MM2090

Assignment 1 – Question 1

Archish S
me20b032
Batch 7

1. Download and use RollList.csv for this question. Assume that the hostel fee of freshers is incremented by 10 percent of the previous year's fees each year. This fee you pay as a fresher continues to be paid every semester after that without increment. E.g., fees for students who joined in 2018 will be lower than those who joined in 2019, but the 2019 batch will pay the same fees every semester and so on. The students who joined in the year 2017 had to pay INR 20,000 as hostel fees.
   The solution includes a script to calculate, and the output of total fees received this semester by the hostel administration from the students on the Roll List. [2 Marks]
   Hint: You can use a shell script or awk script to achieve this.
   Application: Minor numerical operations on a regular dataset in a CSV file are typical in data analysis.

Link to the GitHub repository for this question: GitHub

This awk script takes in the RollList.csv and preprocesses each line, keeping count of number of students from each year.

```
#!/usr/bin/gawk -f
BEGIN{
  FS = ",";
  sum = 0;
};

{
  # Saving the roll number in the variable roll
  roll = $1;

  # Stripping the 3rd and 4th characters of the roll number
  year = int(substr(roll, 3, 2));

  # Incrementing the number of students belonging to that year
  num[year]++;

};

END{
  for (year in num){

            # Calculating the rate first then the sum - doing in one step gave scientific representation
            rate = (1.1)**(year-17)*num[year];
            sum+= 20000*rate;
  };
            print sum;
};
```

This bash script calls the fess.awk wrapper to find the fees paid by the students listed in RollList.csv

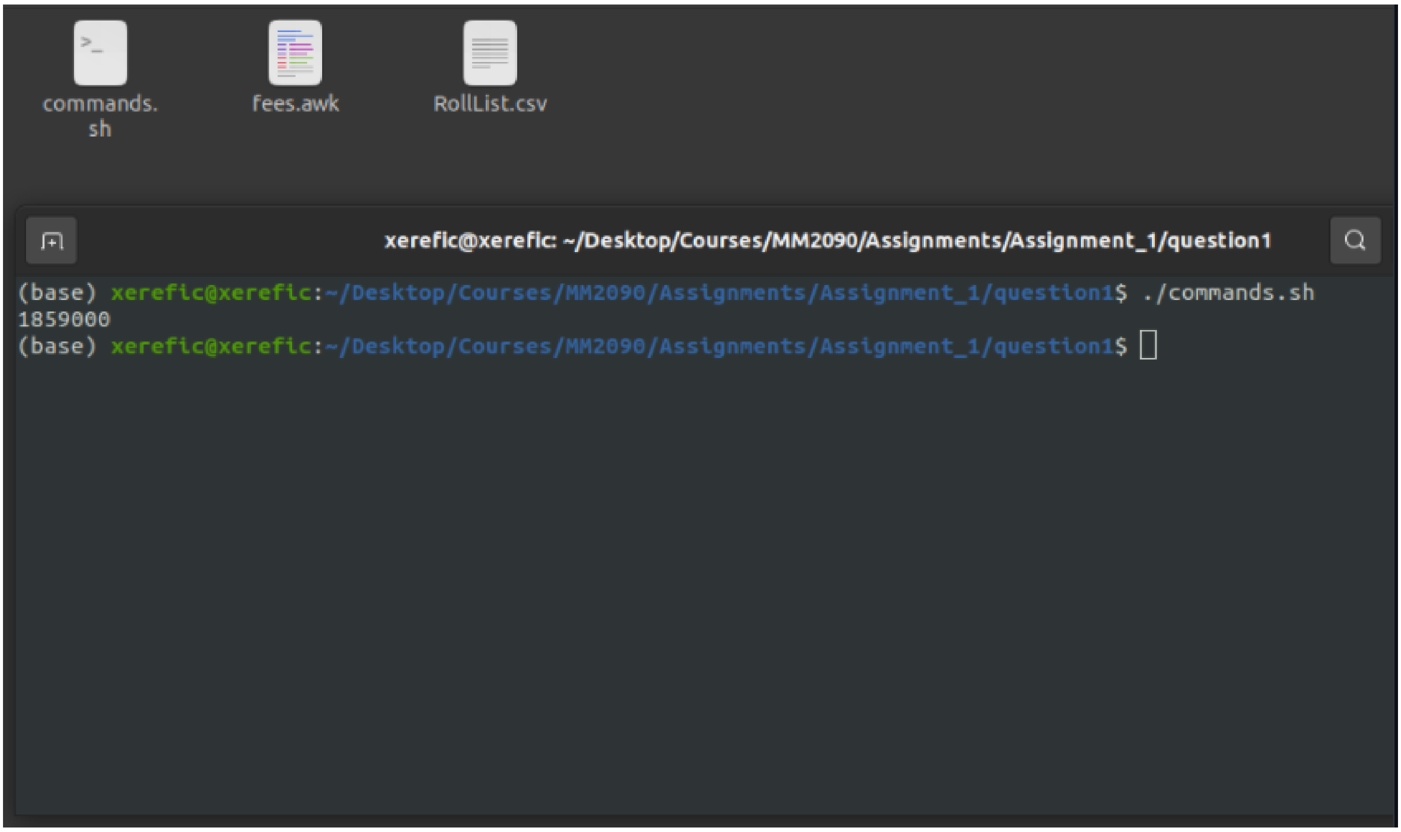
```
#!/bin/bash

./fees.awk < RollList.csv

```

MM2090

Assignment 1 – Question 1

Archish S
me20b032
Batch 7

TERMINAL:



commands.sh
fees.awk
RollList.csv
xerefic@xerefic: ~/Desktop/Courses/MM2090/Assignments/Assignment_1/question1

```
(base) xerefic@xerefic:~/Desktop/Courses/MM2090/Assignments/Assignment_1/question1$ ./commands.sh
1859000
(base) xerefic@xerefic:~/Desktop/Courses/MM2090/Assignments/Assignment_1/question1$ 
```

OUTPUT:

```
1859000
```

MM2090

Assignment 1 – Question 2

Archish S
me20b032
Batch 7

2. As part of data file submission to a platform, the administrator needs to create a set of initial passwords for the students. The password creation is done in the following way. The first part of the password is department code in capital letters followed by the numeric part of the roll number (the last 3 characters of the roll number) followed by the length of the string in their name (this will also count the spaces). Followed by the program in which the student is enrolled in small letter.
   For example, if the roll number is **ME18B009** and the name is **Bharath Chandar** then the password will be **ME00915b**. Use the RollList.csv file available from the above question and write a script to generate the password for all the students.
   The solution includes the script and the password list. [2 Marks]
   Hint: You can use a shell script or awk script to achieve this.
   Application: Applications may need to generate clear text passwords according to certain rules to be given to first time users.

Link to the GitHub repository for this question: GitHub

This awk script takes in the RollList.csv and preprocesses each line and outputs the password in the correct format.

```
#!/usr/bin/gawk -f
BEGIN{
  FS = ",";

  # Printing the header
  printf("Roll,Name,Password\n");
};

{
  # Saving the roll number in the variable roll
  roll = $1;
  # Saving the name in the variable name
  name = $2;

  # Stripping the 1st and 2nd characters of the roll number
  depart = substr(roll, 1, 2);
  # Stripping the last three characters of the roll number
  id = substr(roll, 6, 3);
  # Stripping the 5th characrer of the roll number and storing its lower case
  group = tolower(substr(roll, 5, 1));

  # Finding the length of name
  len_name = length(name);

  printf("%s,%s,%s%s%s%s\n", roll, name, depart, id, len_name, group);

};

END{
};

```

This bash scripts calls the password.awk wrapper to get the passwords of the students in the RollList.csv

```
#!/bin/bash

./password.awk < RollList.csv > Password.csv

```

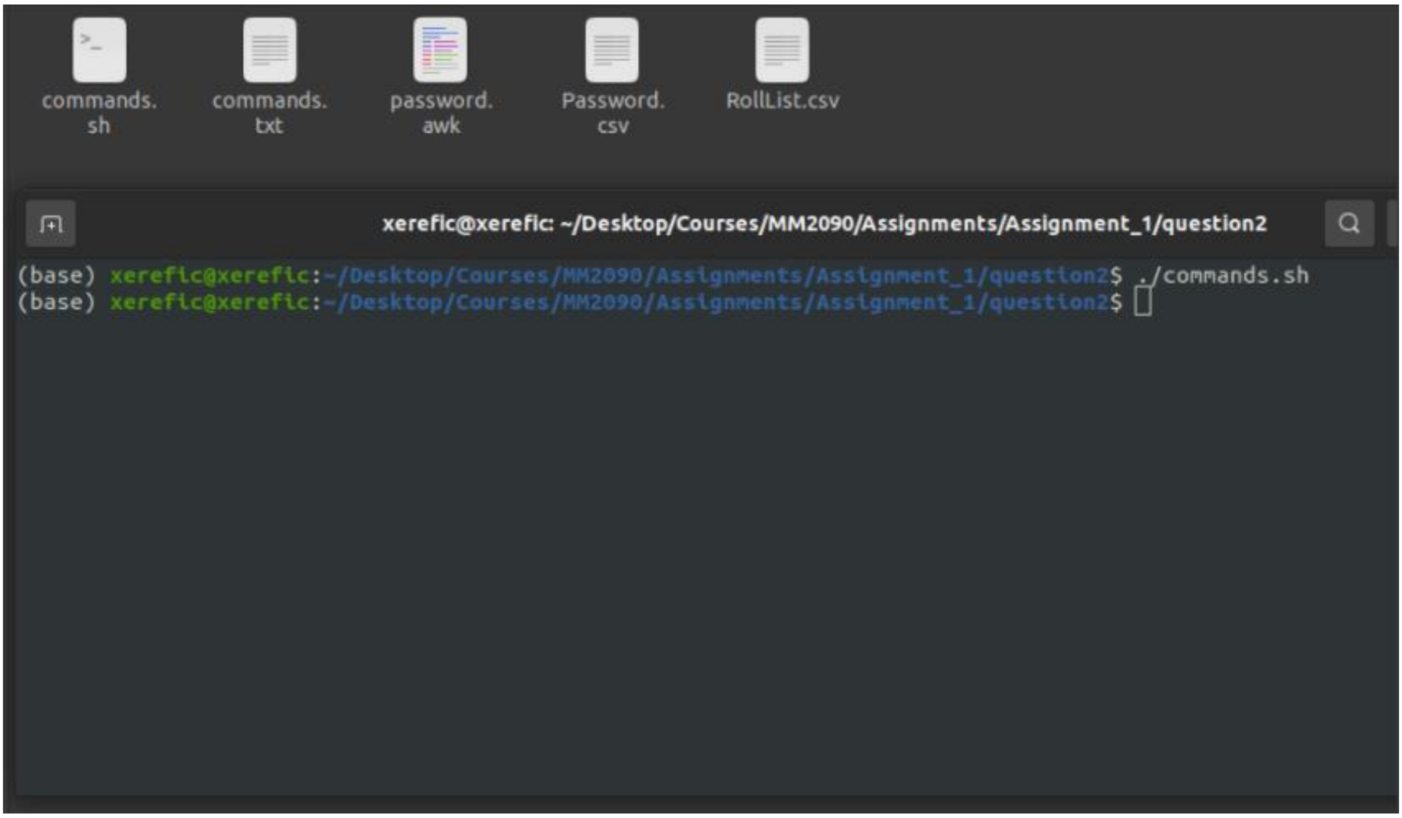
Using this command will display only the passwords which can be piped to a file accordingly.

```
cat Password.csv | cut -d ',' -f 3
```

MM2090 # Assignment 1 – Question 2

Archish S
me20b032
Batch 7

TERMINAL:


commands.sh
commands.txt
password.awk
Password.csv
RollList.csv
xerefic@xerefic: ~/Desktop/Courses/MM2090/Assignments/Assignment_1/question2

```
(base) xerefic@xerefic:~/Desktop/Courses/MM2090/Assignments/Assignment_1/question2$ ./commands.sh
(base) xerefic@xerefic:~/Desktop/Courses/MM2090/Assignments/Assignment_1/question2$ 
```

PS: Commands.txt was another approach that I was trying.

OUTPUT:

Roll	Name	Password
CE19B089	Sruthi Sreeram	CE08914b
ME18B009	Bharath Chandar	ME00915b
ME18B020	Aravindh P	ME02010b
ME18B027	Rajasundaram M	ME02714b
ME18B033	Suganth NN	ME03310b
ME18B046	Deepak G	ME0468b
ME18B086	Arvind Raghav V	ME08615b
ME18B089	Sriharan BS	ME08911b
ME18B145	Ashwin Kumar KS	ME14515b
ME18B146	Vikas Mahendar K	ME14616b
ME18B152	Manoj S	ME1527b
ME18B156	Mukesh V	ME1568b
ME18B157	Mukund Varma T	ME15714b
ME19B001	Abhigyan Chattopadhyay	ME00122b
ME19B003	Akshay Sunil	ME00312b
ME19B003	Akshay Sunil	ME00312b
ME19B012	Dhananjay B	ME01211b
ME19B013	Dhruv Bhatia	ME01312b
ME19B023	Krishna Somasundaram RM	ME02323b
ME19B035	Shreyas S. Paranjape	ME03520b
ME19B036	Parth Lathi	ME03611b
ME19B048	Shyam Sundar PB	ME04815b
ME19B052	Tadeparti Sidharth	ME05218b
ME19B057	Vasudeva Tushar Nilker	ME05722b
ME19B060	Vishalroshan A	ME06014b
ME19B061	Pavithra Mohan	ME06114b
ME19B063	Aditi Kumar	ME06311b
ME19B076	Alphin Davis Pomy	ME07617b
ME19B079	Ananthalakshmi N	ME07916b
ME19B082	Arvind Car	ME08210b
ME19B085	Ashwin Upamanyu Dev	ME08519b
ME19B091	Aswin Raj CR	ME09112b
ME19B095	Krishnathejus D	ME09515b
ME19B098	Abhishek Dhalpe	ME09815b
ME19B105	Gautham Raghav S	ME10516b
ME19B106	Gowrishankar Manu	ME10617b
ME19B107	Gunda Mahathi	ME10713b
ME19B108	Harsh Gupta	ME10811b
ME19B114	Jay Pratik Sanghavi	ME11419b
ME19B121	Kalisytharth SK	ME12115b
ME19B123	Karthik Karumanchi	ME12318b
ME19B128	Srividya L	ME12810b
ME19B131	Manasa Madela	ME13113b
ME19B135	Neelanjana Pramod	ME13517b
ME19B149	Prasanth S	ME14910b
ME19B150	Prathamesh A Jain	ME15017b
ME19B152	Preetha Devi A	ME15214b
ME19B155	Rohan H Rao	ME15511b
ME19B159	Sachin Sagar	ME15912b
ME19B163	Saroopa G	ME1639b
ME19B167	Shreyas Madhav Kulkarni	ME16723b
ME19B175	Srinidhi N	ME17510b
ME19B176	Subham Khan	ME17611b
ME19B177	Suraj Rathi	ME17711b
ME19B179	Tejas Rao M	ME17911b
ME19B185	Vedant Limaye	ME18513b
MM18B034	Shrutasom Das	MM03413b
MM19B002	Akshat Joshi	MM00212b
MM19B005	Bipin V	MM0057b
MM19B009	Kanishkan MS	MM00912b
MM19B012	Pragalbh Vashishtha	MM01219b
MM19B014	Vishnupriyan R	MM01414b
MM19B015	Jai Santhoshi S	MM01515b
MM19B016	Aakhyat Singh	MM01613b
MM19B018	Aditya Raj	MM01810b
MM19B021	Ayesha Ulde	MM02111b
MM19B027	Bhuvanesh P	MM02711b
MM19B029	Chirantandip Mahanta	MM02920b
MM19B031	Gayathri Shrushti V	MM03119b
MM19B033	Hrishabh Srivastava	MM03319b
MM19B038	Mahesh R	MM0388b
MM19B040	Nagappan N	MM04010b
MM19B043	Hardhik Pinjala	MM04315b
MM19B044	Pranav Choudhari	MM04416b
MM19B045	Aswanth R	MM0459b
MM19B046	Rishaab Karthik R	MM04617b
MM19B049	Rohan Korale	MM04912b
MM19B054	Shreya Smitha	MM05413b

MM2090

Assignment 1 – Question 3

Archish S
me20b032
Batch 7

3. Download the file screenshots.tar.gz from course page on the moodle site. The screenshots contain images of the terminal window with few commands that expose the hardware / configuration details of my laptop I use for this course instruction. You need to make a clean pdf that contains two terminal images per page. This involves trimming of the images. Your script should take the pixel values for trimming as a user input and then do the task automatically for all the images in the directory and generate the pdf.
   The solution includes the script and one sample page the output pdf. [2 Marks]
   Hint: Install imagemagick with `sudo apt install imagemagick` and check man page of `convert` command.
   Application: You may need to trim a large set of images from an experiment and use them for image analysis, to create a video sequence and to extract events. High speed imaging of experiments is an important scientific technique.

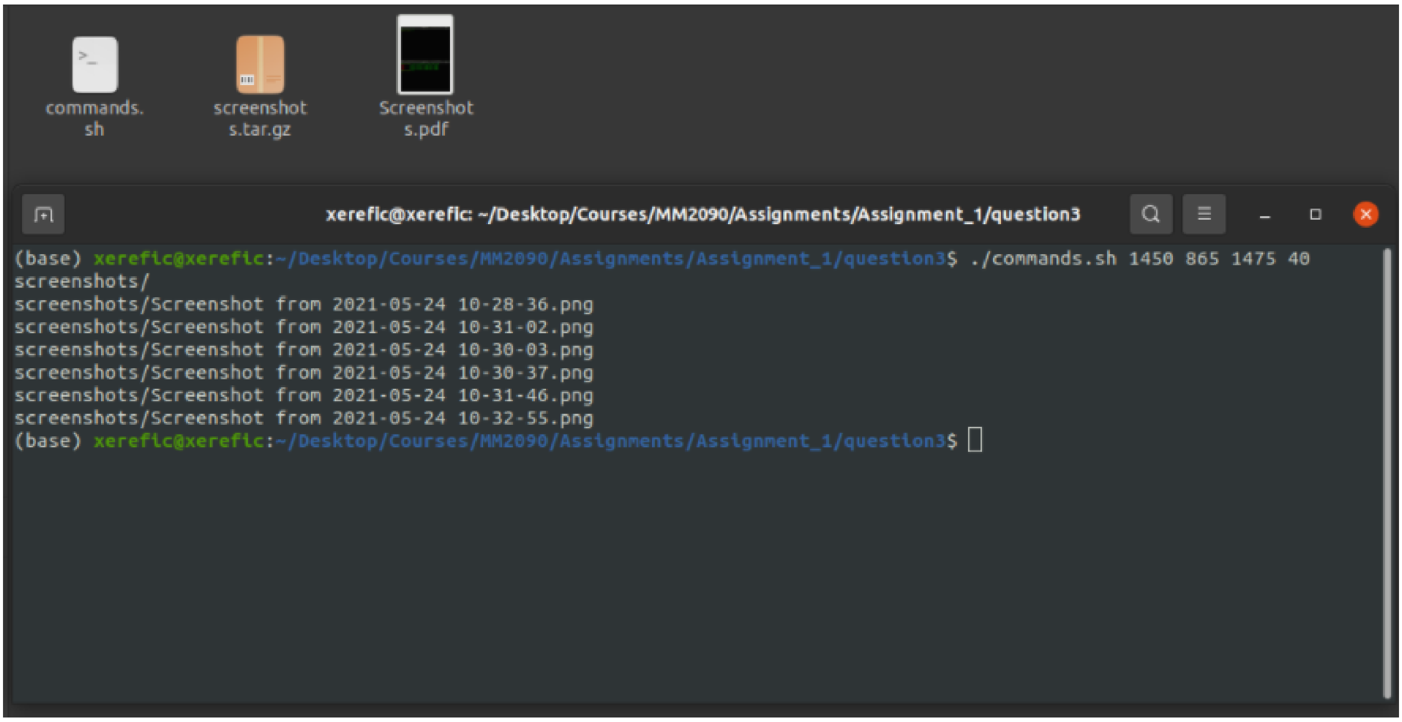
Link to the GitHub repository for this question: GitHub

The bash script takes parameters – width, height and strides and outputs the pdf of the cropped images.

```
#!/bin/bash

# Unzipping the screenshots
tar -xvzf screenshots.tar.gz
mkdir cropped

# Renaming the files for ease of access
cd screenshots/
for f in *\ *; do mv "$f" "${f// /_}"; done
i=0;
for files in `ls`;
do
  mv $files $(basename $i).png;
  i=$(( $i+1 ));
done;

# size: 1450x865+1475+40
cd ..

# Cropping the images
for files in `ls screenshots/`;
do
  if [ -z "$1" ]; then
            width=`identify -ping -format '%w' screenshots/$files`
            height=`identify -ping -format '%h' screenshots/$files`
            x=0
            y=0
  else
            width=$1
            height=$2
            x=$3
            y=$4
  fi
  convert screenshots/$files -crop $width\x$height+$x+$y cropped/$(basename $files .png).png
done;

rm -r screenshots/
mkdir combined

# Combining two images into one by appending them vertically
for j in $(seq 0 $(($i/2-1)) );
do
  convert -append cropped/$(basename $(($j*2))).png cropped/$(basename $(($j*2+1))).png combined/$(basename $j).png
done;

rm -r cropped/

# Joining the combined images into a pdf
convert -page A4 -resize 3508x2480 combined/*.png Screenshots.pdf
```

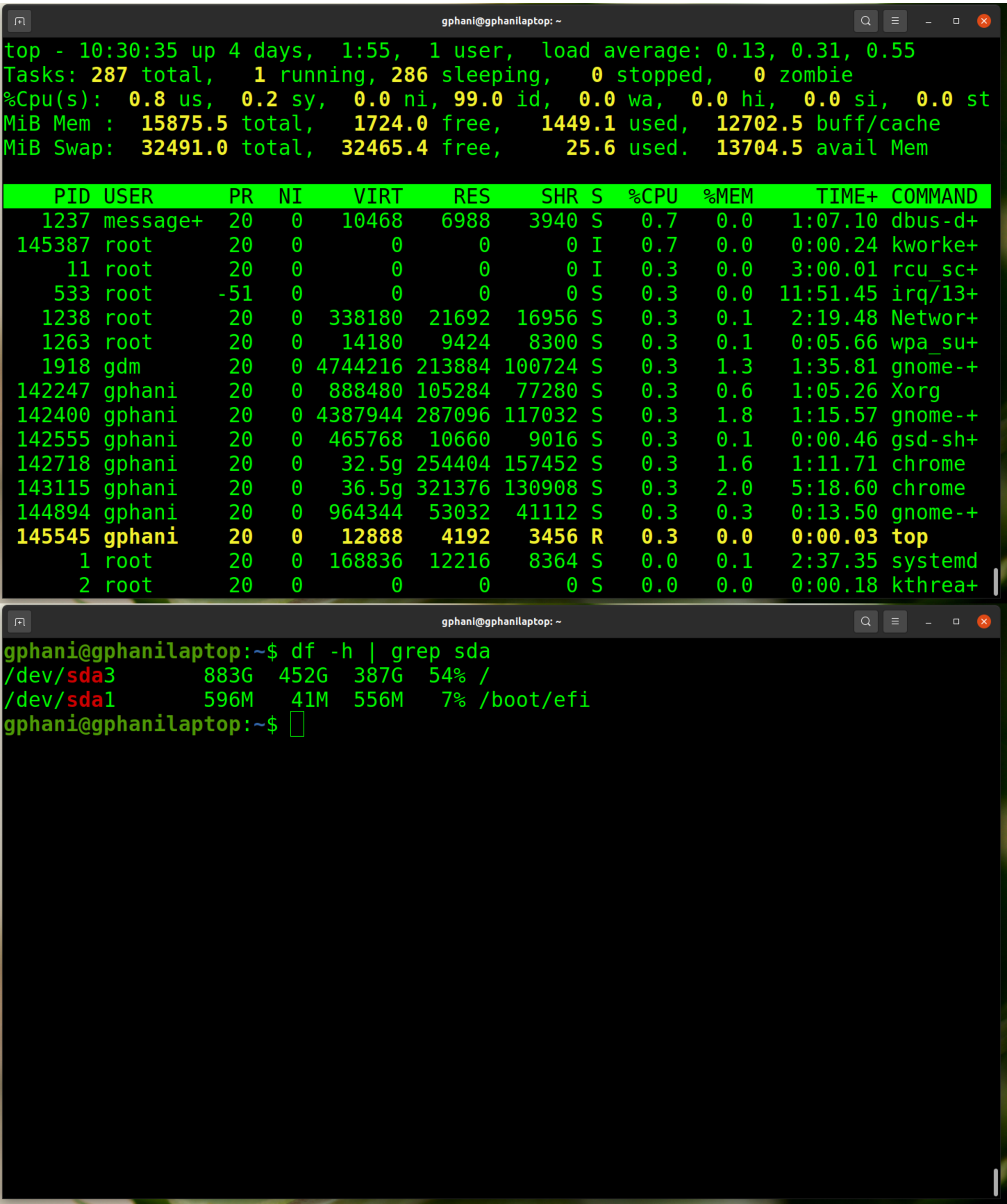
```
rm -r combined/

```

TERMINAL:


commands.sh
screenshots.tar.gz
Screenshots.pdf
xeretic@xeretic: ~/Desktop/Courses/MM2090/Assignments/Assignment_1/question3

```
(base) xeretic@xeretic:~/Desktop/Courses/MM2090/Assignments/Assignment_1/question3$ ./commands.sh 1450 865 1475 40
screenshots/
screenshots/Screenshot from 2021-05-24 10-28-36.png
screenshots/Screenshot from 2021-05-24 10-31-02.png
screenshots/Screenshot from 2021-05-24 10-30-03.png
screenshots/Screenshot from 2021-05-24 10-30-37.png
screenshots/Screenshot from 2021-05-24 10-31-46.png
screenshots/Screenshot from 2021-05-24 10-32-55.png
(base) xeretic@xeretic:~/Desktop/Courses/MM2090/Assignments/Assignment_1/question3$ 
```

OUTPUT:



```
top - 10:30:35 up 4 days,  1:55,  1 user,  load average: 0.13, 0.31, 0.55
Tasks: 287 total,   1 running, 286 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.8 us,  0.2 sy,  0.0 ni, 99.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  15875.5 total,   1724.0 free,   1449.1 used,  12702.5 buff/cache
MiB Swap:  32491.0 total,  32465.4 free,     25.6 used.  13704.5 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   1237 message+  20   0   10468   6988   3940 S   0.7   0.0   1:07.10 dbus-d+
 145387 root      20   0       0      0      0 I   0.7   0.0   0:00.24 kworke+
     11 root      20   0       0      0      0 I   0.3   0.0   3:00.01 rcu_sc+
    533 root     -51   0       0      0      0 S   0.3   0.0  11:51.45 irq/13+
   1238 root      20   0  338180  21692  16956 S   0.3   0.1   2:19.48 Networ+
   1263 root      20   0   14180   9424   8300 S   0.3   0.1   0:05.66 wpa_su+
   1918 gdm       20   0 4744216 213884 100724 S   0.3   1.3   1:35.81 gnome-+
 142247 gphani    20   0  888480 105284  77280 S   0.3   0.6   1:05.26 Xorg
 142400 gphani    20   0 4387944 287096 117032 S   0.3   1.8   1:15.57 gnome-+
 142555 gphani    20   0  465768  10660   9016 S   0.3   0.1   0:00.46 gsd-sh+
 142718 gphani    20   0   32.5g 254404 157452 S   0.3   1.6   1:11.71 chrome
 143115 gphani    20   0   36.5g 321376 130908 S   0.3   2.0   5:18.60 chrome
 144894 gphani    20   0  964344  53032  41112 S   0.3   0.3   0:13.50 gnome-+
 145545 gphani    20   0   12888   4192   3456 R   0.3   0.0   0:00.03 top
      1 root      20   0  168836  12216   8364 S   0.0   0.1   2:37.35 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.18 kthrea+
```

```
gphani@gphanilaptop:~$ df -h | grep sda
/dev/sda3       883G  452G  387G  54% /
/dev/sda1       596M   41M  556M   7% /boot/efi
gphani@gphanilaptop:~$ 
```

MM2090

Assignment 1 – Question 4

Archish S
me20b032
Batch 7

4. Discuss within your group and pick one hardware item type that is different from the rest of the group. The groupings are listed in the roll list in the folder for the course on google drive. Look up the internet to identify at least 3 latest entries of that hardware item and put down their specs. Comment what kind of gadgets / workstations / servers / clusters from OEMs include such a hardware item. Give the source of your information.
(a) CPU (b) GPU (c) DRAM (d) SATA hard disc (e) Network switches for cluster (f) Network card.
The solution includes a tabular listing of specs along with source of information as a reference below. [2 Marks]
Hint: Start with leading OEMs for these items using Wikipedia and then look up latest releases.
Application: One should know hardware limitations for program execution as part of scientific computing in real life.

Graphics Processing Unit

A **graphics processing unit** (**GPU**) is a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device.[1] GPUs are used in embedded systems, mobile phones, personal computers, workstations, and game consoles. Modern GPUs are very efficient at manipulating computer graphics and image processing.

History[2] (Decade)

1970-1980

Arcade system boards have been using specialized graphics circuits since the 1970s. In early video game hardware, the RAM for frame buffers was expensive, so video chips composited data together as the display was being scanned out on the monitor.

1980-1990

The NEC µPD7220 was the first implementation of a PC graphics display processor as a single Large Scale Integration (LSI) integrated circuit chip, enabling the design of low-cost, high-performance video graphics cards such as those from Number Nine Visual Technology.

1990-2000

All major PC graphics chip makers had added 2D acceleration support to their chips. Real-time 3D graphics were becoming increasingly common in arcade, computer and console games, which led to an increasing public demand for hardware-accelerated 3D graphics.

Software implementations of OpenGL were common during this time, although the influence of OpenGL eventually led to widespread hardware support.

2000-2010

Nvidia was first to produce a chip capable of programmable shading. The introduction of the ATI Radeon 9700, the world's first Direct3D 9.0 accelerator, pixel and vertex shaders could implement looping and lengthy floating point math, and were quickly becoming as flexible as CPUs, yet orders of magnitude faster for image-array operations.

Nvidia's CUDA platform, first introduced in 2007, was the earliest widely adopted programming model for GPU computing.

2010-Present

Nvidia launched the RTX 20 series GPUs that add ray-tracing cores to GPUs, improving their performance on lighting effects.

[1] Source: Wikipedia

[2] Source: Wikipedia

AMD released the successor to their Graphics Core Next (GCN) microarchitecture/instruction set. Dubbed as RDNA, the first product line up featuring the first generation of RDNA was the Radeon RX 5000 series of video cards

Types of GPUs[3]

Dedicated Graphics Cards

The GPUs of the most powerful class typically interface with the motherboard by means of an expansion slot such as PCI Express (PCIe). The term "dedicated" refers to the fact that dedicated graphics cards have RAM that is dedicated to the card's use. This RAM is usually specially selected for the expected serial workload of the graphics card (see GDDR).

Technologies such as SLI and NVLink by Nvidia and CrossFire by AMD allow multiple GPUs to draw images simultaneously for a single screen, increasing the processing power available for graphics.

Integrated Graphics Cards

Integrated graphics processors (IGPs) utilize a portion of a computer's system RAM rather than dedicated graphics memory. IGPs can be integrated onto the motherboard as part of the (northbridge) chipset, or on the same die (integrated circuit) with the CPU (like AMD APU or Intel HD Graphics).

Applications[4]

Most GPUs are designed for a specific usage, real-time 3D graphics or other mass calculations:

Gaming

- GeForce GTX, RTX
- Nvidia Titan
- Radeon HD, R5, R7, R9, RX, Vega and Navi series
- Radeon VII

Cloud Gaming

- Nvidia GRID
- Radeon Sky

Workstation

- Nvidia Quadro
- AMD FirePro
- AMD Radeon Pro

Cloud Workstation

- Nvidia Tesla
- AMD FireStream

Artificial Intelligence training and Cloud

- Nvidia Tesla
- AMD Radeon Instinct

Automated/Driverless car

- Nvidia Drive PX

Major Manufacturers [5]

- Nvidia

[3] Source: Wikipedia
[4] Source: Wikipedia
[5] Source

- AMD
- Asus
- Intel
- EVGA
- Gigabyte

Latest Entries[6]

Manufacturer and Model	Launch Date	Specifications
Nvidia GeForce RTX 3060 Ti	February 25, 2021	➢ Architecture: Ampere ➢ Stream Processors: 4,864 ➢ Core Clock: 1.41 GHz (1.67 GHz boost) ➢ Memory: 8 GB GDDR6 ➢ Memory Clock: 14Gbps ➢ Outputs: HDMI 2.1, 3x DisplayPort 1.4a ➢ Power Connectors: 1x PCIe 8-pin ➢ Required Power: 600W
Nvidia GeForce RTX 3090	September 24, 2020	➢ Architecture: Ampere ➢ Stream Processors: 10,496 ➢ Core Clock: 1.40 GHz (1,70 GHz boost) ➢ Memory: 24 GB GDDR6X ➢ Memory Clock: 19.5Gbps ➢ Outputs: HDMI 2.1, 3x DisplayPort 1.4a ➢ Power Connectors: 2x PCIe 8-pin ➢ Required Power: 750W
AMD Radeon RX 6700	March 21, 2021	➢ Architecture: RDnA2 ➢ Stream Processors: 2,560 ➢ Core Clock: 2.32 GHz (2.58 GHz boost) ➢ Memory: 12GB GDDR6 ➢ Memory Clock: 16 Gbps ➢ Power Connectors: 8 pin + 6 pin ➢ Outputs: HDMI 2.1, DisplayPort 1.4 with DSC ➢ Required Power: 230W
AMD Radeon RX 6900 XT	December 8, 2020	➢ Architecture: RDnA2 ➢ Stream Processors: 5,210 ➢ Core Clock: 1.82 GHz (2.25 GHz boost) ➢ Memory: 16GB GDDR6 ➢ Memory Clock: 16Gbps ➢ Power Connectors: 2 x 8 pin ➢ Outputs: DisplayPort 1.4 with DSC, HDMI 2.1 VRR and FRL ➢ Required Power: 300W

[6] Source

MM2090

Assignment 1 – Question 5

Archish S
me20b032
Batch 7

5. Download the transcripts of text chat from the google drive folder for this course into a folder. Create a script that can calculate your attendance as a percentage by looking for your roll number in each of the files.
The solution includes the script, total number of transcript files considered, lecture numbers missed and attendance percentage. [2 Marks]
Hint: Use the features of grep and wc for the task.
Application: When you run a large FEM program, you receive a log output in which certain warnings will be listed about change of algorithm or convergence failure etc., Looking for their occurrence and analysis is a part of engineering simulation work.

Link to the GitHub repository for this question: GitHub

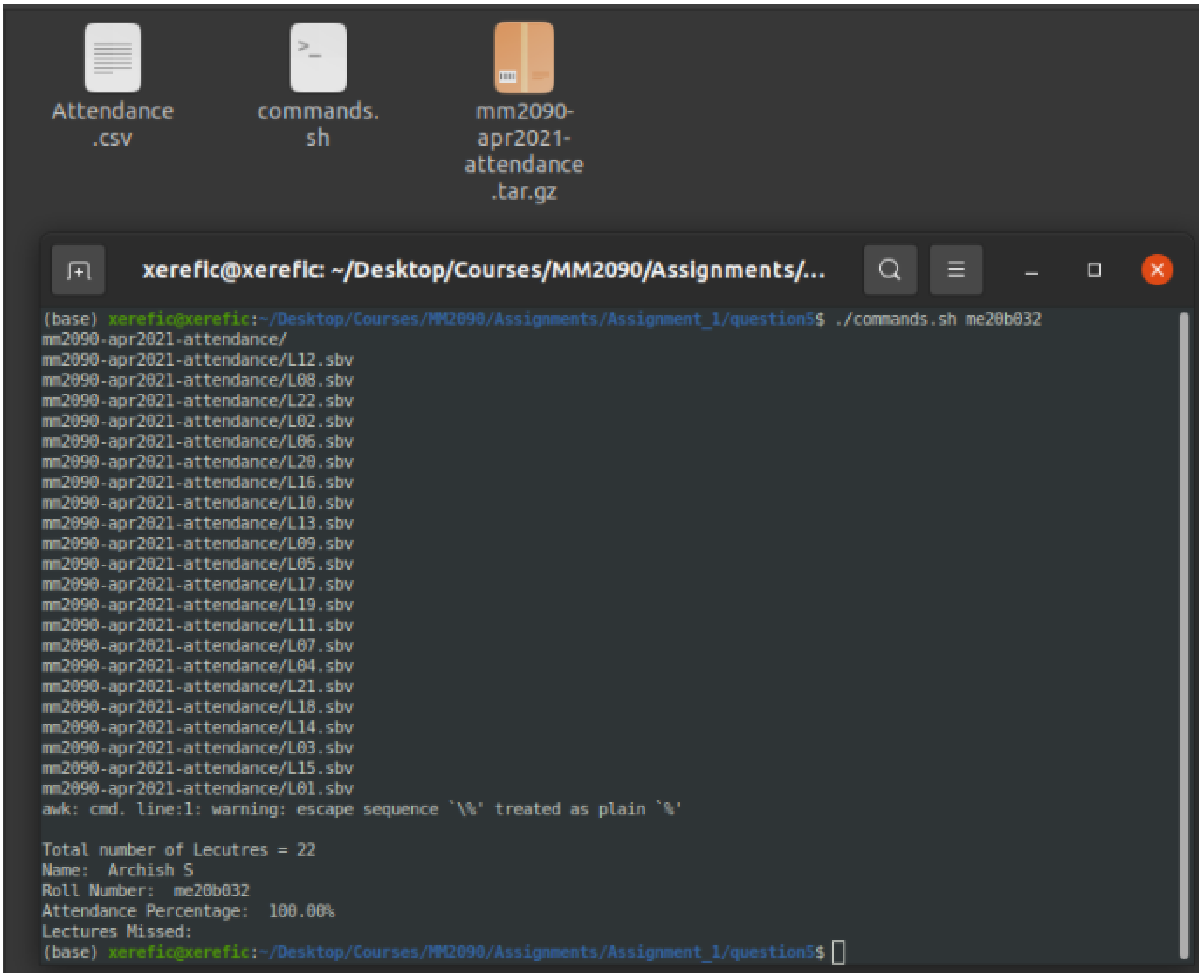
This bash automatically processes the contents of the zip file and gives the attendance report.
The bash script can also take one parameter – the roll number and give the report of that person.

```
#!/bin/bash

# Unzipping the chats
tar -xvzf mm2090-apr2021-attendance.tar.gz
mv mm2090-apr2021-attendance/ transcripts/

# Grepping the name and roll number of students
mkdir registered
lectures=0;
for files in `ls transcripts/`;
do
  lectures=$(( $lectures+1 ));
  cat transcripts/$files | grep -oe '^\(.*\)..[[:digit:]][[:digit:]][[:alpha:]]...:' > registered/$(basename $files .sbv).txt
done;
rm -r transcripts/

# Pre-processing
for files in `ls registered/`;
do
        cat registered/$files | sed -e 's/\(.*\) \(.*\):/\2,\1/g'  > registered/$(basename $files .txt).csv
done;
rm registered/*.txt


mkdir cache
for files in `ls registered/`;
do
  cat registered/$files | sed -e 's/ /_/g' > cache/$(basename $files .csv).csv
done;
rm -r registered/

# Removing multiple instances for each lecture
mkdir attendance
for files in `ls cache/`;
do
  cat cache/$files | awk -F, '{name[$1]=$2;}END{for (id in name){printf("%s,%s\n", id, name[id]);}}' | sort -k1 -n > attendance/$(basename $files .csv).csv
done;
rm -r cache/

# Finding the total registerants
tail -n+1 -q attendance/*.csv | awk -F, '{name[$1]=$2;}END{for (id in name){printf("%s,%s\n", id, name[id]);}}' | sort -k1 -n > registered.csv

# Finding the percentage of attendance
tail -n+1 -q attendance/*.csv | awk -F, -v total=$lectures '{data[$1]=$0;name[$1]++;}END{for (id in name){printf("%s,%.2f\%\n", data[id], name[id]/total*100);}}' | sort -k1 -n > attendance.csv

# Finding absentees in every lecture
```

```
mkdir absent
for files in `ls attendance/`
do
  diff attendance/$files registered.csv | egrep '....b...' | sed 's/> \(.*\)/\1/g' | awk -v lecture=$(basename $files .csv) '{printf("%s,%s\n", $0, lecture)}' > absent/$(basename $files .csv).csv;
done;
rm registered.csv
rm -r attendance/

# Processing the absentee list
tail -n+1 -q absent/*.csv | awk -F, '{name[$1]=$2;absent[$1]=absent[$1]$3;}END{for (id in name){printf("%s,%s,%s\n", id, name[id], absent[id]);}}' | sort -k1 -n > absent.csv
rm -r absent/

# Combining the data
awk -F, 'NR==FNR {absent[$1]=$3; next}{printf("%s,%s\n", $0, absent[$1]);}' absent.csv attendance.csv > cache.csv
rm absent.csv
rm attendance.csv

cat cache.csv | sed 's/_/ /g' | sed 's/\L/ L/g'  > final.csv
rm cache.csv

# Adding header
awk -F, 'BEGIN{printf("Roll,Name,Percentage,Missed Lectures\n");}{print $0;}' < final.csv > Attendance.csv
rm final.csv

echo

echo "Total number of Lecutres = $lectures"

if [ -z "$1" ]; then
  echo
else
  cat Attendance.csv | grep $1 | gawk -F, '{print "Name: ", $2; print "Roll Number: ", $1; print "Attendance Percentage: ", $3; print "Lectures Missed: ", $4;}'
fi

```

Assignment 1 – Question 5

TERMINAL:


Attendance.csv
commands.sh
mm2090-apr2021-attendance.tar.gz
xereflc@xereflc: ~/Desktop/Courses/MM2090/Assignments/...

```
(base) xerefic@xerefic:~/Desktop/Courses/MM2090/Assignments/Assignment_1/question5$ ./commands.sh me20b032
mm2090-apr2021-attendance/
mm2090-apr2021-attendance/L12.sbv
mm2090-apr2021-attendance/L08.sbv
mm2090-apr2021-attendance/L22.sbv
mm2090-apr2021-attendance/L02.sbv
mm2090-apr2021-attendance/L06.sbv
mm2090-apr2021-attendance/L20.sbv
mm2090-apr2021-attendance/L16.sbv
mm2090-apr2021-attendance/L10.sbv
mm2090-apr2021-attendance/L13.sbv
mm2090-apr2021-attendance/L09.sbv
mm2090-apr2021-attendance/L05.sbv
mm2090-apr2021-attendance/L17.sbv
mm2090-apr2021-attendance/L19.sbv
mm2090-apr2021-attendance/L11.sbv
mm2090-apr2021-attendance/L07.sbv
mm2090-apr2021-attendance/L04.sbv
mm2090-apr2021-attendance/L21.sbv
mm2090-apr2021-attendance/L18.sbv
mm2090-apr2021-attendance/L14.sbv
mm2090-apr2021-attendance/L03.sbv
mm2090-apr2021-attendance/L15.sbv
mm2090-apr2021-attendance/L01.sbv
awk: cmd. line:1: warning: escape sequence `\%' treated as plain `%'

Total number of Lecutres = 22
Name:  Archish S
Roll Number:  me20b032
Attendance Percentage:  100.00%
Lectures Missed:
(base) xerefic@xerefic:~/Desktop/Courses/MM2090/Assignments/Assignment_1/question5$ 
```

OUTPUT:

```
Total number of Lectures = 22
Name: Archish S
Roll Number: me20b032
Attendance Percentage: 100%
Lectures Missed:
```

MM2090

Assignment 1 – Question 5

Archish S
me20b032
Batch 7

OUTPUT:

Roll	Name	Percentage	Missed Lectures
me20b001	Aahan Bhargava	77.27%	L15 L16 L18 L21 L22
me20b004	Abhaumika Bijudith	22.73%	L01 L02 L05 L07 L08 L09 L10 L11 L12 L13 L14 L16 L17 L18 L19 L21 L22
me20b005	Abheshek Paramanand Kamble	86.36%	L06 L17 L19
me20b008	Abhishek Yadav	81.82%	L04 L06 L17 L22
me20b014	Aditya Kishore Dhoke	77.27%	L02 L04 L13 L17 L19
me20b017	Akhil Koshy Rajesh	81.82%	L10 L15 L19 L21
me20b020	Akshat Rakesh Garhwal	86.36%	L08 L09 L21
me20b021	Alpha P Jose	72.73%	L01 L03 L04 L10 L14 L19
me20b022	Amar Muhammed	9.09%	L01 L03 L04 L05 L06 L07 L08 L09 L10 L11 L12 L13 L14 L15 L16 L17 L18 L19 L21 L22
me20b024	Ankit Kumar	86.36%	L02 L03 L04
me20b027	Anshid K	31.82%	L06 L07 L08 L09 L11 L13 L14 L15 L16 L17 L18 L19 L20 L21 L22
me20b028	Anushka Asit Vadhavkar	100.00%	
me20b029	Anushka S	86.36%	L01 L15 L22
me20b032	Archish S	100.00%	
me20b036	Arun Palaniappan	95.45%	L03
me20b050	Cecil Jacob Thomas	95.45%	L04
me20b053	Chinmayee Tushar Kolhe	86.36%	L05 L08 L13
me20b055	Chris Joy Beck	40.91%	L01 L06 L08 L10 L12 L13 L15 L16 L17 L18 L19 L20 L21
me20b088	Jay Harish Shah	100.00%	
me20b112	Monisha C	90.91%	L03 L16
me20b132	Prabhat Bedida	86.36%	L04 L16 L22
me20b150	Rithwin K Ashraf	54.55%	L03 L05 L07 L10 L14 L15 L16 L18 L19 L21
me20b162	Shrid Suresh	27.27%	L04 L05 L06 L07 L08 L09 L10 L11 L12 L13 L16 L18 L19 L20 L21 L22
me20b163	Shriya Shukla	100.00%	
me20b166	Siddhagavali Shital Bhiku	90.91%	L05 L17
me20b178	Sukeerth Ramkumar	100.00%	
me20b183	Swapnil Paresh Mehta	100.00%	
mm20b005	Albin George	50.00%	L04 L06 L09 L11 L12 L13 L15 L17 L18 L21 L22
mm20b009	Bankar Niranjan Janardan	4.55%	L02 L03 L04 L05 L06 L07 L08 L09 L10 L11 L12 L13 L14 L15 L16 L17 L18 L19 L20 L21 L22
mm20b011	Bhagat Singh S	27.27%	L04 L05 L08 L09 L10 L11 L12 L13 L14 L15 L16 L17 L19 L20 L21 L22
mm20b017	Divya Jyothi D	86.36%	L01 L08 L16
mm20b019	Gatkal Siddhesh Sarjerao	95.45%	L07
mm20b020	Gokul C	81.82%	L04 L16 L18 L22
mm20b042	Nayanatara Deepak	81.82%	L01 L04 L13 L17
mm20b043	Nedunchezhiyan K	59.09%	L01 L02 L05 L07 L09 L10 L15 L19 L20
mm20b049	Prithviraj Pratap Bhosle	77.27%	L06 L09 L10 L13 L22
mm20b057	Shreya Rajesh	81.82%	L01 L02 L04 L17
mm20b059	Sumanth Manjunath Hegde	77.27%	L06 L08 L10 L16 L21

MM2090

Assignment 1 – Question 6

Archish S
me20b032
Batch 7

6. In the following URLs, the atomic radii are given as a list that you can readily copy paste into a csv file: Link 1 or Link 2, skip empty rows as needed. Pick an element that has the atomic number ending with the same digit as your roll number and list all elements whose atomic radii are within a certain percentage. Choose this percentage to be 10% to start with and increase if needed to be able to list at least 3 elements that are of comparable size.
   The solution includes your script, the element chosen, its radius in picometers, criterion for closeness in size and the list of elements that are close in size. [3 Marks]
   Hint: First create a csv file for input data and try a mix of shell and awk scripts.
   Application: Analysis of data available in the open domain in not so readily usable form is part of scientific computing research. Webpage scraping, scripts to atomate it and storing historical data is part of research.

Link to the GitHub repository for this question: GitHub

This awk script takes in the Atomic Radii data and outputs the elements whose radii is close to the selected element.

```
#!/usr/bin/gawk -f
BEGIN{
  FS = "\t";
  criterion = criterion/100;
  # Printing the header
  printf("Atomic Number,Element,Atomic Radii,Deviation\n")
  printf("%d,%s,%.2f,0\%\n", atomic, element, key);
};

{
  # Getting the absolute difference in atomic radii between the chosen element and iterated
  diff = $3-key;
  if (diff<0) diff = -1*diff;

  # Checking if the deviation is <10% and outputting if True
  if (diff<=key*criterion && $1!=32) printf("%d,%s,%.2f,%.2f\%\n", $1, $2, $3,diff/key*100);
};

END{
};

```

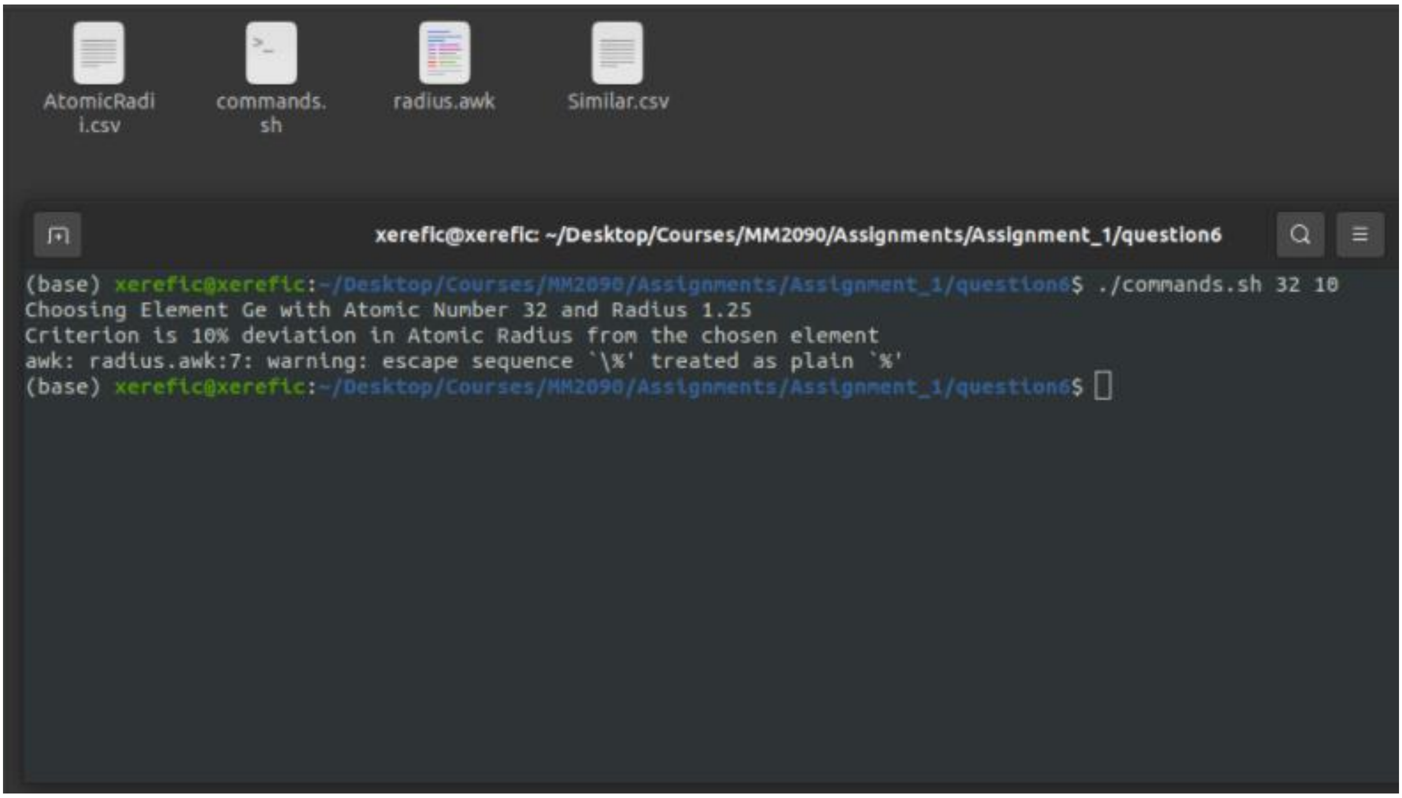
This bash takes the element of interest and the criterion as input and calls the radius.awk wrapper to get the elements with similar atomic radii.

```
#!/bin/bash

atomic=$1
element=`cat AtomicRadii.csv | sed 's/\t/,/g' | awk -v atomic=$atomic -F, '{if($1==atomic) print $2;}'`
key=`cat AtomicRadii.csv | sed 's/\t/,/g' | awk -v atomic=$atomic -F, '{if($1==atomic) print $3;}'`

echo "Choosing Element $element with Atomic Number $atomic and Radius $key"
echo "Criterion is $2% deviation in Atomic Radius from the chosen element"
awk -v atomic=$atomic -v element=$element -v key=$key -v criterion=$2 -f radius.awk < AtomicRadii.csv > Similar.csv

```

Element Chosen is Germanium (Atomic Number 32).

Assignment 1 – Question 6

TERMINAL:


AtomicRadii.csv
commands.sh
radius.awk
Similar.csv
xerefic@xerefic: ~/Desktop/Courses/MM2090/Assignments/Assignment_1/question6

```
(base) xerefic@xerefic:~/Desktop/Courses/MM2090/Assignments/Assignment_1/question6$ ./commands.sh 32 10
Choosing Element Ge with Atomic Number 32 and Radius 1.25
Criterion is 10% deviation in Atomic Radius from the chosen element
awk: radius.awk:7: warning: escape sequence `\%' treated as plain `%'
(base) xerefic@xerefic:~/Desktop/Courses/MM2090/Assignments/Assignment_1/question6$ 
```

OUTPUT:

```
Choosing Element Ge with Atomic Number 32 and Radius 1.25
Criterion is 10% deviation in Atomic Radius from the chosen element
```

Atomic Number	Element	Atomic Radii	Deviation
32	Ge	1.25	0%
13	Al	1.18	5.60%
31	Ga	1.36	8.80%
33	As	1.14	8.80%
51	Sb	1.33	6.40%
52	Te	1.23	1.60%
53	I	1.15	8.00%
84	Po	1.35	8.00%
85	At	1.27	1.60%
86	Rn	1.2	4.00%

INPUT (tab separated):

1 H 0.53 0.25 0.37 1.20 0.10
2 He 0.31 0.31 0.32 1.40
3 Li 1.67 1.45 1.34 1.82 0.90
4 Be 1.12 1.05 0.90 0.41
5 B 0.87 0.85 0.82 0.25
6 C 0.67 0.70 0.77 1.70 0.29
7 N 0.56 0.65 0.75 1.55 0.30
8 O 0.48 0.60 0.73 1.52 1.21
9 F 0.42 0.50 0.71 1.47 1.19
10 Ne 0.38 0.38 0.69 1.54
11 Na 1.90 1.80 1.54 2.27 1.16
12 Mg 1.45 1.50 1.30 1.73 0.86
13 Al 1.18 1.25 1.18 0.53
14 Si 1.11 1.10 1.11 2.10 0.40
15 P 0.98 1.00 1.06 1.80 0.31
16 S 0.88 1.00 1.02 1.80 0.43
17 Cl 0.79 1.00 0.99 1.75 1.67
18 Ar 0.71 0.71 0.97 1.88
19 K 2.43 2.20 1.96 2.75 1.52
20 Ca 1.94 1.80 1.74 1.14
21 Sc 1.84 1.60 1.44 0.89
22 Ti 1.76 1.40 1.36 0.75
23 V 1.71 1.35 1.25 0.68
24 Cr 1.66 1.40 1.27 0.76
25 Mn 1.61 1.40 1.39 0.81
26 Fe 1.56 1.40 1.25 0.69
27 Co 1.52 1.35 1.26 0.54
28 Ni 1.49 1.35 1.21 1.63 0.70
29 Cu 1.45 1.35 1.38 1.40 0.71
30 Zn 1.42 1.35 1.31 1.39 0.74
31 Ga 1.36 1.30 1.26 1.87 0.76
32 Ge 1.25 1.25 1.22 0.53
33 As 1.14 1.15 1.19 1.85 0.72
34 Se 1.03 1.15 1.16 1.90 0.56
35 Br 0.94 1.15 1.14 1.85 1.82
36 Kr 0.88 0.88 1.10 2.02
37 Rb 2.65 2.35 2.11 1.66
38 Sr 2.19 2.00 1.92 1.32
39 Y 2.12 1.85 1.62 1.04
40 Zr 2.06 1.55 1.48 0.86
41 Nb 1.98 1.45 1.37 0.78
42 Mo 1.90 1.45 1.45 0.79
43 Tc 1.83 1.35 1.56 0.79
44 Ru 1.78 1.30 1.26 0.82
45 Rh 1.73 1.35 1.35 0.81
46 Pd 1.69 1.40 1.31 1.63 0.78
47 Ag 1.65 1.60 1.53 1.72 1.29
48 Cd 1.61 1.55 1.48 1.58 0.92
49 In 1.56 1.55 1.44 1.93 0.94
50 Sn 1.45 1.45 1.41 2.17 0.69
51 Sb 1.33 1.45 1.38 0.90
52 Te 1.23 1.40 1.35 2.06 1.11
53 I 1.15 1.40 1.33 1.98 2.06
54 Xe 1.08 1.08 1.30 2.16 0.62
55 Cs 2.98 2.60 2.25 1.81
56 Ba 2.53 2.15 1.98 1.49
57 La 1.95 1.95 1.69 1.36
58 Ce 1.85 1.85 1.15
59 Pr 2.47 1.85 1.32
60 Nd 2.06 1.85 1.30
61 Pm 2.05 1.85 1.28
62 Sm 2.38 1.85 1.10
63 Eu 2.31 1.85 1.31
64 Gd 2.33 1.80 1.08
65 Tb 2.25 1.75 1.18
66 Dy 2.28 1.75 1.05
67 Ho 2.26 1.75 1.04
68 Er 2.26 1.75 1.03
69 Tm 2.22 1.75 1.02
70 Yb 2.22 1.75 1.13
71 Lu 2.17 1.75 1.60 1.00
72 Hf 2.08 1.55 1.50 0.85
73 Ta 2.00 1.45 1.38 0.78
74 W 1.93 1.35 1.46 0.74
75 Re 1.88 1.35 1.59 0.77
76 Os 1.85 1.30 1.28 0.77
77 Ir 1.80 1.35 1.37 0.77
78 Pt 1.77 1.35 1.28 1.75 0.74
79 Au 1.74 1.35 1.44 1.66 1.51
80 Hg 1.71 1.50 1.49 1.55 0.83
81 Tl 1.56 1.90 1.48 1.96 1.03
82 Pb 1.54 1.80 1.47 2.02 1.49
83 Bi 1.43 1.60 1.46 1.17
84 Po 1.35 1.90 1.08
85 At 1.27 1.27 0.76
86 Rn 1.20 1.20 1.45
87 Fr 1.94
88 Ra 2.15 1.62
89 Ac 1.95 1.95 1.26
90 Th 1.80 1.80 1.19
91 Pa 1.80 1.80 1.09
92 U 1.75 1.75 1.86 0.87
93 Np 1.75 1.75
94 Pu 1.75 1.75 1.00
95 Am 1.75 1.75 1.12
96 Cm 1.11