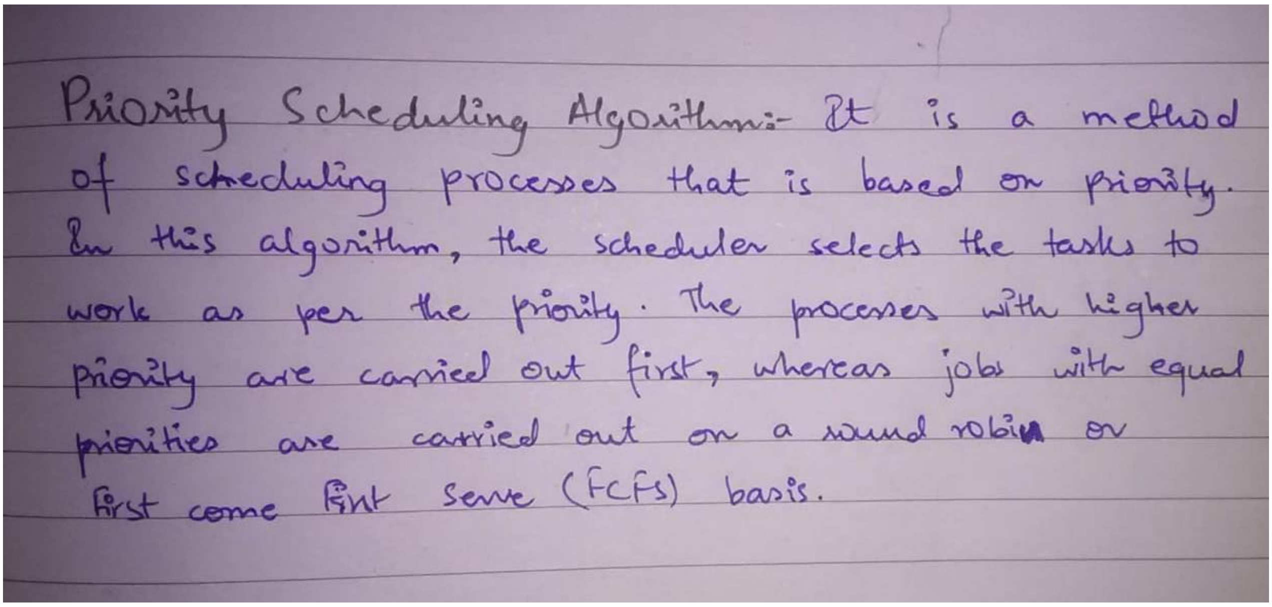
OPERATING SYSTEMS ASSIGNMENT – 01

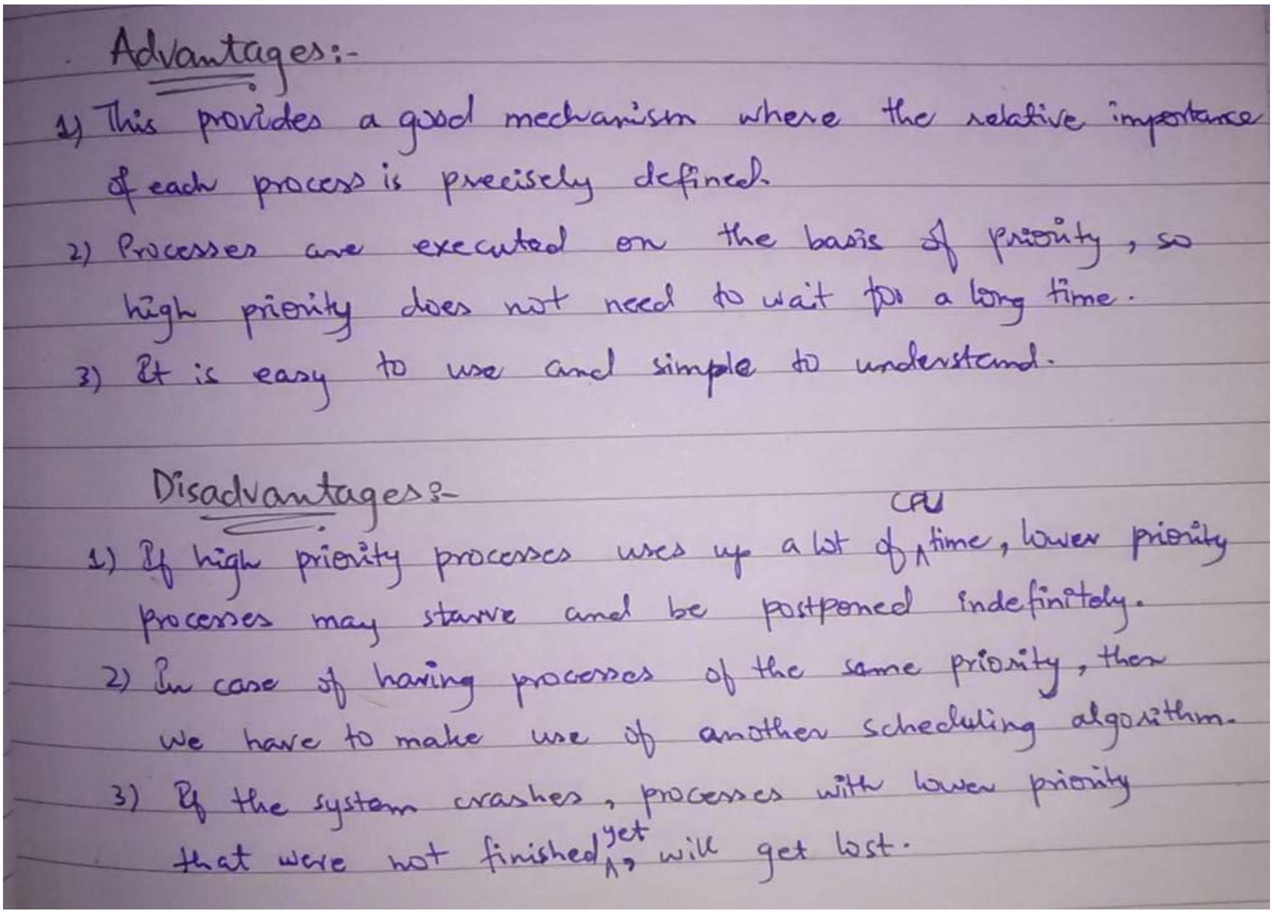
Name: Kamisha Salim
S.ID: 11070

QUESTION – 1

Part – a: Define priority scheduling algorithm and also its advantages and disadvantages.



Priority Scheduling Algorithm:- It is a method of scheduling processes that is based on priority. In this algorithm, the scheduler selects the tasks to work as per the priority. The processes with higher priority are carried out first, whereas jobs with equal priorities are carried out on a round robin or first come first serve (FCFS) basis.



Advantages:-

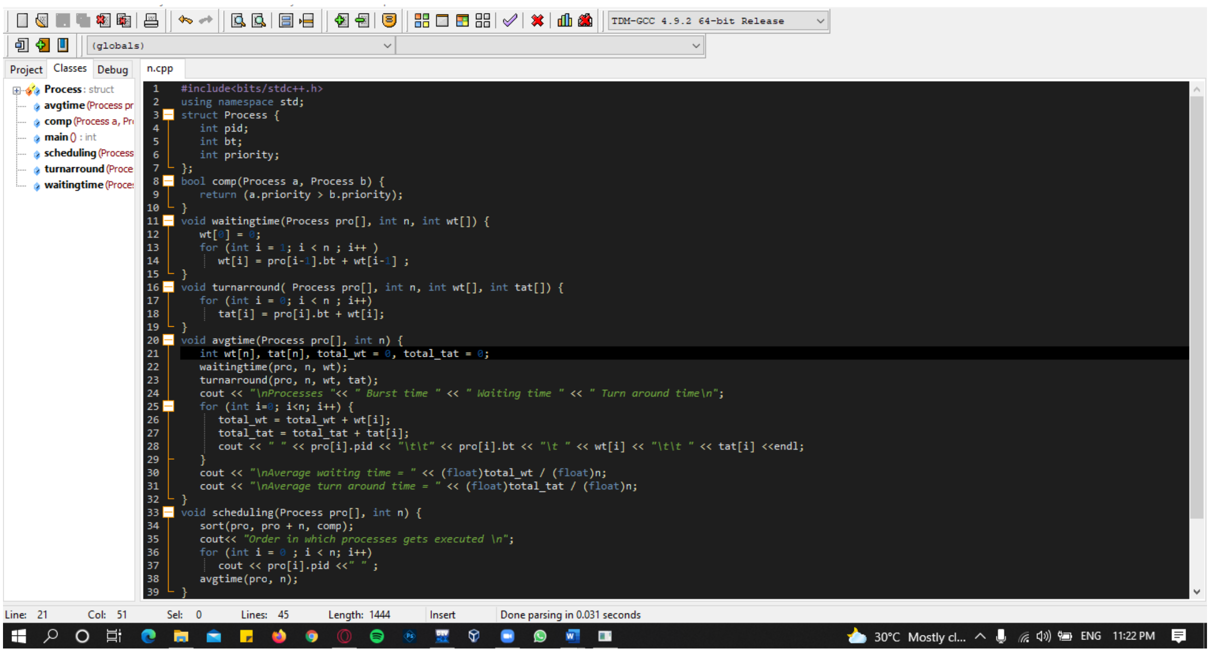
1) This provides a good mechanism where the relative importance of each process is precisely defined.
2) Processes are executed on the basis of priority, so high priority does not need to wait for a long time.
3) It is easy to use and simple to understand.

Disadvantages:-

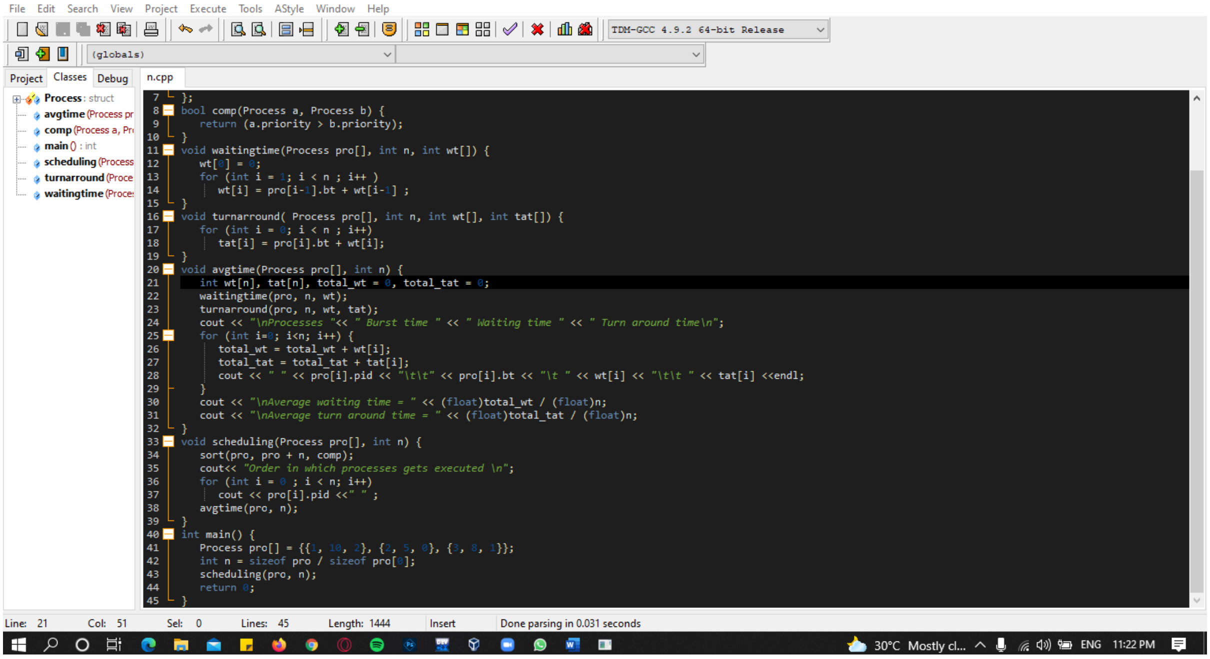
1) If high priority processes uses up a lot of CPU time, lower priority processes may starve and be postponed indefinitely.
2) In case of having processes of the same priority, then we have to make use of another scheduling algorithm.
3) If the system crashes, processes with lower priority that were not finished yet, will get lost.

Part – b: Write a program of the given algorithm. Apply priority scheduling algorithm using C/C++.

CODE:



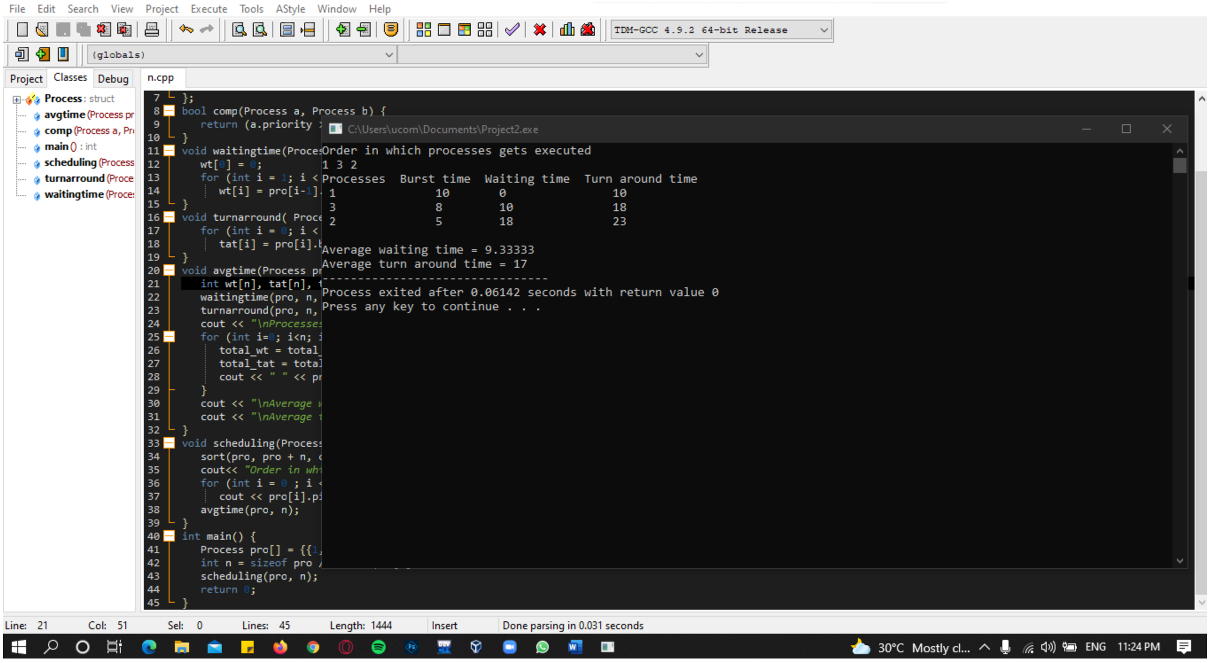
```cpp
#include<bits/stdc++.h>
using namespace std;
struct Process {
   int pid;
   int bt;
   int priority;
};
bool comp(Process a, Process b) {
   return (a.priority > b.priority);
}
void waitingtime(Process pro[], int n, int wt[]) {
   wt[0] = 0;
   for (int i = 1; i < n ; i++ )
      wt[i] = pro[i-1].bt + wt[i-1] ;
}
void turnarround( Process pro[], int n, int wt[], int tat[]) {
   for (int i = 0; i < n ; i++)
      tat[i] = pro[i].bt + wt[i];
}
void avgtime(Process pro[], int n) {
   int wt[n], tat[n], total_wt = 0, total_tat = 0;
   waitingtime(pro, n, wt);
   turnarround(pro, n, wt, tat);
   cout << "\nProcesses "<< " Burst time " << " Waiting time " << " Turn around time\n";
   for (int i=0; i<n; i++) {
      total_wt = total_wt + wt[i];
      total_tat = total_tat + tat[i];
      cout << " " << pro[i].pid << "\t\t" << pro[i].bt << "\t " << wt[i] << "\t\t " << tat[i] <<endl;
   }
   cout << "\nAverage waiting time = " << (float)total_wt / (float)n;
   cout << "\nAverage turn around time = " << (float)total_tat / (float)n;
}
void scheduling(Process pro[], int n) {
   sort(pro, pro + n, comp);
   cout<< "Order in which processes gets executed \n";
   for (int i = 0 ; i < n; i++)
      cout << pro[i].pid <<" " ;
   avgtime(pro, n);
}
```


File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
n.cpp
Process : struct
avgtime (Process pr
comp (Process a, Pro
main () : int
scheduling (Process
turnaround (Proce
waitingtime (Proces
Line: 21 Col: 51 Sel: 0 Lines: 45 Length: 1444 Insert Done parsing in 0.031 seconds
30°C Mostly cl... ENG 11:22 PM

```cpp
};
bool comp(Process a, Process b) {
   return (a.priority > b.priority);
}
void waitingtime(Process pro[], int n, int wt[]) {
   wt[0] = 0;
   for (int i = 1; i < n ; i++ )
      wt[i] = pro[i-1].bt + wt[i-1] ;
}
void turnarround( Process pro[], int n, int wt[], int tat[]) {
   for (int i = 0; i < n ; i++)
      tat[i] = pro[i].bt + wt[i];
}
void avgtime(Process pro[], int n) {
   int wt[n], tat[n], total_wt = 0, total_tat = 0;
   waitingtime(pro, n, wt);
   turnarround(pro, n, wt, tat);
   cout << "\nProcesses "<< " Burst time " << " Waiting time " << " Turn around time\n";
   for (int i=0; i<n; i++) {
      total_wt = total_wt + wt[i];
      total_tat = total_tat + tat[i];
      cout << " " << pro[i].pid << "\t\t" << pro[i].bt << "\t " << wt[i] << "\t\t " << tat[i] <<endl;
   }
   cout << "\nAverage waiting time = " << (float)total_wt / (float)n;
   cout << "\nAverage turn around time = " << (float)total_tat / (float)n;
}
void scheduling(Process pro[], int n) {
   sort(pro, pro + n, comp);
   cout<< "Order in which processes gets executed \n";
   for (int i = 0 ; i < n; i++)
      cout << pro[i].pid <<" " ;
   avgtime(pro, n);
}
```

```cpp
int main() {
   Process pro[] = {{1, 10, 2}, {2, 5, 0}, {3, 8, 1}};
   int n = sizeof pro / sizeof pro[0];
   scheduling(pro, n);
   return 0;
}
```

OUTPUT:



```
};
bool comp(Process a, Process b) {
    return (a.priority
}
void waitingtime(Proce
    wt[0] = 0;
    for (int i = 1; i <
        wt[i] = pro[i-1].
}
void turnarround( Proce
    for (int i = 0; i <
        tat[i] = pro[i].b
}
void avgtime(Process pr
    int wt[n], tat[n], t
    waitingtime(pro, n,
    turnarround(pro, n,
    cout << "\nProcesses
    for (int i=0; i<n; i
        total_wt = total_
        total_tat = total
        cout << " " << pr
    }
    cout << "\nAverage w
    cout << "\nAverage t
}
void scheduling(Process
    sort(pro, pro + n, c
    cout<< "Order in whi
    for (int i = 0 ; i <
        cout << pro[i].pi
    avgtime(pro, n);
}
int main() {
    Process pro[] = {{1,
    int n = sizeof pro /
    scheduling(pro, n);
    return 0;
}
```

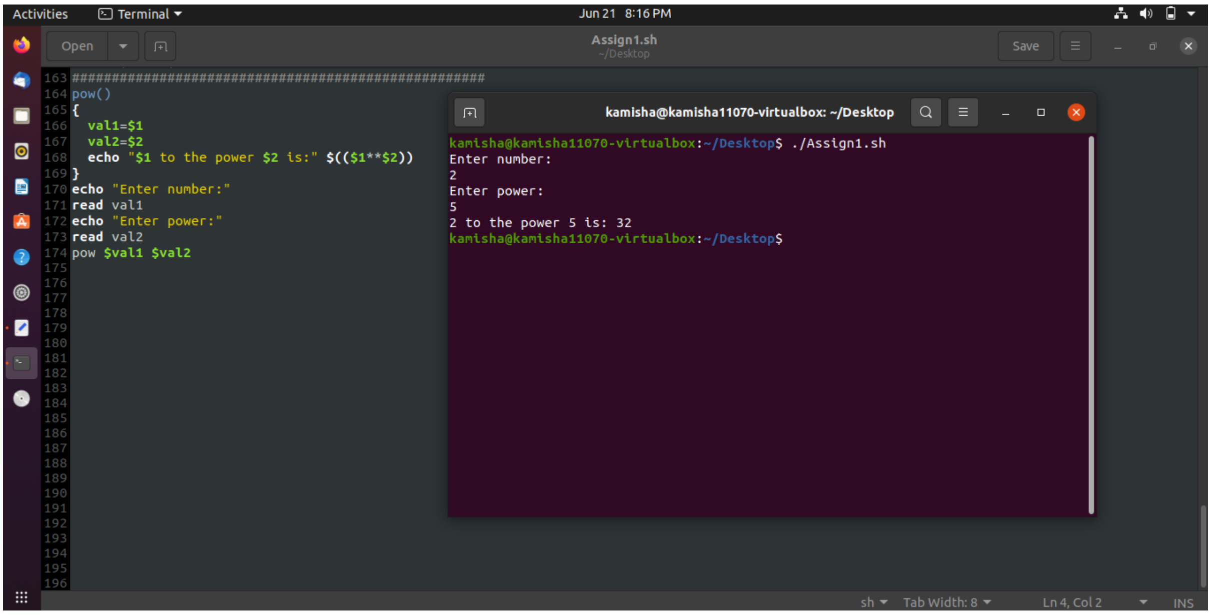
C:\Users\ucom\Documents\Project2.exe

```
Order in which processes gets executed
1 3 2
Processes  Burst time  Waiting time  Turn around time
 1              10        0                10
 3              8         10               18
 2              5         18               23

Average waiting time = 9.33333
Average turn around time = 17
--------------------------------
Process exited after 0.06142 seconds with return value 0
Press any key to continue . . .
```

QUESTION – 2

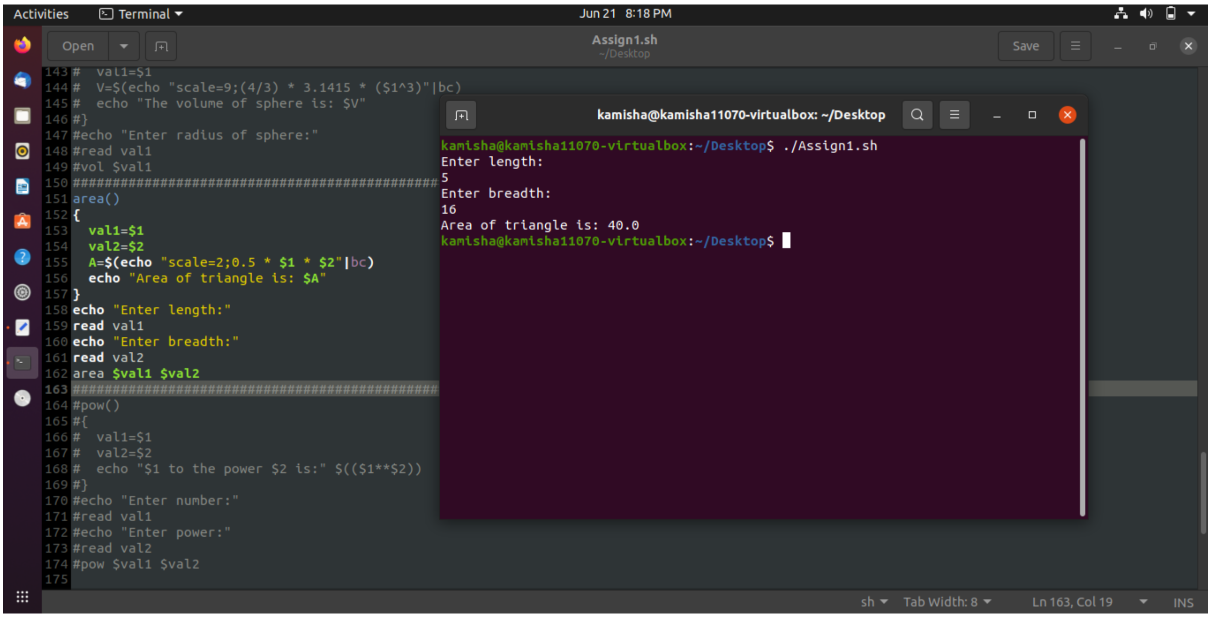
Part – a: Print a power b



```
####################################################
pow()
{
  val1=$1
  val2=$2
  echo "$1 to the power $2 is:" $(($1**$2))
}
echo "Enter number:"
read val1
echo "Enter power:"
read val2
pow $val1 $val2
```

```
kamisha@kamisha11070-virtualbox:~/Desktop$ ./Assign1.sh
Enter number:
2
Enter power:
5
2 to the power 5 is: 32
kamisha@kamisha11070-virtualbox:~/Desktop$
```

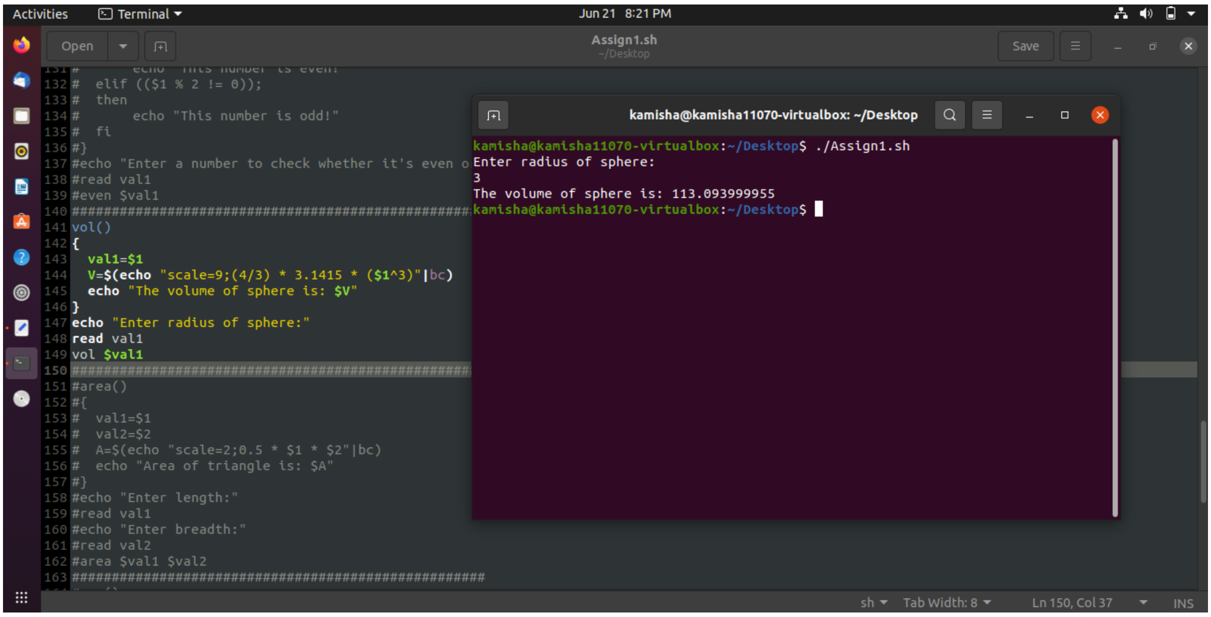
Part – b: Calculate area of triangle



```
#  val1=$1
#  V=$(echo "scale=9;(4/3) * 3.1415 * ($1^3)"|bc)
#  echo "The volume of sphere is: $V"
#}
#echo "Enter radius of sphere:"
#read val1
#vol $val1
##############################################
area()
{
  val1=$1
  val2=$2
  A=$(echo "scale=2;0.5 * $1 * $2"|bc)
  echo "Area of triangle is: $A"
}
echo "Enter length:"
read val1
echo "Enter breadth:"
read val2
area $val1 $val2
##############################################
#pow()
#{
#  val1=$1
#  val2=$2
#  echo "$1 to the power $2 is:" $(($1**$2))
#}
#echo "Enter number:"
#read val1
#echo "Enter power:"
#read val2
#pow $val1 $val2
```

```
kamisha@kamisha11070-virtualbox:~/Desktop$ ./Assign1.sh
Enter length:
5
Enter breadth:
16
Area of triangle is: 40.0
kamisha@kamisha11070-virtualbox:~/Desktop$
```

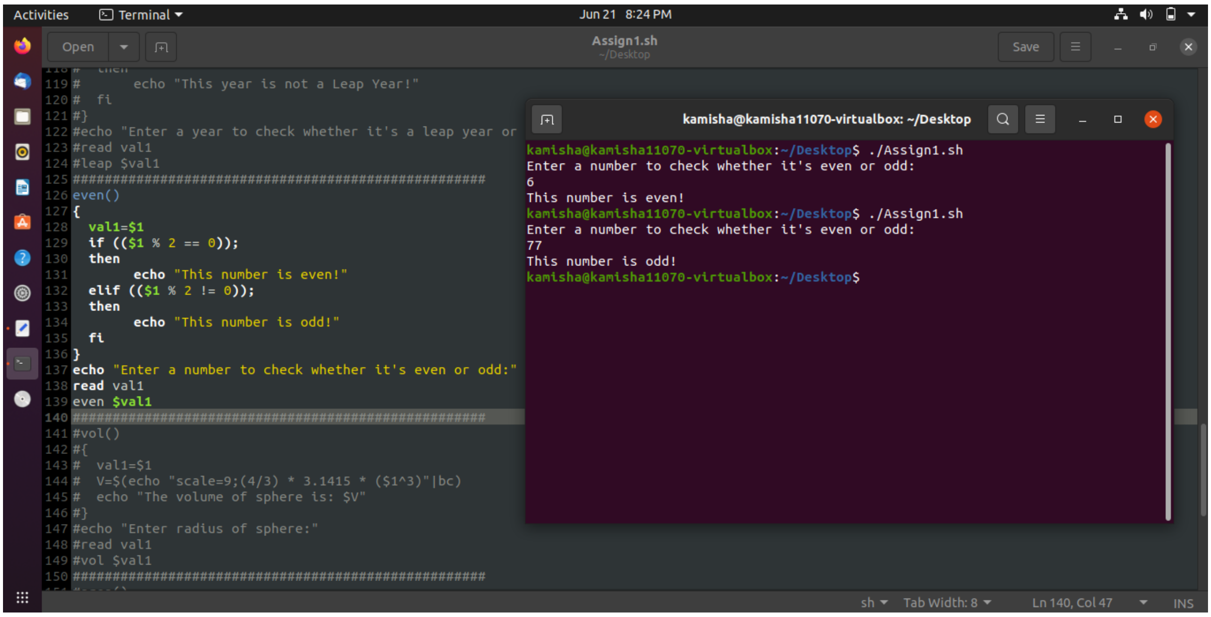
Part – c: Calculate volume of sphere



```
#       echo "This number is even!"
#  elif (($1 % 2 != 0));
#  then
#       echo "This number is odd!"
#  fi
#}
#echo "Enter a number to check whether it's even o
#read val1
#even $val1
####################################################
vol()
{
  val1=$1
  V=$(echo "scale=9;(4/3) * 3.1415 * ($1^3)"|bc)
  echo "The volume of sphere is: $V"
}
echo "Enter radius of sphere:"
read val1
vol $val1
####################################################
#area()
#{
#  val1=$1
#  val2=$2
#  A=$(echo "scale=2;0.5 * $1 * $2"|bc)
#  echo "Area of triangle is: $A"
#}
#echo "Enter length:"
#read val1
#echo "Enter breadth:"
#read val2
#area $val1 $val2
####################################################
```

```
kamisha@kamisha11070-virtualbox:~/Desktop$ ./Assign1.sh
Enter radius of sphere:
3
The volume of sphere is: 113.093999955
kamisha@kamisha11070-virtualbox:~/Desktop$ 
```

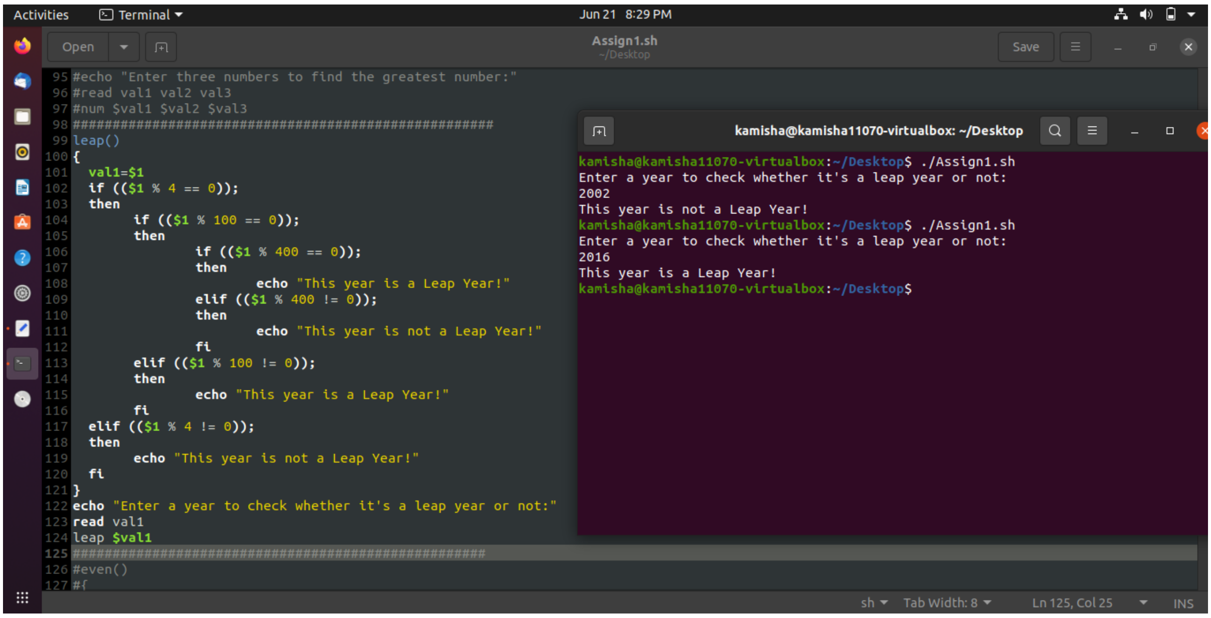
Part – d: Check if a number is odd or even



```
#       echo "This year is not a Leap Year!"
#  fi
#}
#echo "Enter a year to check whether it's a leap year or
#read val1
#leap $val1
####################################################
even()
{
  val1=$1
  if (($1 % 2 == 0));
  then
        echo "This number is even!"
  elif (($1 % 2 != 0));
  then
        echo "This number is odd!"
  fi
}
echo "Enter a number to check whether it's even or odd:"
read val1
even $val1
####################################################
#vol()
#{
#  val1=$1
#  V=$(echo "scale=9;(4/3) * 3.1415 * ($1^3)"|bc)
#  echo "The volume of sphere is: $V"
#}
#echo "Enter radius of sphere:"
#read val1
#vol $val1
####################################################
```

```
kamisha@kamisha11070-virtualbox:~/Desktop$ ./Assign1.sh
Enter a number to check whether it's even or odd:
6
This number is even!
kamisha@kamisha11070-virtualbox:~/Desktop$ ./Assign1.sh
Enter a number to check whether it's even or odd:
77
This number is odd!
kamisha@kamisha11070-virtualbox:~/Desktop$
```

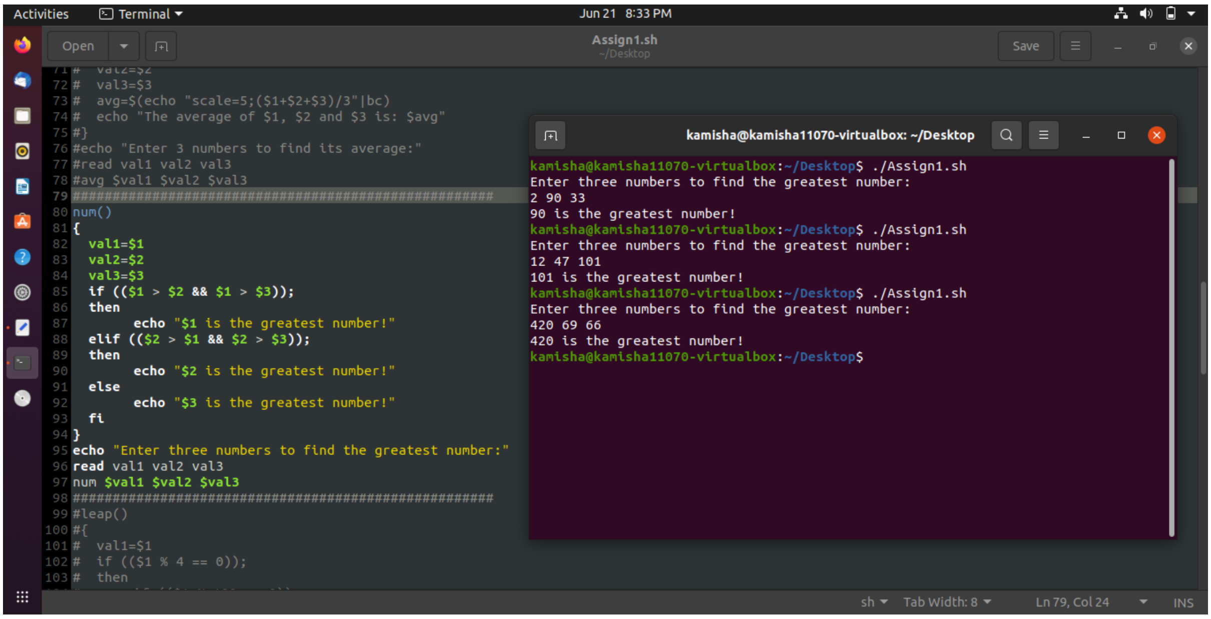
Part – e: Check if a year is leap year or not



```
#echo "Enter three numbers to find the greatest number:"
#read val1 val2 val3
#num $val1 $val2 $val3
####################################################
leap()
{
  val1=$1
  if (($1 % 4 == 0));
  then
        if (($1 % 100 == 0));
        then
                if (($1 % 400 == 0));
                then
                        echo "This year is a Leap Year!"
                elif (($1 % 400 != 0));
                then
                        echo "This year is not a Leap Year!"
                fi
        elif (($1 % 100 != 0));
        then
                echo "This year is a Leap Year!"
        fi
  elif (($1 % 4 != 0));
  then
        echo "This year is not a Leap Year!"
  fi
}
echo "Enter a year to check whether it's a leap year or not:"
read val1
leap $val1
###################################################
#even()
#{
```

```
kamisha@kamisha11070-virtualbox:~/Desktop$ ./Assign1.sh
Enter a year to check whether it's a leap year or not:
2002
This year is not a Leap Year!
kamisha@kamisha11070-virtualbox:~/Desktop$ ./Assign1.sh
Enter a year to check whether it's a leap year or not:
2016
This year is a Leap Year!
kamisha@kamisha11070-virtualbox:~/Desktop$
```

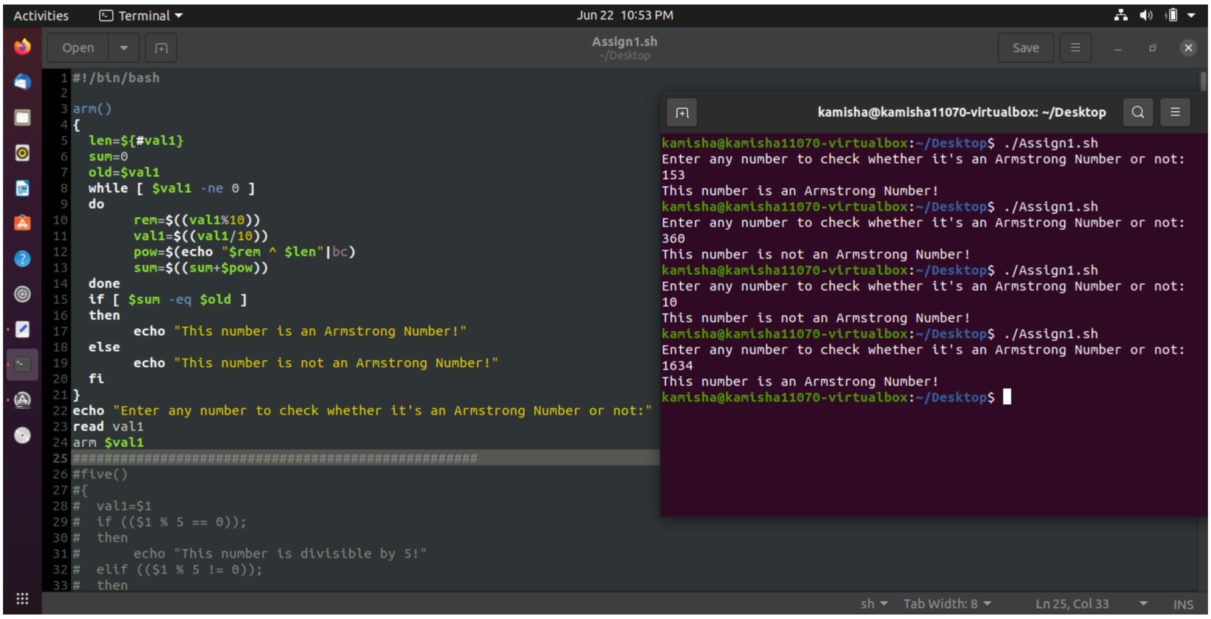
Part – f: Find the largest number among three numbers



```
#   val2=$2
#   val3=$3
#   avg=$(echo "scale=5;($1+$2+$3)/3"|bc)
#   echo "The average of $1, $2 and $3 is: $avg"
#}
#echo "Enter 3 numbers to find its average:"
#read val1 val2 val3
#avg $val1 $val2 $val3
####################################################
num()
{
  val1=$1
  val2=$2
  val3=$3
  if (($1 > $2 && $1 > $3));
  then
        echo "$1 is the greatest number!"
  elif (($2 > $1 && $2 > $3));
  then
        echo "$2 is the greatest number!"
  else
        echo "$3 is the greatest number!"
  fi
}
echo "Enter three numbers to find the greatest number:"
read val1 val2 val3
num $val1 $val2 $val3
####################################################
#leap()
#{
#   val1=$1
#   if (($1 % 4 == 0));
#   then
```

```
kamisha@kamisha11070-virtualbox:~/Desktop$ ./Assign1.sh
Enter three numbers to find the greatest number:
2 90 33
90 is the greatest number!
kamisha@kamisha11070-virtualbox:~/Desktop$ ./Assign1.sh
Enter three numbers to find the greatest number:
12 47 101
101 is the greatest number!
kamisha@kamisha11070-virtualbox:~/Desktop$ ./Assign1.sh
Enter three numbers to find the greatest number:
420 69 66
420 is the greatest number!
kamisha@kamisha11070-virtualbox:~/Desktop$
```

Part – g: Check if a number is an Armstrong number or not

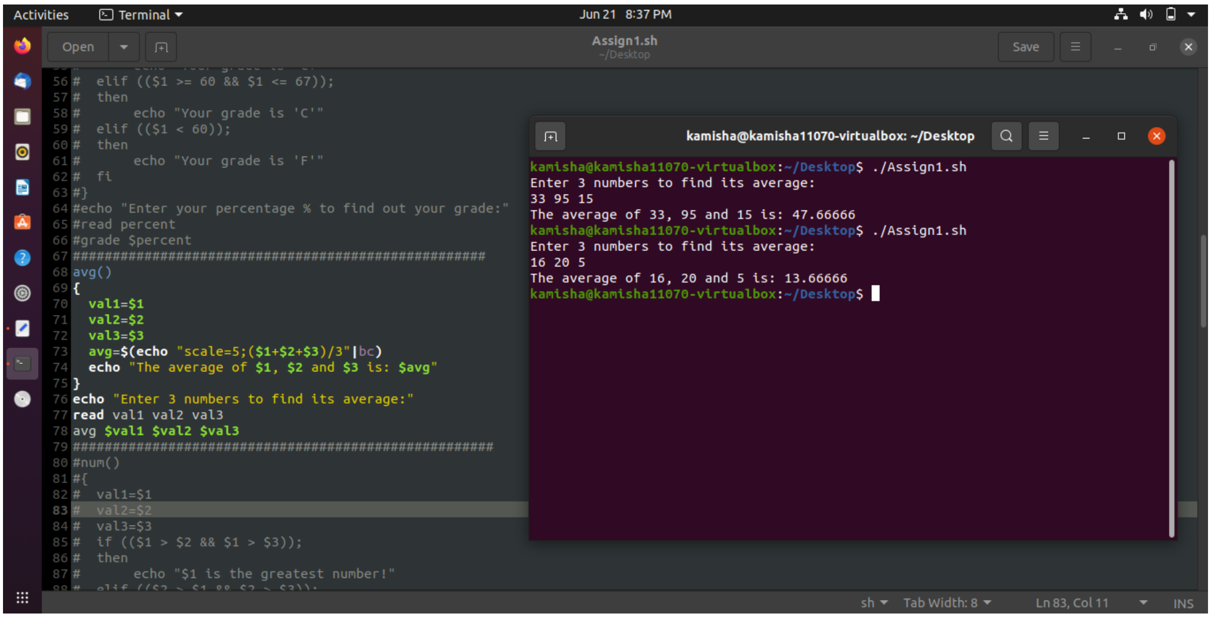


```
#!/bin/bash

arm()
{
  len=${#val1}
  sum=0
  old=$val1
  while [ $val1 -ne 0 ]
  do
        rem=$((val1%10))
        val1=$((val1/10))
        pow=$(echo "$rem ^ $len"|bc)
        sum=$((sum+$pow))
  done
  if [ $sum -eq $old ]
  then
        echo "This number is an Armstrong Number!"
  else
        echo "This number is not an Armstrong Number!"
  fi
}
echo "Enter any number to check whether it's an Armstrong Number or not:"
read val1
arm $val1
####################################################
#five()
#{
#   val1=$1
#   if (($1 % 5 == 0));
#   then
#        echo "This number is divisible by 5!"
#   elif (($1 % 5 != 0));
#   then
```

```
kamisha@kamisha11070-virtualbox:~/Desktop$ ./Assign1.sh
Enter any number to check whether it's an Armstrong Number or not:
153
This number is an Armstrong Number!
kamisha@kamisha11070-virtualbox:~/Desktop$ ./Assign1.sh
Enter any number to check whether it's an Armstrong Number or not:
360
This number is not an Armstrong Number!
kamisha@kamisha11070-virtualbox:~/Desktop$ ./Assign1.sh
Enter any number to check whether it's an Armstrong Number or not:
10
This number is not an Armstrong Number!
kamisha@kamisha11070-virtualbox:~/Desktop$ ./Assign1.sh
Enter any number to check whether it's an Armstrong Number or not:
1634
This number is an Armstrong Number!
kamisha@kamisha11070-virtualbox:~/Desktop$
```

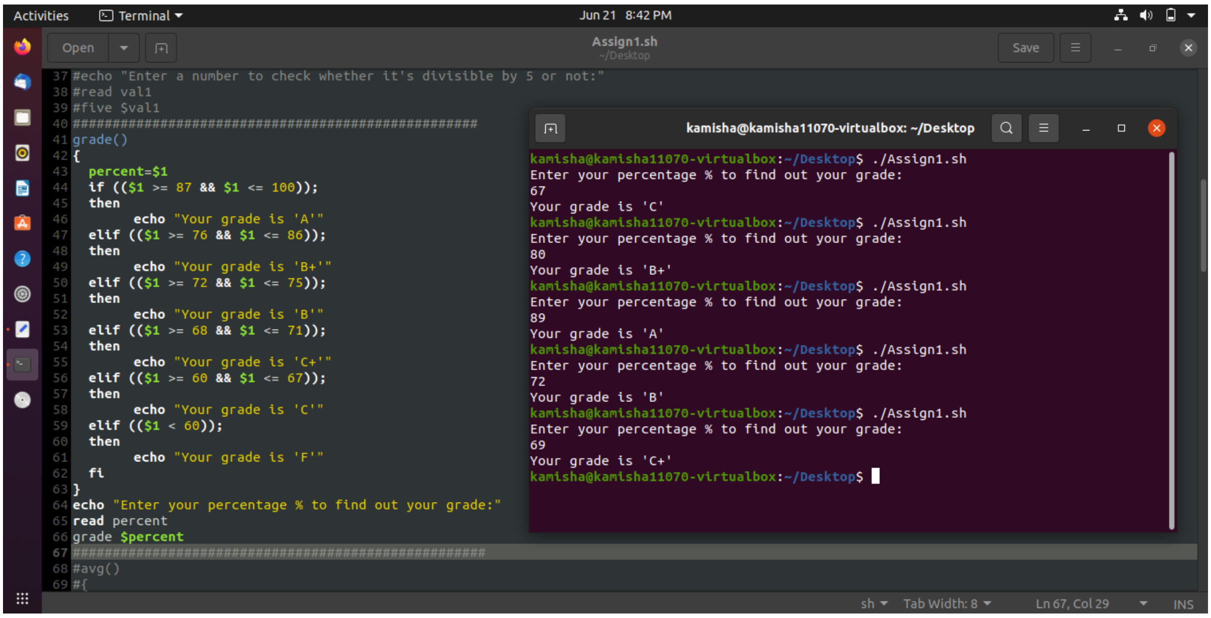
Part – h: Find the average of three values



```
#   elif (($1 >= 60 && $1 <= 67));
#   then
#        echo "Your grade is 'C'"
#   elif (($1 < 60));
#   then
#        echo "Your grade is 'F'"
#   fi
#}
#echo "Enter your percentage % to find out your grade:"
#read percent
#grade $percent
#####################################################
avg()
{
  val1=$1
  val2=$2
  val3=$3
  avg=$(echo "scale=5;($1+$2+$3)/3"|bc)
  echo "The average of $1, $2 and $3 is: $avg"
}
echo "Enter 3 numbers to find its average:"
read val1 val2 val3
avg $val1 $val2 $val3
######################################################
#num()
#{
#   val1=$1
#   val2=$2
#   val3=$3
#   if (($1 > $2 && $1 > $3));
#   then
#        echo "$1 is the greatest number!"
#   elif (($2 > $1 && $2 > $3));
```

```
kamisha@kamisha11070-virtualbox:~/Desktop$ ./Assign1.sh
Enter 3 numbers to find its average:
33 95 15
The average of 33, 95 and 15 is: 47.66666
kamisha@kamisha11070-virtualbox:~/Desktop$ ./Assign1.sh
Enter 3 numbers to find its average:
16 20 5
The average of 16, 20 and 5 is: 13.66666
kamisha@kamisha11070-virtualbox:~/Desktop$
```

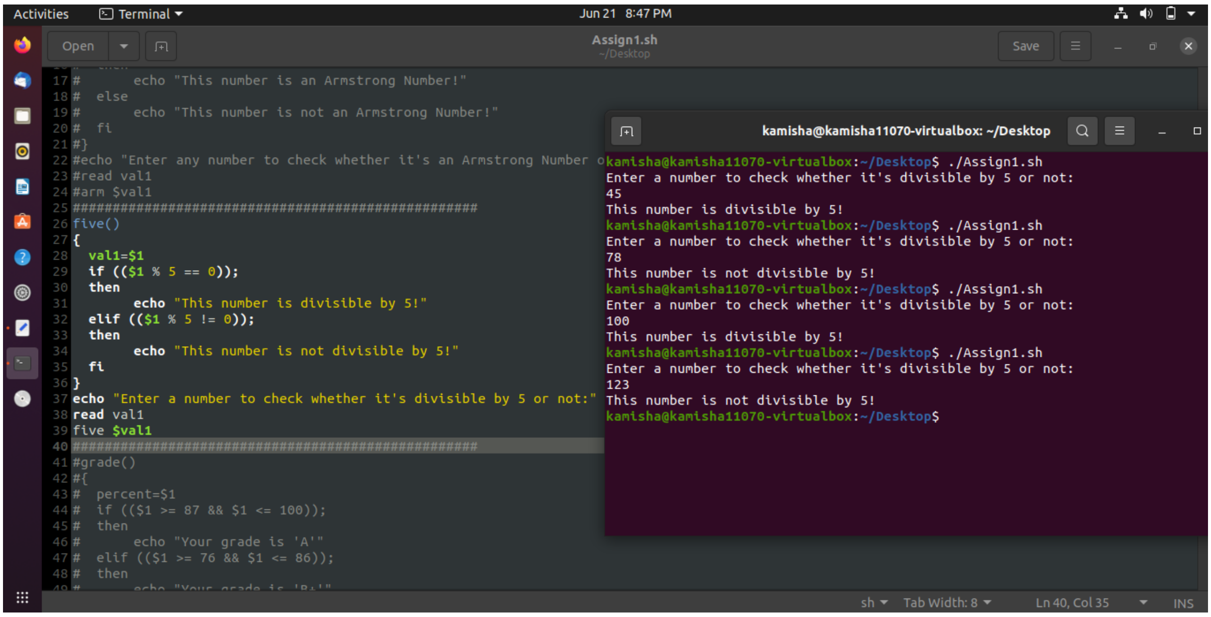
Part – i: Calculate grade ‘A’ to ‘F’ after inputting percentage



```
#echo "Enter a number to check whether it's divisible by 5 or not:"
#read val1
#five $val1
###################################################
grade()
{
  percent=$1
  if (($1 >= 87 && $1 <= 100));
  then
        echo "Your grade is 'A'"
  elif (($1 >= 76 && $1 <= 86));
  then
        echo "Your grade is 'B+'"
  elif (($1 >= 72 && $1 <= 75));
  then
        echo "Your grade is 'B'"
  elif (($1 >= 68 && $1 <= 71));
  then
        echo "Your grade is 'C+'"
  elif (($1 >= 60 && $1 <= 67));
  then
        echo "Your grade is 'C'"
  elif (($1 < 60));
  then
        echo "Your grade is 'F'"
  fi
}
echo "Enter your percentage % to find out your grade:"
read percent
grade $percent
####################################################
#avg()
#{
```

```
kamisha@kamisha11070-virtualbox:~/Desktop$ ./Assign1.sh
Enter your percentage % to find out your grade:
67
Your grade is 'C'
kamisha@kamisha11070-virtualbox:~/Desktop$ ./Assign1.sh
Enter your percentage % to find out your grade:
80
Your grade is 'B+'
kamisha@kamisha11070-virtualbox:~/Desktop$ ./Assign1.sh
Enter your percentage % to find out your grade:
89
Your grade is 'A'
kamisha@kamisha11070-virtualbox:~/Desktop$ ./Assign1.sh
Enter your percentage % to find out your grade:
72
Your grade is 'B'
kamisha@kamisha11070-virtualbox:~/Desktop$ ./Assign1.sh
Enter your percentage % to find out your grade:
69
Your grade is 'C+'
kamisha@kamisha11070-virtualbox:~/Desktop$
```

Part – j: Check if a value is divisible by 5 or not



```
#       echo "This number is an Armstrong Number!"
#   else
#       echo "This number is not an Armstrong Number!"
#   fi
#}
#echo "Enter any number to check whether it's an Armstrong Number o
#read val1
#arm $val1
###################################################
five()
{
  val1=$1
  if (($1 % 5 == 0));
  then
        echo "This number is divisible by 5!"
  elif (($1 % 5 != 0));
  then
        echo "This number is not divisible by 5!"
  fi
}
echo "Enter a number to check whether it's divisible by 5 or not:"
read val1
five $val1
###################################################
#grade()
#{
#   percent=$1
#   if (($1 >= 87 && $1 <= 100));
#   then
#       echo "Your grade is 'A'"
#   elif (($1 >= 76 && $1 <= 86));
#   then
```

```
kamisha@kamisha11070-virtualbox:~/Desktop$ ./Assign1.sh
Enter a number to check whether it's divisible by 5 or not:
45
This number is divisible by 5!
kamisha@kamisha11070-virtualbox:~/Desktop$ ./Assign1.sh
Enter a number to check whether it's divisible by 5 or not:
78
This number is not divisible by 5!
kamisha@kamisha11070-virtualbox:~/Desktop$ ./Assign1.sh
Enter a number to check whether it's divisible by 5 or not:
100
This number is divisible by 5!
kamisha@kamisha11070-virtualbox:~/Desktop$ ./Assign1.sh
Enter a number to check whether it's divisible by 5 or not:
123
This number is not divisible by 5!
kamisha@kamisha11070-virtualbox:~/Desktop$
```