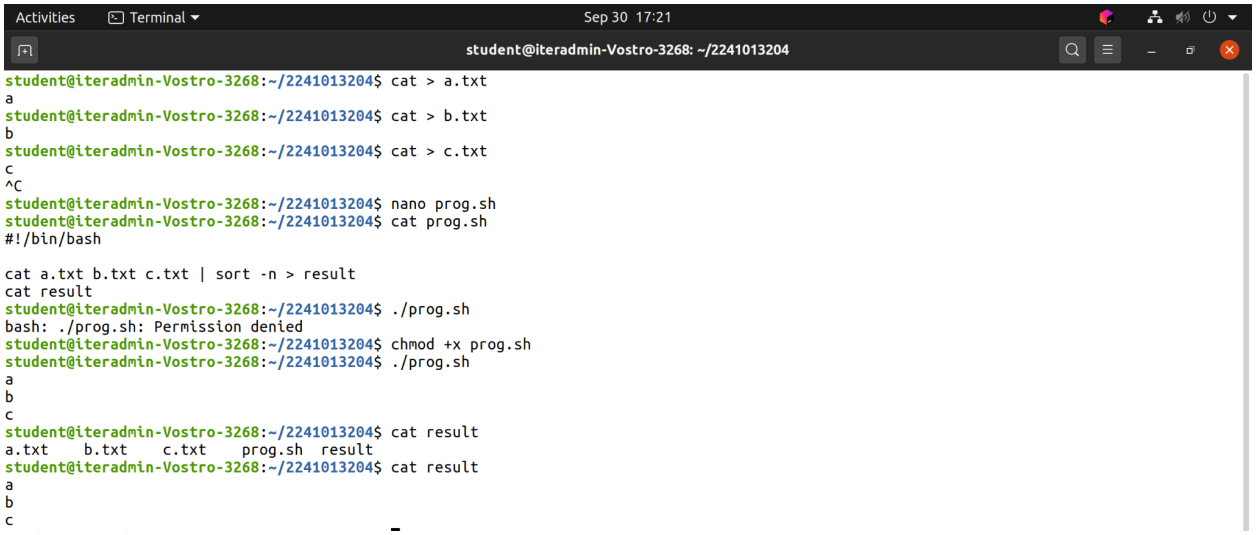
1. Write a shell script named as prog for merge the content of files a.txt, b.txt, and c.txt sort them and save the result in a file called result and display the sorted output on the screen.
(Note: a.txt, b.txt and c.txt file contain some numerical value. Make the script an executable file and run it as a command using its name only.)




```
student@iteradmin-Vostro-3268:~/2241013204$ cat > a.txt
a
student@iteradmin-Vostro-3268:~/2241013204$ cat > b.txt
b
student@iteradmin-Vostro-3268:~/2241013204$ cat > c.txt
c
^C
student@iteradmin-Vostro-3268:~/2241013204$ nano prog.sh
student@iteradmin-Vostro-3268:~/2241013204$ cat prog.sh
#!/bin/bash

cat a.txt b.txt c.txt | sort -n > result
cat result
student@iteradmin-Vostro-3268:~/2241013204$ ./prog.sh
bash: ./prog.sh: Permission denied
student@iteradmin-Vostro-3268:~/2241013204$ chmod +x prog.sh
student@iteradmin-Vostro-3268:~/2241013204$ ./prog.sh
a
b
c
student@iteradmin-Vostro-3268:~/2241013204$ cat result
a.txt    b.txt    c.txt    prog.sh  result
student@iteradmin-Vostro-3268:~/2241013204$ cat result
a
b
c
```

2. Write a shell script named as systeminfo that will display the information about the login name of the user, name of the Unix system used by the user, type of the SHELL, Path of current working directory of the user and list of file contain in current working directory.
(Make the script an executable file and run it as a command using its name only.)

```
student@iteradmin-Vostro-3268:~/2241013204$ nano systeminfo
student@iteradmin-Vostro-3268:~/2241013204$ cat systeminfo
#!/bin/bash

echo "Login Name: $USER"

echo "Unix System: $(uname -s)"

echo "Shell Type: $SHELL"

echo "Current Working Directory: $(pwd)"

echo "Files in Current Working Directory:"
ls -1
student@iteradmin-Vostro-3268:~/2241013204$ chmod +x systeminfo
student@iteradmin-Vostro-3268:~/2241013204$ ./systeminfo
Login Name: student
Unix System: Linux
Shell Type: /bin/bash
Current Working Directory: /home/student/2241013204
Files in Current Working Directory:
a.txt
b.txt
c.txt
dtcal
prog.sh
result
systeminfo
```

3. Write a shell script named as dtcal for displaying both the system date and calendar for specific month, say march 2022, in the given format:-
Date
:specific
date
Calender : current calendar
(Make the script an executable file and run it as a command using its name only.)

```
student@iteradmin-Vostro-3268:~/2241013204$ nano dtcal
student@iteradmin-Vostro-3268:~/2241013204$ cat dtcal
#!/bin/bash

current_date=$(date +"%Y-%m-%d")
echo "Date:"
echo ": $current_date"

echo "Calendar:"
cal 03 2022
student@iteradmin-Vostro-3268:~/2241013204$ chmod +x dtcal
student@iteradmin-Vostro-3268:~/2241013204$ ./dtcal
Date:
: 2024-09-30
Calendar:
     March 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

4. Write a shell script named as nvwc which will display the filename and linecount, wordcount and char count of the file dtcal in the following format:
Filename:
dtcal
Line count: -
Word count: -
Charcout: -
(Make the script an executable file and run it as a command using its name only.)

```
student@iteradmin-Vostro-3268:~/2241013204$ nano nvwc
student@iteradmin-Vostro-3268:~/2241013204$ cat nvwc
#!/bin/bash

filename="dtcal"

if [ ! -f "$filename" ]; then
    echo "File '$filename' does not exist."
    exit 1
fi

line_count=$(wc -l < "$filename")
word_count=$(wc -w < "$filename")
char_count=$(wc -c < "$filename")

echo "Filename:"
echo "$filename"
echo "Line count: $line_count"
echo "Word count: $word_count"
echo "Char count: $char_count"
student@iteradmin-Vostro-3268:~/2241013204$ chmod +x nvwc
student@iteradmin-Vostro-3268:~/2241013204$ ./nvwc
Filename:
dtcal
Line count: 8
Word count: 13
Char count: 112
```

5. Write a shell script named as nvwc2 which will display the filename and linecount, word count and char count of any file given as argument to nvwc2 in the following format:
filename linecount wordcount charcount
file1 - - -
(Make the script an executable file and run it as a command using its name only.)

```
student@iteradmin-Vostro-3268:~/2241013204$ nano nvwc2
student@iteradmin-Vostro-3268:~/2241013204$ chmod +x nvwc2
student@iteradmin-Vostro-3268:~/2241013204$ cat nvwc2
#!/bin/bash

if [ $# -ne 1 ]; then
    echo "Usage: $0 <filename>"
    exit 1
fi

filename="$1"

if [ ! -f "$filename" ]; then
    echo "File '$filename' does not exist."
    exit 1
fi

line_count=$(wc -l < "$filename")
word_count=$(wc -w < "$filename")
char_count=$(wc -c < "$filename")

echo "$filename $line_count $word_count $char_count"
student@iteradmin-Vostro-3268:~/2241013204$ ./nvwc2
Usage: ./nvwc2 <filename>
student@iteradmin-Vostro-3268:~/2241013204$ ./nvwc2 result
result 3 3 6
```

6. Write a shell script named as darg to display the total number of command line arguments along with the first two arguments.
-Modify the script to display all the arguments.
(Make the script an executable file and run it as a command using its name only.)

```
student@iteradmin-Vostro-3268:~/2241013204$ nano darg
student@iteradmin-Vostro-3268:~/2241013204$ cat darg
#!/bin/bash

# Display the total number of command line arguments
echo "Total number of arguments: $#"

# Display the first two arguments if they exist
echo -n "First argument: "
if [ $# -ge 1 ]; then
    echo "$1"
else
    echo "None"
fi

echo -n "Second argument: "
if [ $# -ge 2 ]; then
    echo "$2"
else
    echo "None"
fi

# Display all arguments
echo "All arguments: $@"
student@iteradmin-Vostro-3268:~/2241013204$ chmod +x darg
student@iteradmin-Vostro-3268:~/2241013204$ ./darg
Total number of arguments: 0
First argument: None
Second argument: None
All arguments:
student@iteradmin-Vostro-3268:~/2241013204$ ./darg "abcd" 1
Total number of arguments: 2
First argument: abcd
Second argument: 1
All arguments: abcd 1
```

7. Write a shell script named as ndisp that will take three command line arguments specifying the value of n, m and a filename and display the first n number of lines and last m number of lines of the file given as argument.
(Make the script an executable file and run it as a command using its name only.)

```
student@iteradmin-Vostro-3268:~/2241013204$ nano ndisp
student@iteradmin-Vostro-3268:~/2241013204$ cat ndisp
#!/bin/bash

# Check if exactly three arguments are provided
if [ $# -ne 3 ]; then
    echo "Usage: $0 <n> <m> <filename>"
    exit 1
fi

# Assign command line arguments to variables
n="$1"
m="$2"
filename="$3"

# Check if the specified file exists
if [ ! -f "$filename" ]; then
    echo "File '$filename' does not exist."
    exit 1
fi

# Display the first n lines
echo "First $n lines of '$filename':"
head -n "$n" "$filename"

# Display the last m lines
echo "Last $m lines of '$filename':"
tail -n "$m" "$filename"
student@iteradmin-Vostro-3268:~/2241013204$ chmod +x ndisp

student@iteradmin-Vostro-3268:~/2241013204$ cat > file.txt
Line 1
Line 2
Line 3
Line 4
Line 5
Line 6
Line 7
Line 8
Line 9
Line 10
student@iteradmin-Vostro-3268:~/2241013204$ ./ndisp 5 3 file.txt
First 5 lines of 'file.txt':
Line 1
Line 2
Line 3
Line 4
Line 5
Last 3 lines of 'file.txt':
Line 8
Line 9
Line 10
```