Program 5a:

Develop an awk script that accepts date argument in the form of dd-mm-yy and displays it in the form of month, day and year. The script should should check the validity of the argument and in case of error, display a suitable message.

```
exam@ThinkCentre-M70t:~$ vi 5a.awk
```

```
{ split ( $0, arr, "-" )

if ((arr[1]<1) || (arr[1]>31) || (arr[2]<1) || (arr[2]>12))
{
print "invalid date"
exit 0
}
else {
if (arr[2]==1)
        print "Jan"
if (arr[2]==2)
        print "Feb"
if (arr[2]==3)
        print "March"
if (arr[2]==4)
        print "April"
if (arr[2]==5)
        print "May"
if (arr[2]==6)
        print "Jun"
if (arr[2]==7)
        print "Jul"
if (arr[2]==8)
        print "Aug"
if (arr[2]==9)
        print "Sep"
if (arr[2]==10)
        print "Oct"
if (arr[2]==11)
        print "Nov"
if (arr[2]==12)
        print "Dec"
print arr[1]
print arr[3]
}
}
```

Output:-

```
exam@ThinkCentre-M70t:~$ awk -f 5a.awk
12-04-2024
April
12
2024
```

Program 5b:

Develop an awk script to delete duplicated lines from a text file. The order of the original lines must remain unchanged.

```
exam@ThinkCentre-M70t:~$ vi 5b.awk
```

```
BEGIN {
print "Removing Duplicated lines"
}
{
  line [++no]=$0
  }
  END {
     for (i=1;i<=no;i++)
     {
             flag=1
             for (j=1;j<i;j++)
             if (line[i] == line[j])
             flag=0
             if(flag==1)
                     print line[i]>>"out.txt"
     }
     }
```

Output:-

Create a file with some contents to give as an argument while executing the awk script.

```
exam@ThinkCentre-M70t:~$ vi t1.txt
```

```
abcd
abcd
xyz
hello
Hai
Hai
```

Now execute the awk script, like shown below:

```
exam@ThinkCentre-M70t:~$ awk -f 5b.awk t1.txt
Removing Duplicated lines
exam@ThinkCentre-M70t:~$
```

The awk script removed the duplicated lines in t1.txt. Check the out.txt to see the unique lines.

```
exam@ThinkCentre-M70t:~$ cat out.txt
abcd
xyz
hello
Hai
```

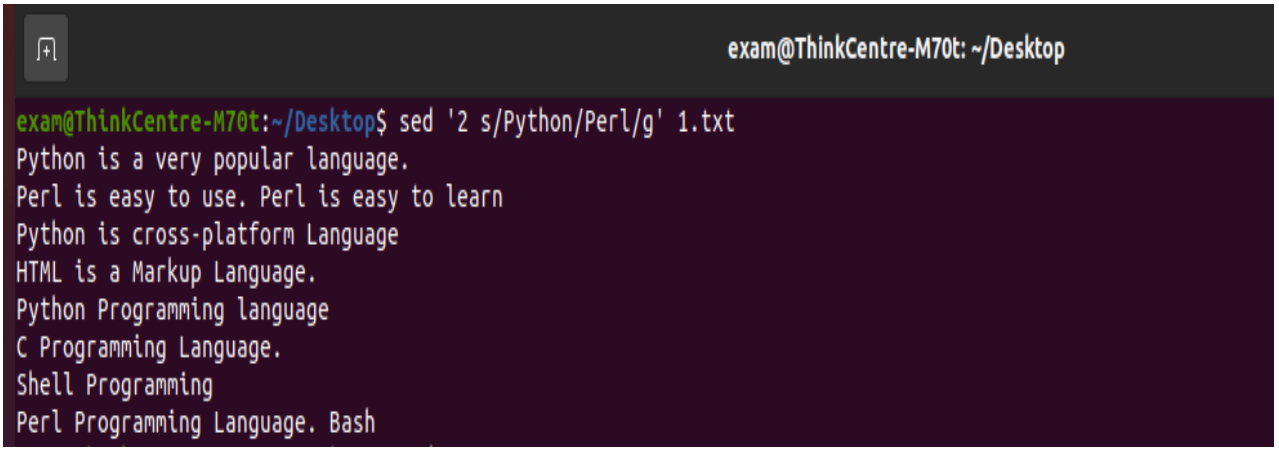
Program 6a:

Type the below text, Perform the following operations by using sed

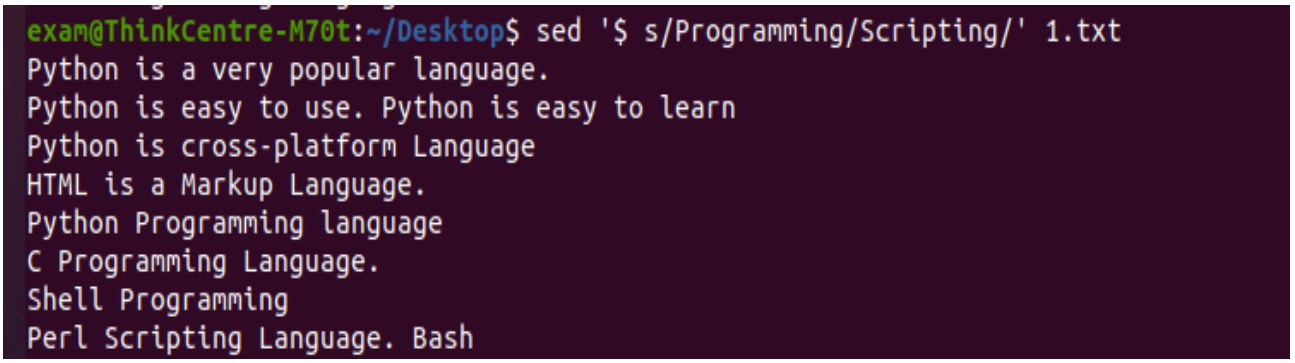
"Python is a very popular language.
Python is easy to use. Python is easy to learn.
Python is a cross -platform language.
HTML is a Markup Language.
Python Programming Language
C Programming Language
Shell Programming
Perl Programming language. Bash"

```
exam@ThinkCentre-M70t:~$ vi 1.txt
```

i) Replace all the instances of a Python in a second line of 1.txt with Perl.



ii) Replace the last occurrence of Programming with Scripting only of match, not other instances.



iii) Create a text file in the path /home/exam/Msrit/Python.txt. Replace the full path with just the filename no directory (such as Python.txt) and display it on standard output.

In /home/exam path create a Msrit directory and create a file named Python.txt inside it.

```
exam@ThinkCentre-M70t:~$ mkdir Msrit
exam@ThinkCentre-M70t:~$ cd Msrit
exam@ThinkCentre-M70t:~/Msrit$ echo "hai" > Python.txt
exam@ThinkCentre-M70t:~/Msrit$ ls
Python.txt
exam@ThinkCentre-M70t:~/Msrit$ pwd
/home/exam/Msrit
exam@ThinkCentre-M70t:~/Msrit$ echo "/home/exam/Msrit/Python.txt" | sed 's/.*\///'
Python.txt
```

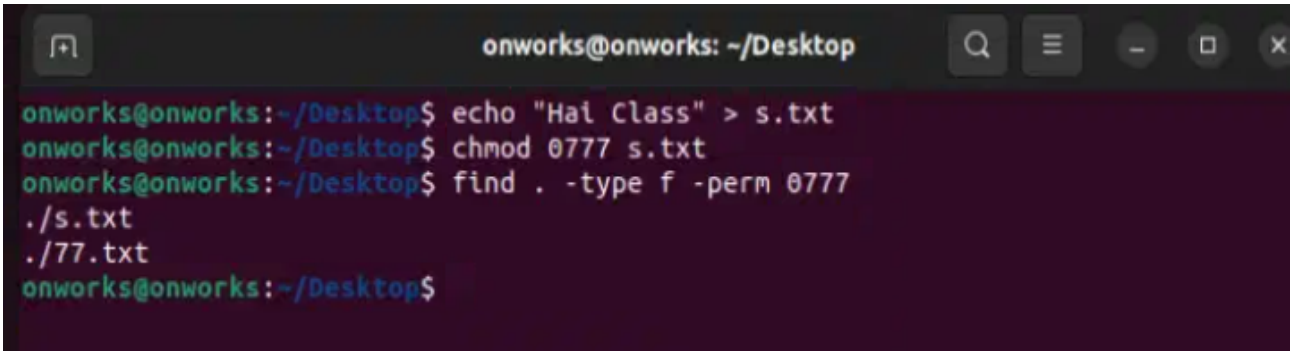
iv) Add string before and after the matching pattern using '\1'. In the above-given text, navigate yourself to the last line, you can find the *Bash* keyword, Add *Learn* before *Bash* and *Programming* after the *Bash* keyword.

```
exam@ThinkCentre-M70t:~/Desktop$ echo "Bash Language"|sed 's/\(Bash\)/ Learn\1 Programming/' 1.txt
Python is a very popular language.
Python is easy to use. Python is easy to learn
Python is cross-platform Language
HTML is a Markup Language.
Python Programming language
C Programming Language.
Shell Programming
Perl Programming Language.  LearnBash Programming
exam@ThinkCentre-M70t:~/Desktop$
```

6. b. Perform the following execution using find command

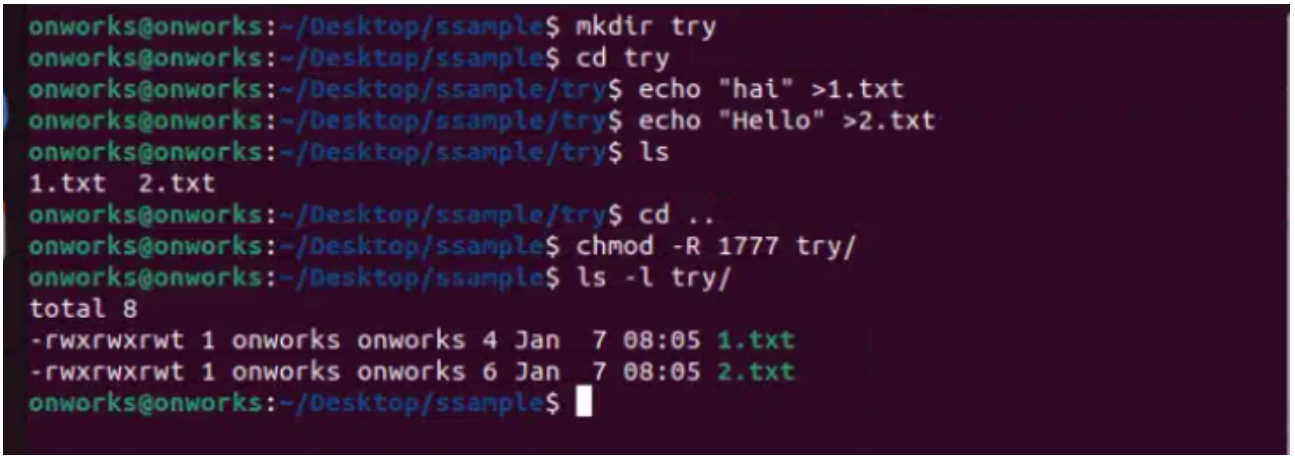
i) Find all the files in a current directory, whose permissions are 0777.

find . -type f -perm 0777



ii) Assign a sticky bit to all the files in a current directory.

chmod -R 1777 try/



Here in the directory try is created and there are two files 1.txt and 2.txt is created inside it. Then using the chmod command sticky bit permission is assigned to all files in a directory named try. Once the sticky bit is assigned only the owner of those can delete or rename those files and directory. Other users cannot delete or rename it because of a sticky bit.
A sticky bit is indicated with(t) in the permissions field.

iii) Find Directories with full permissions (777) and by using the chmod command change the permissions by assigning read, write and execute permissions to the owner and only read & and execute to the group and others.

```
onworks@onworks:~/Desktop$ find . -type d -perm 777|chmod -R 755 .
onworks@onworks:~/Desktop$ ls -l
total 16
-rwxr-xr-x 1 onworks onworks   14 Jan  7 07:51 77.txt
drwxr-xr-x 3 onworks onworks 4096 Jan  7 08:05 ssample
drwxr-xr-x 2 onworks onworks 4096 Jan  7 07:56 sticky
-rwxr-xr-x 1 onworks onworks   10 Jan  7 07:53 s.txt
onworks@onworks:~/Desktop$
```

iv) Find last 20 days modified files, accessed files.

```
onworks@onworks:~/Desktop$ find . -mtime -20
.
./2.txt
./1.txt
onworks@onworks:~/Desktop$ find . -atime -20
.
./2.txt
./1.txt
onworks@onworks:~/Desktop$
```

Here in the current directory, we are finding files modified and accessed in the last 20 days.

v) Find all the files that have been modified in the last 1 hour.

find . -mmin -60

```
onworks@onworks:~/Desktop$ vi 33.txt
onworks@onworks:~/Desktop$ find . -mmin -60
.
./1.txt
./33.txt
onworks@onworks:~/Desktop$
```