

## **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
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:** Venkata Mani Mohana Rishitha Srikakulapu

**Campus ID:** vsrikakulapu1

**Panther #:** 002523638

### 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*

To copy the man text of 10 commands, for loop is used and the related screenshot for execution is below :

```
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ for((i=1;i<=10;i++));do echo Enter a c
ommand;read cmd; echo $cmd\ (STARTCOMMAND >> mandatabase.txt; man $cmd >> mandata
base.txt; echo $cmd\ (ENDCOMMAND >> mandatabase.txt; done
Enter a command
ls
Enter a command
cp
Enter a command
mv
Enter a command
rm
Enter a command
gzip
Enter a command
sed
Enter a command
awk
Enter a command
cat
Enter a command
head
Enter a command
more
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$
```

Then a shell script **helpme.sh** asks the user to type in a command and print the manual's text associated with that corresponding command from the mandatabase.txt file which is created above.

```
#!/bin/bash

echo Please enter a command:
read cmd

line1=`grep -n $cmd\ (STARTCOMMAND mandatabase.txt|awk -F: '{print $1}'`
l2=`grep -n $cmd\ (ENDCOMMAND mandatabase.txt|awk -F: '{print $1}'`

l1=`expr $line1 + 1`

if [ $l1 -ne 0 ]
then
    echo `tail -n +$l1 mandatabase.txt| head -n $((l2-l1))`
else
    echo sorry,I cannot help you
fi

~
~
~
~
~
~
~
```

16,2 All

**helpme.sh** should be run/executed by using **./helpme.sh** command. The execution starts and the user is asked to type in the command; after typing the command the manual text related to that command is printed as output.

```
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ ./helpme.sh
Please enter a command:
cat
CAT(1) User Commands CAT(1) NAME cat - concatenate files and print on the stand
rd output SYNOPSIS cat [OPTION]... [FILE]... DESCRIPTION Concatenate FILE(s), or
standard input, to standard output. -A, --show-all equivalent to -vET -b, --num
ber-nonblank number nonempty output lines, overrides -n -e equivalent to -vE -E,
--show-ends display $ at end of each line -n, --number number all output lines
-s, --squeeze-blank suppress repeated empty output lines -t equivalent to -vT -T
, --show-tabs display TAB characters as ^I -u (ignored) -v, --show-nonprinting u
se ^ and M- notation, except for LFD and TAB --help display this help and exit -
-version output version information and exit With no FILE, or when FILE is -, re
ad standard input. EXAMPLES cat f - g Output f's contents, then standard input,
then g's contents. cat Copy standard input to standard output. GNU coreutils onl
ine help: <http://www.gnu.org/software/coreutils/> Report cat translation bugs t
o <http://translationproject.org/team/> AUTHOR Written by Torbjorn Granlund and
Richard M. Stallman. COPYRIGHT Copyright © 2013 Free Software Foundation, Inc. L
icense GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>. Th
is is free software: you are free to change and redistribute it. There is NO WAR
RANTY, to the extent permitted by law. SEE ALSO tac(1) The full documentation fo
r cat is maintained as a Texinfo manual. If the info and cat programs are proper
ly installed at your site, the com- mand info coreutils 'cat invocation' should
give you access to the complete manual. GNU coreutils 8.22 November 2020 CAT(1)
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$
```

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)

Manually copy pasted the text from Wikipedia page into myexamfile.txt and then used Filezilla to copy the myexamfile.txt from computer to remote snowball server. The created a directory called **midterm** inside the pwd directory using the command

```
$ mkdir midterm
```

Then moved to midterm directory using **cd**

```
$ cd midterm
```

Then copied myexamfile.txt to midterm directory using

```
$ cp myexamfile.txt midterm
```

After the creation of midterm directory and copying myexamfile.txt to midterm directory. Command **cd** is used to change to midterm directory and then by **ls** command list of contents of midterm directory are shown as myexamfile.txt and wordcount.sh. wordcount.sh is then opened by using the command

```
$ vi wordCount.sh
```

```
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ cd midterm  
[vsrikakulapu1@gsuad.gsu.edu@snowball midterm]$ ls  
myexamfile.txt wordCount.sh  
[vsrikakulapu1@gsuad.gsu.edu@snowball midterm]$ vi wordCount.sh
```

Now, the shell script named wordCount.sh is created to find the number of occurrences of a particular keyword typed by the user.



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.

**notAccessed.sh** is the shell script that performs the task of finding files in my home directory ( /home/vsrrikapulap1) that have not been accessed for N days, where N is typed in by the user.

```
#!/bin/bash
echo Enter the value of N
read value
N=`expr $value - 1`
find /home/vsrikakulapu1 -type f -atime +$N -name "*" | gzip > Compressed.gz
```

"notAccessed.sh" 6L, 144C 6,0-1

The shell script **notAccessed.sh** is created and viewed using **\$ vi notAccessed.sh** and is executed using **\$ ./notAccessed.sh** and **N** is entered when prompted. A new compressed file **Compressed.gz** is created by the file list output for files not accessed for N days.

```
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ vi notAccessed.sh
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ ./notAccessed.sh
Enter the value of N
8
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ ls -l Compressed.gz
-rw-rw-r--. 1 vsrikakulapu1@gsuad.gsu.edu vsrikakulapu1@gsuad.gsu.edu 474 Mar  2
04:44 Compressed.gz
```

We can decompress the compressed file using the command

```
$ gunzip Compressed.gz
```

It always asks if we want to override the compressed file or not. After decompressing it, we can see the file list in the file Compressed i.e., without the .gz extension.

The contents of file Compressed shows the file list that is actually compressed into a single **.gz** file

```
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ ./notAccessed.sh
Enter the value of N
30
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ gunzip Compressed.gz
gzip: Compressed already exists; do you wish to overwrite (y or n)? y
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ cat -n Compressed
  1 /home/vsrikakulapu1/csc3320/lab2/myLab2.txt
  2 /home/vsrikakulapu1/csc3320/myLab2.txt
  3 /home/vsrikakulapu1/Lab3/test.txt
  4 /home/vsrikakulapu1/homeworks/homework_instructions.txt
  5 /home/vsrikakulapu1/csc2720/lab1/myLab1.txt
  6 /home/vsrikakulapu1/public/myRealEstate.csv
  7 /home/vsrikakulapu1/public/Public/Submission/Lab2/Lab2_P2/RealEstate.csv
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$
```

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.

A csv file named **pHdatabase.csv** is created and few names, addresses, phone numbers are added to create a phonebook. This file is created and accessed using

**\$ vi pHdatabase.csv**

```
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ vi pHdatabase.csv
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$
```

When opened using vi, the file is shown as below :

```
Name      Address Phone number
Albert einstein '681 Morning Glory CirJesup, Georgia(GA), 31546' (561) 964-9868
Isaac  '681 Main St Jeffersonville, Georgia(GA), 31044' (614) 523-0426
Marie curie      '6817 Crest HWY Molena, Georgia(GA), 30258' (615) 220-4260
Georgialilée    '682 Stonemill Mnr Lithonia, Georgia(GA), 30058' (617) 926-3534
Anand  '682 Sir Charles Dr Fairburn, Georgia(GA), 30213' (478) 240-9372
Ram    '8828 Georgia 112 HWY Rochelle, Georgia(GA), 31079' (912) 579-9977
Bob    '137 Easy St Americus, Georgia(GA), 31709' (601) 795-8630
Collin '10892 Crabapple Rd # Roswell, Georgia(GA), 30075' (229) 365-7584
Pardhu '8821 Georgia 15 HWY N Warthen, Georgia(GA), 31094' (931) 707-8086
Jimmy  '8822 Waycross HWY Screven, Georgia(GA), 31560' (931) 707-8027
~
~
```

The **pHdatabase.csv** file is the phonebook database whose permissions are to be set as only the user can access the file and nobody else can access it so the screenshot is below:

```
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ ls -l pHdatabase.csv
-rw-rw-r--. 1 vsrikakulapu1@gsuad.gsu.edu vsrikakulapu1@gsuad.gsu.edu 874 Mar 2
05:13 pHdatabase.csv
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ chmod 700 pHdatabase.csv
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ ls -l pHdatabase.csv
-rwx-----. 1 vsrikakulapu1@gsuad.gsu.edu vsrikakulapu1@gsuad.gsu.edu 874 Mar 2
05:13 pHdatabase.csv
```

This csv file is then modified with a shell script **phoneDataBase.sh** which can be created and accessed using

**\$ vi phoneDataBase.sh**

```
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ vi phoneDataBase.sh
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$
```



The shell script **phoneDataBase.sh** helps to read contents, add new contents, edit existing ones and delete the current contents of **pHdatabase.csv** file. The script is as follows:

```
#!/bin/bash

echo "Please enter the task number you wish to do ?"
echo "1 - Read all contents"
echo "2 - Add new contents"
echo "3 - Edit existing phone contents"
echo "4 - Delete present contents"

read task

case $task in
1) echo "The phone book utility contents are : "
  awk '{print $0}' pHdatabase.csv | (sed -u 1q ;sort +0 -1);;

2) echo "Enter full Name of contact"
  read name
  echo "Enter the address"
  read adr
  echo "Enter phone number"
  read phNum
  echo $name$'\t'$adr$'\t'$phNum >> pHdatabase.csv
  echo Yay! A new contact has been added
  echo After adding, the phone book contents are :
  awk '{print $0}' pHdatabase.csv | (sed -u 1q ;sort +0 -1);;

3) echo "Enter content you want to edit : "
  read edit
  echo "Enter content you want be substituted as : "
  read subs
  sed -i s/$edit/$subs/g pHdatabase.csv
  echo After editing, the phone book contents are :
  awk '{print $0}' pHdatabase.csv | (sed -u 1q ;sort +0 -1);;

4) echo "Enter content you want to delete : "
  read delete
  #sed -i /$delete/d pHdatabase.csv
  sed -i s/$delete//g pHdatabase.csv
  echo After deleting, the phone book contents are :
  #awk '{print $0}' pHdatabase.csv | sort +0 -1;;
  awk '{print $0}' pHdatabase.csv | (sed -u 1q ;sort +0 -1);;
&#92;sac
~
~
```

The first option allows to read all the contents in sorted format. The output after executing the shell script **phoneDataBase.sh** using command

```
$ ./phoneDataBase.sh
```

allows to read and modify contents of **pHdatabase.csv** file based on the task chosen.

```
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ ./phoneDataBase.sh
Please enter the task number you wish to do ?
1 - Read all contents
2 - Add new contents
3 - Edit existing phone contents
4 - Delete present contents
1
The phone book utility contents are :
Name      Address Phone number
Albert einstein 681 Morning Glory CirJesup, Georgia(GA), 31546 (561) 964-9868
Anand      682 Sir Charles Dr Fairburn, Georgia(GA), 30213 (478) 240-9372
Bob        137 Easy St Americus, Georgia(GA), 31709 (601) 795-8630
Collin     10892 Crabapple Rd # Roswell, Georgia(GA), 30075 (229) 365-7584
Georgialilée 682 Stonemill Mnr Lithonia, Georgia(GA), 30058 (617) 926-3534
Isaac      681 Main St Jeffersonville, Georgia(GA), 31044 (614) 523-0426
Jimmy      8822 Waycross HWY Screven, Georgia(GA), 31560 (931) 707-8027
Marie curie 6817 Crest HWY Molena, Georgia(GA), 30258 (615) 220-4260
Mickey Oberoi 36 Driveway Highland coast Ohio (OH), 30186 (368) 583-8940
Pardhu     8821 Georgia 15 HWY N Warthen, Georgia(GA), 31094 (931) 707-8086
Ram        8828 Georgia 112 HWY Rochelle, Georgia(GA), 31079 (912) 579-9977
```

The second option allows user to add new contents to the phonebook and the contents of sorted, modified file is outputted as below :

```
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ ./phoneDataBase.sh
Please enter the task number you wish to do ?
1 - Read all contents
2 - Add new contents
3 - Edit existing phone contents
4 - Delete present contents
2
Enter full Name of contact
Dan Kenny
Enter the address
782 Chatanooga Bay Greenside road, Georgia(GA), 30284
Enter phone number
(456) 372-9483
Yay! A new contact has been added
After adding, the phone book contents are :
Name      Address Phone number
Albert einstein 681 Morning Glory CirJesup, Georgia(GA), 31546 (561) 964-9868
Anand      682 Sir Charles Dr Fairburn, Georgia(GA), 30213 (478) 240-9372
Bob        137 Easy St Americus, Georgia(GA), 31709 (601) 795-8630
Collin     10892 Crabapple Rd # Roswell, Georgia(GA), 30075 (229) 365-7584
Dan Kenny   782 Chatanooga Bay Greenside road, Georgia(GA), 30284 (456) 372-9483
Georgialilée 682 Stonemill Mnr Lithonia, Georgia(GA), 30058 (617) 926-3534
Isaac      681 Main St Jeffersonville, Georgia(GA), 31044 (614) 523-0426
Jimmy      8822 Waycross HWY Screven, Georgia(GA), 31560 (931) 707-8027
Marie curie 6817 Crest HWY Molena, Georgia(GA), 30258 (615) 220-4260
Mickey Oberoi 36 Driveway Highland coast Ohio (OH), 30186 (368) 583-8940
Pardhu     8821 Georgia 15 HWY N Warthen, Georgia(GA), 31094 (931) 707-8086
Ram        8828 Georgia 112 HWY Rochelle, Georgia(GA), 31079 (912) 579-9977
```

The third option allows user to edit existing contents of the phonebook and the contents of sorted, modified file is outputted as below :

```
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ ./phoneDataBase.sh
Please enter the task number you wish to do ?
1 - Read all contents
2 - Add new contents
3 - Edit existing phone contents
4 - Delete present contents
3
Enter content you want to edit :
Hwy
Enter content you want be substituted as :
Hwy
After editing, the phone book contents are :
Name      Address Phone number
Albert einstein 681 Morning Glory CirJesup, Georgia(GA), 31546 (561) 964-9868
Anand      682 Sir Charles Dr Fairburn, Georgia(GA), 30213 (478) 240-9372
Bob        137 Easy St Americus, Georgia(GA), 31709 (601) 795-8630
Collin     10892 Crabapple Rd # Roswell, Georgia(GA), 30075 (229) 365-7584
Dan Kenny  782 Chatanooga Bay Greenside road, Georgia(GA), 30284 (456) 372-9483
Georgialilée 682 Stonemill Mnr Lithonia, Georgia(GA), 30058 (617) 926-3534
Isaac      681 Main St Jeffersonville, Georgia(GA), 31044 (614) 523-0426
Jimmy      8822 Waycross Hwy Screven, Georgia(GA), 31560 (931) 707-8027
Marie curie 6817 Crest Hwy Molena, Georgia(GA), 30258 (615) 220-4260
Mickey Oberoi 36 Driveway Highland coast Ohio (OH), 30186 (368) 583-8940
Pardhu     8821 Georgia 15 Hwy N Warthen, Georgia(GA), 31094 (931) 707-8086
Ram        8828 Georgia 112 Hwy Rochelle, Georgia(GA), 31079 (912) 579-9977
```

The fourth option allows user to delete existing contents of the phonebook and the contents of sorted, modified file is outputted as below :

```
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ ./phoneDataBase.sh
Please enter the task number you wish to do ?
1 - Read all contents
2 - Add new contents
3 - Edit existing phone contents
4 - Delete present contents
4
Enter content you want to delete :
road
After deleting, the phone book contents are :
Name      Address Phone number
Albert einstein 681 Morning Glory CirJesup, Georgia(GA), 31546 (561) 964-9868
Anand      682 Sir Charles Dr Fairburn, Georgia(GA), 30213 (478) 240-9372
Bob        137 Easy St Americus, Georgia(GA), 31709 (601) 795-8630
Collin     10892 Crabapple Rd # Roswell, Georgia(GA), 30075 (229) 365-7584
Dan Kenny  782 Chatanooga Bay Greenside , Georgia(GA), 30284 (456) 372-9483
Georgialilée 682 Stonemill Mnr Lithonia, Georgia(GA), 30058 (617) 926-3534
Isaac      681 Main St Jeffersonville, Georgia(GA), 31044 (614) 523-0426
Jimmy      8822 Waycross Hwy Screven, Georgia(GA), 31560 (931) 707-8027
Marie curie 6817 Crest Hwy Molena, Georgia(GA), 30258 (615) 220-4260
Mickey Oberoi 36 Driveway Highland coast Ohio (OH), 30186 (368) 583-8940
Pardhu     8821 Georgia 15 Hwy N Warthen, Georgia(GA), 31094 (931) 707-8086
Ram        8828 Georgia 112 Hwy Rochelle, Georgia(GA), 31079 (912) 579-9977
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$
```

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

The C script that will calculate the factorial of a number typed by the user is **factorial.c** which is coded as :

```
#include <stdio.h>

int main(void)
{
    int n,f=1,i=1;
    printf("Enter a number : \n");
    scanf("%d",&n);
    while(i<=n){
        f = f*i;
        i++;
    }

    printf("factorial is %d\n",f);
    return 0;
}
```

The C program **factorial.c** is compiled using command

**\$ cc factorial.c**

And executed using command

**\$ ./a.out**

And the user types in the number whose factorial is to be calculated.

```
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ vi factorial.c
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ cc factorial.c
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ ./a.out
Enter a number :
6
factorial is 720
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$
```

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)

The C script that will find the new integer value of an original integer, which is manually entered in code, when it is bit-shifted left by 3 bits and added to its complement is **bitShift.c** which is coded as :

```
#include <stdio.h>

int main(void)
{
    unsigned int num = 25;
    printf("1s Compliment of %d is = %d\n", num, ~num);

    printf("Integer bitshifted left by 3 bits is = %d\n", num<<3);

    unsigned int newNum = (num << 3) + (~num);

    printf("Sum of 1s complement + 3 left bitshifts of %d is New number which is %d\n", num, newNum);
    return 0;
}
```

The C program **bitShift.c** is compiled using command

**\$ cc bitShift.c**

And executed using command

**\$ ./a.out**

The output is as follows:

```
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ vi bitShift.c
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ cc bitShift.c
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ ./a.out
1s Compliment of 25 is = -26
Integer bitshifted left by 3 bits is = 200
Sum of 1s complement + 3 left bitshifts of 25 is New number which is 174
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$
```

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

The C script to convert the integer to binary is **intToBinary.c** which is written, compiled and executed as :

```
#include<stdio.h>

int main(){
int iNum,x;
int bNum[20];
printf("Enter the integer you want to convert to binary: ");
scanf("%d",&iNum);
int Number = iNum;
for(x=0;iNum>0;x++){
bNum[x] = iNum%2;
iNum = iNum/2;
}
printf("Binary of %d is : ",Number);
for(x=x-1;x>=0;x--){
printf("%d",bNum[x]);
}
printf("\n");
return 0;
}
```

**intToBinary.c** is written, compiled and executed as :

```
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ vi intToBinary.c
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ cc intToBinary.c
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ ./a.out
Enter the integer you want to convert to binary: 27
Binary of 27 is : 11011
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$
```

The C script to convert the binary to integer is **binToInt.c** which is written, compiled and executed as :

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>

int binToInt(long bNum)
{
    int iNum=0,mod,x=0;
    while(bNum!=0)
    {
        mod = bNum % 10;
        bNum = bNum / 10;
        iNum = iNum + mod*pow(2,x);
        x++;
    }
}

int main()
{
    long b;
    printf("Enter the binary number you want to convert : ");
    scanf("%d",&b);
    printf("Integer of binary number is %d\n",binToInt(b));
    return 0;
}
```

**binToInt.c** is written, compiled and executed as :

```
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ vi binToInt.c
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ cc binToInt.c -lm
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ ./a.out
Enter the binary number you want to convert : 10110
Integer of binary number is 22
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$
```

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

The shell script that will execute the C script of a factorial of a number typed by the user and execute the C script of finding new number after bit shift and adding 1s complement is :

[illegible]

The execution of this shell script is shown below :

```
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ vi executeC.sh
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$ ./executeC.sh
Factorial execution is :
Enter a number :
4
factorial is 24

Bit Shift execution is :
1s Compliment of 25 is = -26
Integer bitshifted left by 3 bits is = 200
Sum of 1s complement + 3 left bitshifts of 25 is New number which is 174
[vsrikakulapu1@gsuad.gsu.edu@snowball ~]$
```