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: Ramey Serdah

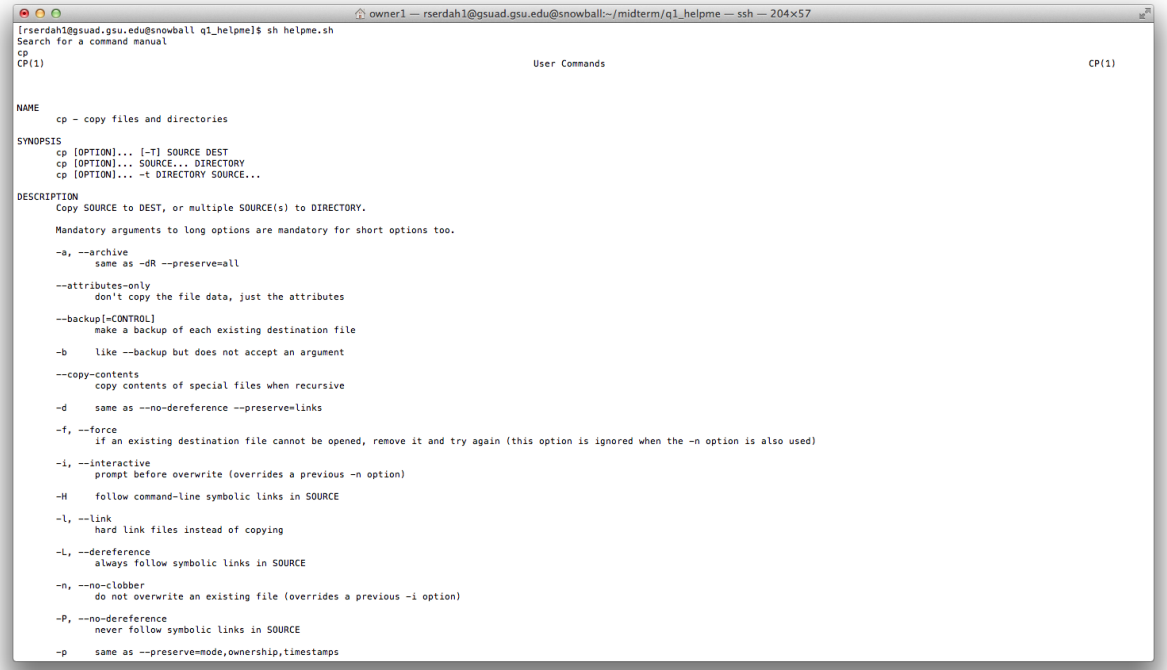
Campus ID: rserdah1

Panther #: 002 48 9675

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

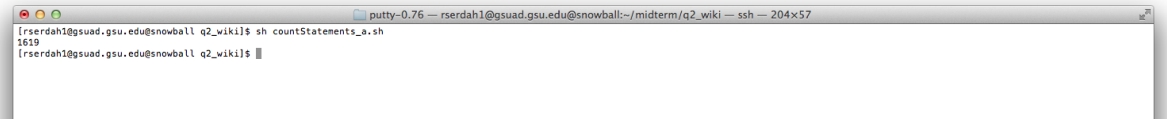


```
owner1 — rserdah1@gsuad.gsu.edu@snowball:~/midterm/q1_helpme — ssh — 204×57
[rserdah1@gsuad.gsu.edu@snowball q1_helpme]$ sh helpme.sh
Search for a command manual
cp
CP(1)                                                User Commands                                                CP(1)


NAME
       cp - copy files and directories

SYNOPSIS
       cp [OPTION]... [-T] SOURCE DEST
       cp [OPTION]... SOURCE... DIRECTORY
       cp [OPTION]... -t DIRECTORY SOURCE...

DESCRIPTION
       Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

       Mandatory arguments to long options are mandatory for short options too.

       -a, --archive
              same as -dR --preserve=all

       --attributes-only
              don't copy the file data, just the attributes

       --backup[=CONTROL]
              make a backup of each existing destination file

       -b     like --backup but does not accept an argument

       --copy-contents
              copy contents of special files when recursive

       -d     same as --no-dereference --preserve=links

       -f, --force
              if an existing destination file cannot be opened, remove it and try again (this option is ignored when the -n option is also used)

       -i, --interactive
              prompt before overwrite (overrides a previous -n option)

       -H     follow command-line symbolic links in SOURCE

       -l, --link
              hard link files instead of copying

       -L, --dereference
              always follow symbolic links in SOURCE

       -n, --no-clobber
              do not overwrite an existing file (overrides a previous -i option)

       -P, --no-dereference
              never follow symbolic links in SOURCE

       -p     same as --preserve=mode,ownership,timestamps
```

To run helpme.sh, use sh helpme.sh

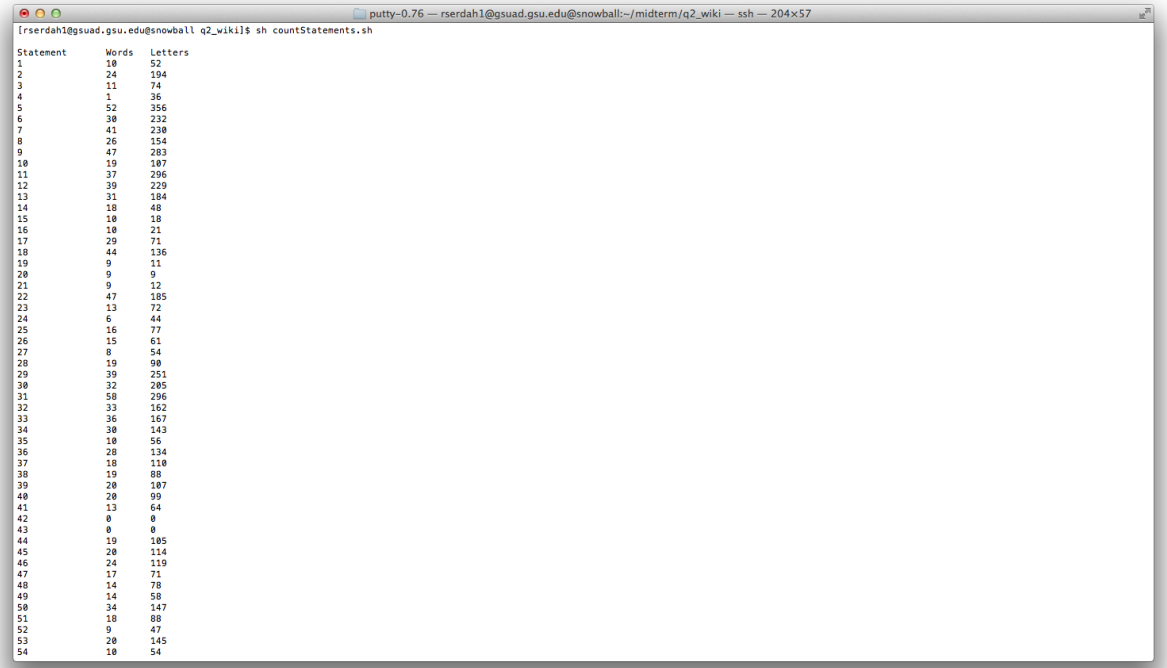
2. (10pts each) On your computer open your favourite Wikipedia page. Copy the text from that page into a text file **myexamfile.txt** and then copy that file to a directory named **midterm** (use mkdir to create the directory if it doesn't exist) in your snowball server home directory (use any FTP tool 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).



```
putty-0.76 — rserdah1@gsuad.gsu.edu@snowball:~/midterm/q2_wiki — ssh — 204×57
[rserdah1@gsuad.gsu.edu@snowball q2_wiki]$ sh countStatements_a.sh
1619
[rserdah1@gsuad.gsu.edu@snowball q2_wiki]$
```

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

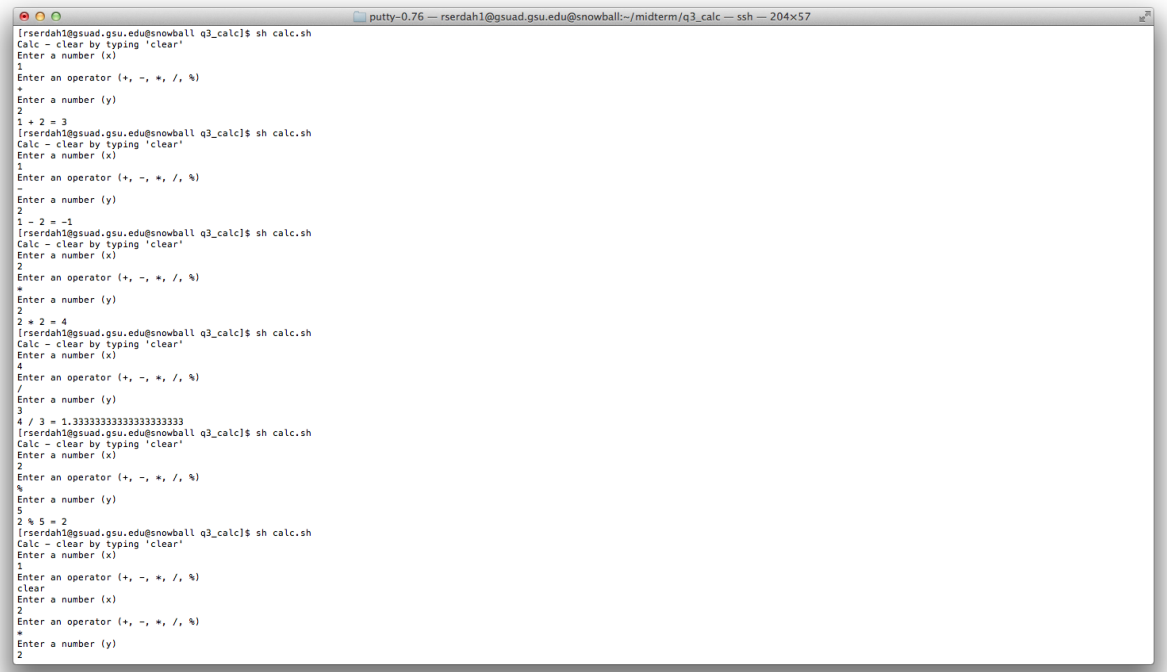


```
putty-0.76 — rserdah1@gsuad.gsu.edu@snowball:~/midterm/q2_wiki — ssh — 204×57
[rserdah1@gsuad.gsu.edu@snowball q2_wiki]$ sh countStatements.sh

Statement       Words   Letters
1               10      52
2               24      194
3               11      74
4               1       36
5               52      356
6               30      232
7               41      230
8               26      154
9               47      283
10              19      107
11              37      296
12              39      229
13              31      184
14              18      48
15              10      18
16              10      21
17              29      71
18              44      136
19              9       11
20              9       9
21              9       12
22              47      185
23              13      72
24              6       44
25              16      77
26              15      61
27              8       54
28              19      90
29              39      251
30              32      205
31              58      296
32              33      162
33              36      167
34              30      143
35              10      56
36              28      134
37              18      110
38              19      88
39              20      107
40              20      99
41              13      64
42              0       0
43              0       0
44              19      105
45              20      114
46              24      119
47              17      71
48              14      78
49              14      58
50              34      147
51              18      88
52              9       47
53              20      145
54              10      54
```

Run with `sh countStatements_a.sh` and `sh countStatements_a.sh`

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.

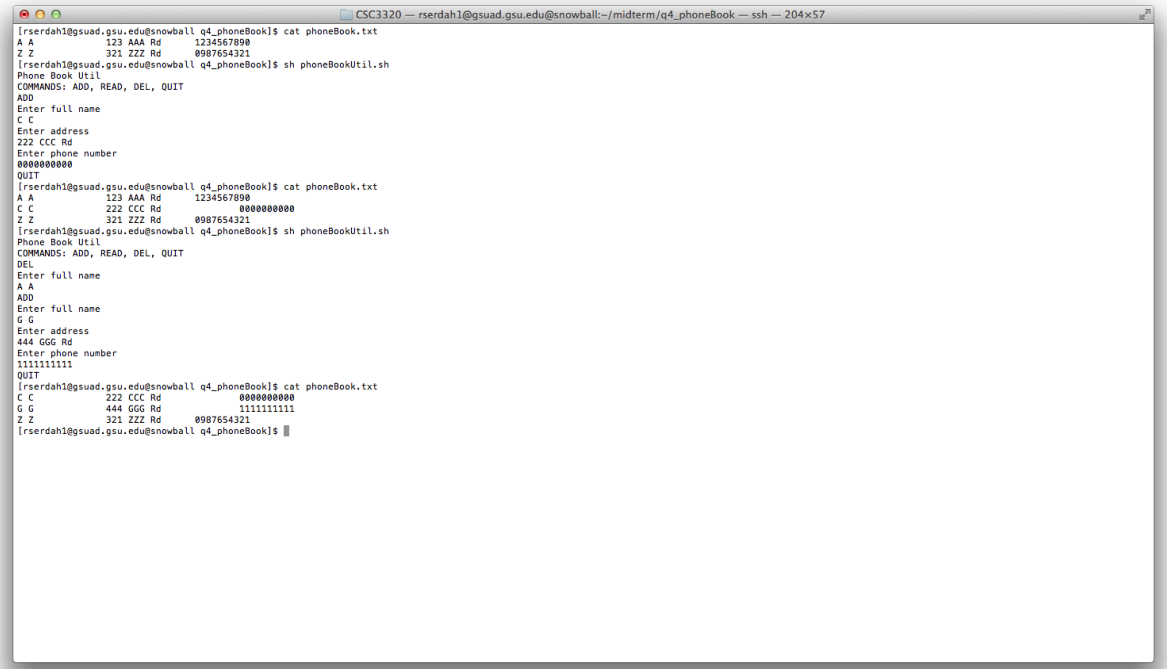
Run with `sh calc.sh`




```
[rserdah1@gsuad.gsu.edu@snowball q3_calc]$ sh calc.sh
Calc - clear by typing 'clear'
Enter a number (x)
1
Enter an operator (+, -, *, /, %)
+
Enter a number (y)
2
1 + 2 = 3
[rserdah1@gsuad.gsu.edu@snowball q3_calc]$ sh calc.sh
Calc - clear by typing 'clear'
Enter a number (x)
1
Enter an operator (+, -, *, /, %)
-
Enter a number (y)
2
1 - 2 = -1
[rserdah1@gsuad.gsu.edu@snowball q3_calc]$ sh calc.sh
Calc - clear by typing 'clear'
Enter a number (x)
2
Enter an operator (+, -, *, /, %)
*
Enter a number (y)
2
2 * 2 = 4
[rserdah1@gsuad.gsu.edu@snowball q3_calc]$ sh calc.sh
Calc - clear by typing 'clear'
Enter a number (x)
4
Enter an operator (+, -, *, /, %)
/
Enter a number (y)
3
4 / 3 = 1.3333333333333333333
[rserdah1@gsuad.gsu.edu@snowball q3_calc]$ sh calc.sh
Calc - clear by typing 'clear'
Enter a number (x)
2
Enter an operator (+, -, *, /, %)
%
Enter a number (y)
5
2 % 5 = 2
[rserdah1@gsuad.gsu.edu@snowball q3_calc]$ sh calc.sh
Calc - clear by typing 'clear'
Enter a number (x)
1
Enter an operator (+, -, *, /, %)
clear
Enter a number (x)
2
Enter an operator (+, -, *, /, %)
*
Enter a number (y)
2
```

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).

Run with `sh phoneBookUtil.sh`



```
CSC3320 — rserdah1@gsuad.gsu.edu@snowball:~/midterm/q4_phoneBook — ssh — 204×57
[rserdah1@gsuad.gsu.edu@snowball q4_phoneBook]$ cat phoneBook.txt
A A             123 AAA Rd      1234567890
Z Z             321 ZZZ Rd      0987654321
[rserdah1@gsuad.gsu.edu@snowball q4_phoneBook]$ sh phoneBookUtil.sh
Phone Book Util
COMMANDS: ADD, READ, DEL, QUIT
ADD
Enter full name
C C
Enter address
222 CCC Rd
Enter phone number
0000000000
QUIT
[rserdah1@gsuad.gsu.edu@snowball q4_phoneBook]$ cat phoneBook.txt
A A             123 AAA Rd      1234567890
C C             222 CCC Rd              0000000000
Z Z             321 ZZZ Rd      0987654321
[rserdah1@gsuad.gsu.edu@snowball q4_phoneBook]$ sh phoneBookUtil.sh
Phone Book Util
COMMANDS: ADD, READ, DEL, QUIT
DEL
Enter full name
A A
ADD
Enter full name
G G
Enter address
444 GGG Rd
Enter phone number
1111111111
QUIT
[rserdah1@gsuad.gsu.edu@snowball q4_phoneBook]$ cat phoneBook.txt
C C             222 CCC Rd              0000000000
G G             444 GGG Rd              1111111111
Z Z             321 ZZZ Rd      0987654321
[rserdah1@gsuad.gsu.edu@snowball q4_phoneBook]$
```

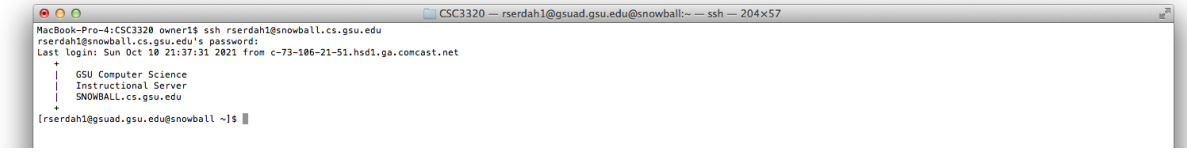
5. (4 pts each) Give brief answers with examples, wherever relevant

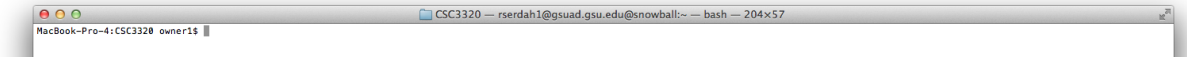
A. What is the use of a shell?

To provide a human readable interface between the user and 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.

There are not many visual differences besides the login confirmation and the title message. The directories are different as they change from the local directory of the computer to the remote directory of the server.





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

The compiler translates the human readable C program into the not very human readable machine language. The CPU can then interpret the translated machine 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 command printf() works in C, but echo and printf work in BASH. However, it is possible to call echo from a C script *through* the shell.

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 snowball.cs.gsu.edu` : This command is to connect to the GSU Snowball server through Secure Shell protocol. This allows students to log into the server.

`scp rserdah1@snowball.cs.gsu.edu:fromRemote.txt .` : This command was called from my local machine to download a test file from the Snowball server. The command secure copy can be used to copy files from a remote server to a local machine, a local machine to a server, or from one server to another.

`wget` http://ftp.gnu.org/gnu/wget/wget-1.20.tar.gz : This command was used to download the wget source as a test. The wget command allows us to download files at a given URL.