The main idea of this lab exercise to give hands on experience on -
grep
constructs
command line arguments

1. write a shell script to get the value of the pattern and file name from the user and check if the pattern exists or not. If the pattern exists print the relevant message , if pattern not found print relevant message.

```
  GNU nano 8.0                          script.sh
#!/bin/bash
echo "Enter the pattern to search for:"
read pattern

# Prompt the user for filenames
echo "Enter the filenames (space-separated if more than one):"
read -a filenames

# Check each file for the pattern
for file in "${filenames[@]}"; do
    if grep -q "$pattern" "$file"; then
        echo "Pattern '$pattern' found in file '$file'."
    else
        echo "Pattern '$pattern' not found in file '$file'."
    fi
done
```

```
┌──(surya_jjp㉿kali)-[~]
└─$ nano script.sh

┌──(surya_jjp㉿kali)-[~]
└─$ ./script.sh
Enter the pattern to search for:
Are
Enter the filenames (space-separated if more than one):
a1.txt poem.txt dump.txt
Pattern 'Are' not found in file 'a1.txt'.
Pattern 'Are' found in file 'poem.txt'.
Pattern 'Are' not found in file 'dump.txt'.
```

2. Modify the above script to pass the arguments from command line arguments.

```
#!/bin/bash

pattern=$1

shift
filenames=("$@")
# Check each file for the pattern
for file in "${filenames[@]}"; do
    if grep -q "$pattern" "$file"; then
        echo "Pattern '$pattern' found in file '$file'."
    else
        echo "Pattern '$pattern' not found in file '$file'."
    fi
done
```

```
┌──(surya_jjp㉿kali)-[~]
└─$ nano script.sh

┌──(surya_jjp㉿kali)-[~]
└─$ ./script.sh Are a1.txt.poem.txt dump.txt
grep: a1.txt.poem.txt: No such file or directory
Pattern 'Are' not found in file 'a1.txt.poem.txt'.
Pattern 'Are' not found in file 'dump.txt'.
```

3. Modify the above script to pass the values inside the script.

```
  GNU nano 8.0                                    script.
#!/bin/bash
pattern="Are"
filenames=("a1.txt" "poem.txt" "dump.txt")
# Check each file for the pattern
for file in "${filenames[@]}"; do
    if grep -q "$pattern" "$file"; then
        echo "Pattern '$pattern' found in file '$file'."
    else
        echo "Pattern '$pattern' not found in file '$file'."
    fi
done
```

```
┌──(surya_jjp㉿kali)-[~]
└─$ nano script.sh

┌──(surya_jjp㉿kali)-[~]
└─$ ./script.sh
Pattern 'Are' not found in file 'a1.txt'.
Pattern 'Are' found in file 'poem.txt'.
Pattern 'Are' not found in file 'dump.txt'.
```

4. validate the script (script 1, script 2)
    - the file exists or not
    - arguments passed or not

```
  GNU nano 8.0                                          script.sh
#!/bin/bash
if [ "$#" -lt 2 ]; then
    echo "Usage: $0 pattern file1 [file2 ... fileN]"
    exit 1
fi

# The first argument is the pattern
pattern=$1

# The rest of the arguments are filenames
shift
filenames=("$@")

# Validate if the files exist
for file in "${filenames[@]}"; do
    if [ ! -f "$file" ]; then
        echo "File '$file' does not exist. Exiting."
        exit 1
    fi
done
# Check each file for the pattern
for file in "${filenames[@]}"; do
    if grep -q "$pattern" "$file"; then
        echo "Pattern '$pattern' found in file '$file'."
    else
        echo "Pattern '$pattern' not found in file '$file'."
    fi
done
```

```
┌──(surya_jjp㉿kali)-[~]
└─$ nano script.sh

┌──(surya_jjp㉿kali)-[~]
└─$ ./script.sh Are a1.txt poem.txt dump.txt
Pattern 'Are' not found in file 'a1.txt'.
Pattern 'Are' found in file 'poem.txt'.
Pattern 'Are' not found in file 'dump.txt'.
```

5. Apply grep commands

Note: Make sure to use the options -e -c -n -q -s -f -A -B -C -i -h, -l -o -w

Frame the questions (as per your choice)

to extract user information

```
┌──(surya_jjp㉿kali)-[~]
└─$ grep -e 'surya_jjp' -n -c /etc/passwd
1
```

to extract network information

```
┌──(surya_jjp㉿kali)-[~]
└─$ grep -i -B 2 -A 2 'inet' /etc/network/interfaces
# The loopback network interface
auto lo
iface lo inet loopback
```

to extract login details

```
┌──(surya_jjp㉿kali)-[~]
└─$ grep -o -w 'login' /usr/share/man/man3/log.3.gz| wc -l
0
```

to exact the multiple words that matches the given pattern

```
┌──(surya_jjp㉿kali)-[~]
└─$ grep -w -e "I" -e "and" -i content.txt
I spent the winter my father died down in the basement,
on the table. And for months while the snow fell
and my father sat in the big chair by the Philco, dying,
I worked my way up deck by deck, story by story,
And there it loomed, a blazing city of the dead.
and placed my father at the railings, my mother
```

to count matching line

```
┌──(surya_jjp㉿kali)-[~]
└─$ grep -w -e "I" -e "and" -i content.txt | wc -l
6

┌──(surya_jjp㉿kali)-[~]
└─$ grep -c -w -e "I" -e "and" -i content.txt
6
```

Finding Words That Match Exactly

```
┌──(surya_jjp㉿kali)-[~]
└─$ grep -n -o "story" content.txt
8:story
8:story
```

Displaying File Names with a Specific Pattern

```
┌──(surya_jjp㉿kali)-[~]
└─$ grep -l "config" *
grep: Desktop: Is a directory
grep: Documents: Is a directory
grep: Downloads: Is a directory
grep: Music: Is a directory
grep: Pictures: Is a directory
grep: Public: Is a directory
grep: Templates: Is a directory
grep: Videos: Is a directory
grep: work: Is a directory
```

Displaying Context Around a Search Term

```
┌──(surya_jjp㉿kali)-[~]
└─$ grep -C 2 'ing' content.txt
on the table. And for months while the snow fell

and my father sat in the big chair by the Philco, dying,
I worked my way up deck by deck, story by story,

from steerage to first class, until at last it was done,
stacks, deck chairs, all the delicate rigging.

And there it loomed, a blazing city of the dead.
Then painted the gaping hole at the waterline

and placed my father at the railings, my mother
in a lifeboat pulling away from the wreckage.
```