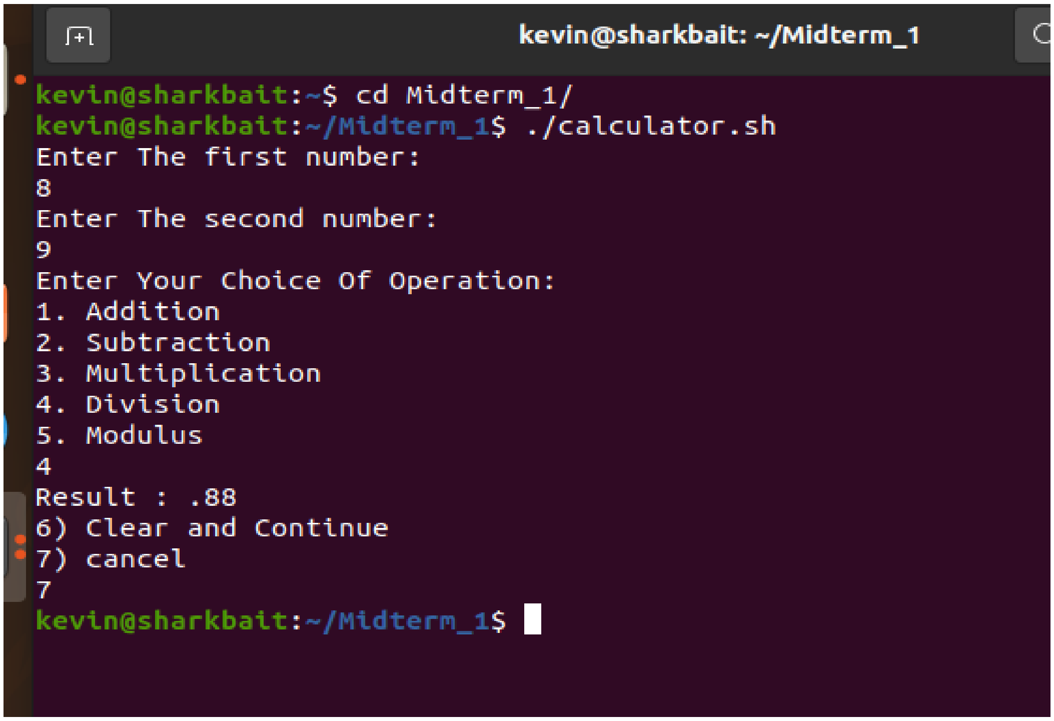


```
kevin@sharkbait:~$ ls
Desktop    Downloads  Music     Public  Templates
Documents  Midterm_1  Pictures  snap    Videos
kevin@sharkbait:~$ cd Midterm_1/
kevin@sharkbait:~/Midterm_1$ ls
counter2.sh  counter.sh  helpme.sh  mandatabase.txt  myexamfile.txt
kevin@sharkbait:~/Midterm_1$ chmod +755 counter2.sh
kevin@sharkbait:~/Midterm_1$ ./counter2.sh
Text file: myexamfile.txt
                Words   letters
Statement 1:      19      125
Statement 2:      33      187
Statement 3:      23      133
Statement 4:      19      80
Statement 5:      22      143
Statement 6:      28      163
Statement 7:      1       4
Statement 8:      30      155
Statement 9:      20      135
Statement 10:     25      125
Statement 11:     2       13
Statement 12:     37      201
Statement 13:     10      40
Statement 14:     12      55
Statement 15:     35      182
Statement 16:     21      103
Statement 17:     21      131
Statement 18:     22      102
kevin@sharkbait:~/Midterm_1$ 
```

Approach: Made a for loop to print the statement number and as the statement is being counter so are the number of letters and words and being printed to the screen.

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.

1) Chmod +755 calculator.sh
2) ./calculator.sh
3) 8
4) 9
5) 4
6) 7

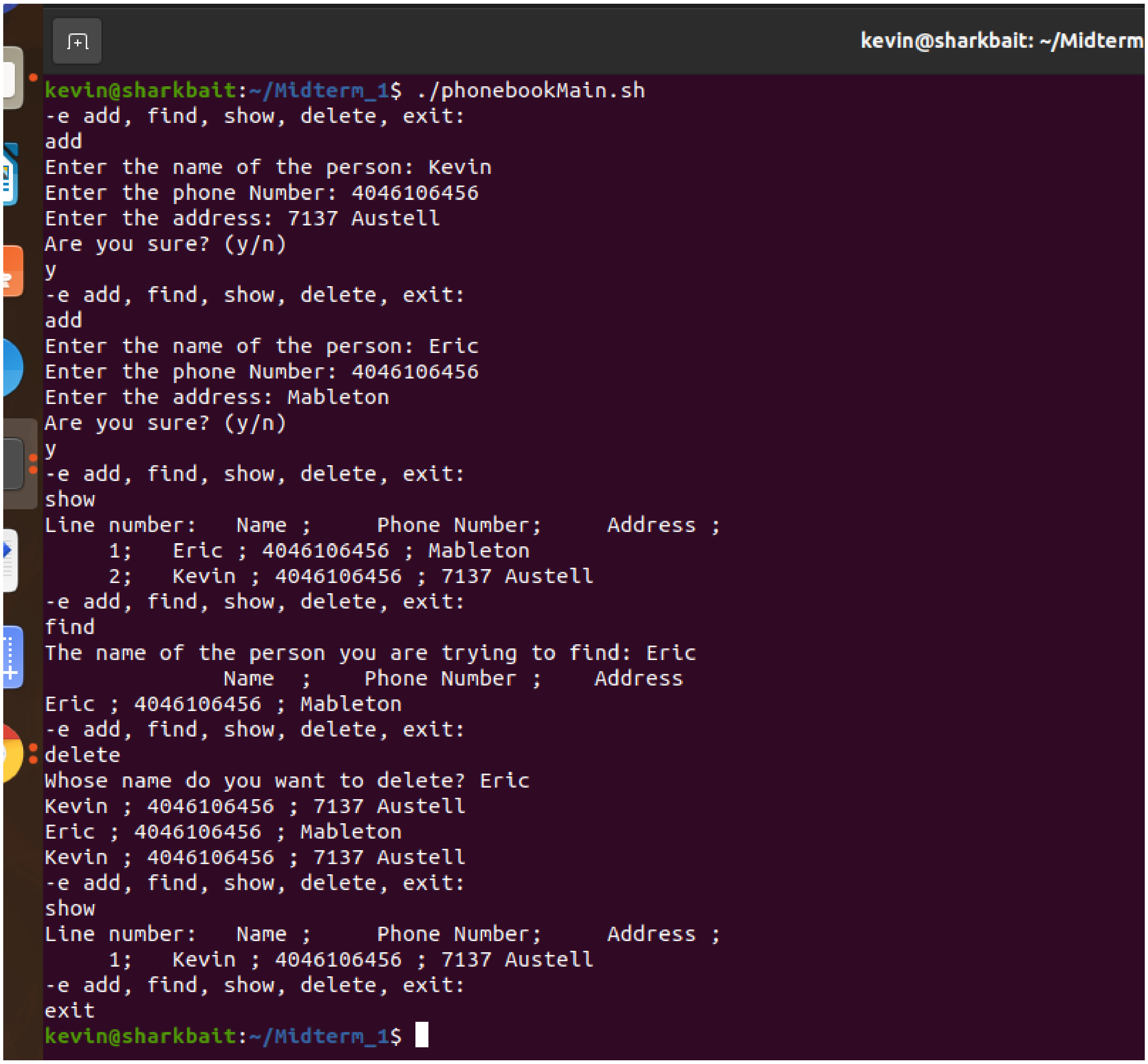


```
kevin@sharkbait:~$ cd Midterm_1/
kevin@sharkbait:~/Midterm_1$ ./calculator.sh
Enter The first number:
8
Enter The second number:
9
Enter Your Choice Of Operation:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Modulus
4
Result : .88
6) Clear and Continue
7) cancel
7
kevin@sharkbait:~/Midterm_1$
```

Approach: Take user input. Take one number at a time and then ask for what operation the user wants. Compare to cases and apply the corresponding operation. Ask users if they want to clear and continue or cancel and end the program.

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).

1) Chmod +755 phonebookMain.sh
2) Chmod +755 createEntry.sh
3) Chmod +755 deleteEntry.sh
4) Chmod +755 getEntry.sh
5) Chmod +755 showEntry.sh
6) ./phonebookMain.sh
7) add
8) Kevin Gallardo
9) 7137 Silver Mine Xing Austell GA 30168
10) Y
11) Show
12) Find
13) Kevin
14) Exit



```
kevin@sharkbait:~/Midterm_1$ ./phonebookMain.sh
-e add, find, show, delete, exit:
add
Enter the name of the person: Kevin
Enter the phone Number: 4046106456
Enter the address: 7137 Austell
Are you sure? (y/n)
y
-e add, find, show, delete, exit:
add
Enter the name of the person: Eric
Enter the phone Number: 4046106456
Enter the address: Mableton
Are you sure? (y/n)
y
-e add, find, show, delete, exit:
show
Line number:   Name ;     Phone Number;     Address ;
     1;   Eric ; 4046106456 ; Mableton
     2;   Kevin ; 4046106456 ; 7137 Austell
-e add, find, show, delete, exit:
find
The name of the person you are trying to find: Eric
              Name  ;    Phone Number ;    Address
Eric ; 4046106456 ; Mableton
-e add, find, show, delete, exit:
delete
Whose name do you want to delete? Eric
Kevin ; 4046106456 ; 7137 Austell
Eric ; 4046106456 ; Mableton
Kevin ; 4046106456 ; 7137 Austell
-e add, find, show, delete, exit:
show
Line number:   Name ;     Phone Number;     Address ;
     1;   Kevin ; 4046106456 ; 7137 Austell
-e add, find, show, delete, exit:
exit
kevin@sharkbait:~/Midterm_1$
```

Approach: Created the main shell to echo the commands available to the user. The user chooses a command and then the main shell goes into a while loop calling different functions until the user chooses to exit. The main shell calls the supporting shells to add find show delete entries and has an exit function.

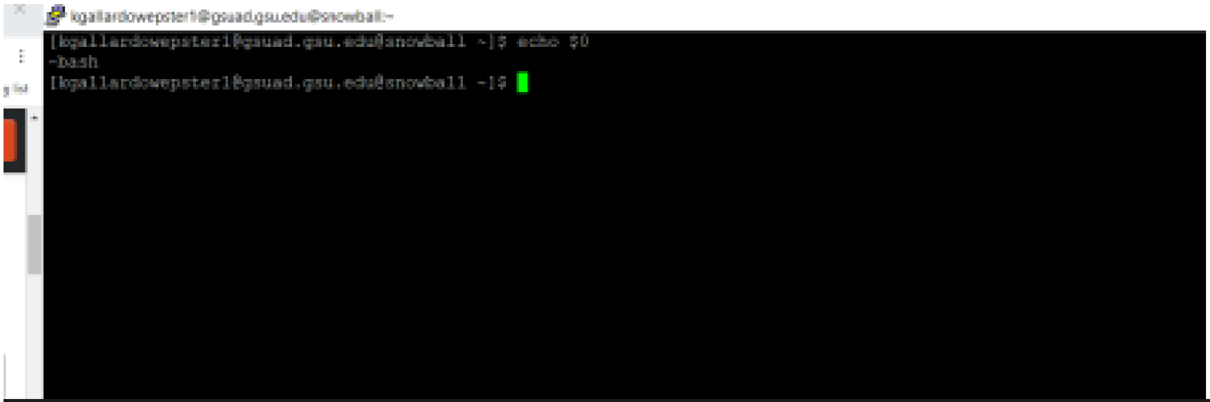
5. (4 pts each) Give brief answers with examples, wherever relevant

A. What is the use of a shell?

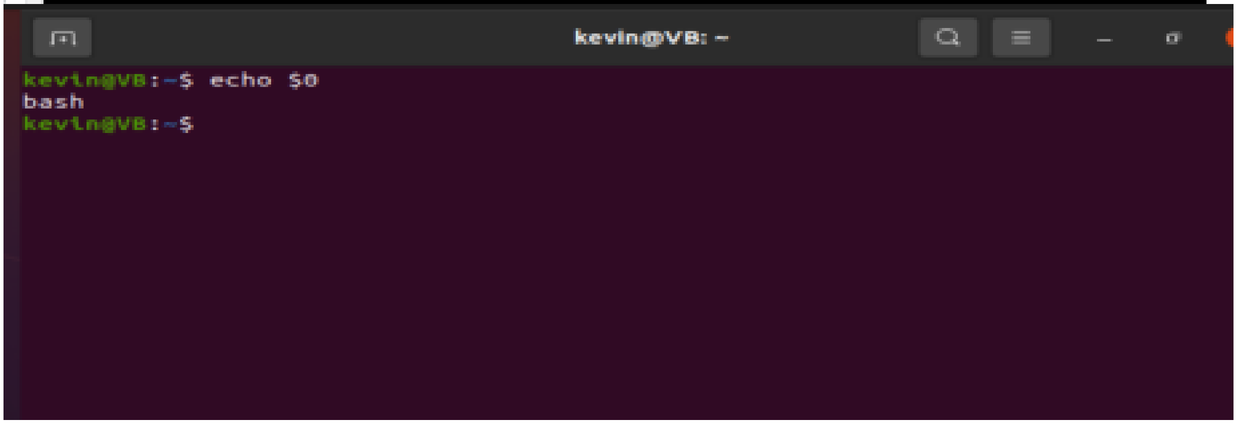
It interprets commands for meaning received from standard input or command-line argument or redirected standard output. Those commands are translated in order to interact with the system.

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.

Both the server and my virtual machine running Ubuntu have the same shell. There is no difference between them.



```
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$ echo $0
-bash
[kgallardowepster1@gsuad.gsu.edu@snowball ~]$
```



```
kevin@VB:~$ echo $0
bash
kevin@VB:~$
```

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

The software would be the compiler that converts a C program into machine code and the CPU would be the hardware that carries out the machine code.

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 command echo has a built-in newline character whereas the command print() is added manually with \n.

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 secured shell and it’s used to connect two systems. In the snow server, we always connect using Ssh and then give out username and password. This enables the user to execute commands remotely.