UNIX PROGRAMMING

1) A shell script to accept a filename as argument and displays the last modification time if the file exists and a suitable message if it does not.

```
kali@kali: ~/Desktop/1by19is047                                    _ □ ×

File  Actions  Edit  View  Help

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ cat 1
#!/bin/sh
echo "Enter the file name: "
read fileName
if [ -e $fileName ];
then
echo "File exist\n$fileName was last modified on $(stat -c '%y' $fileName| cut -c 1-20)"
else
echo "File does not exist"
fi

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ sh 1
Enter the file name:
samplefile1
File exist
samplefile1 was last modified on 2021-12-08 10:58:47.

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ sh 1
Enter the file name:
samplefile0
File does not exist

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ ▮
```

2) A shell script to accept 2 file names & check if the permission for these files are identical and if they are not identical, display each filename followed by permission.

```
kali@kali: ~/Desktop/1by19is047                                    _ □ ×

File  Actions  Edit  View  Help

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ cat 2
#!/bin/sh
echo "Enter 1st file name: "
read file1
echo "Enter 2nd file name: "
read file2
if [ -e $file1 ];
then
echo "$file1 exist"
else
echo "$file1 does not exist"
exit 0
fi
if [ -e $file2 ];
then
echo "$file2 exist"
else
echo "$file2 does not exist"
exit 0
fi
per1=`ls -l $file1 | cut -c 2-10`
per2=`ls -l $file2 | cut -c 2-10`
if [ $per1 = $per2 ];
then
echo "Permissions are Identical:permission is $per1"
else
echo "Permissions are not Identical"
echo "Permission of $file1 is $per1"
echo "Permission of $file2 is $per2"
fi

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ ▯
```

```
┌──(kali㊀kali)-[~/Desktop/1by19is047]
└─$ sh 2
Enter 1st file name:
samplefile1
Enter 2nd file name:
samplefile2
samplefile1 exist
samplefile2 exist
Permissions are Identical:permission is rw-r--r--

┌──(kali㊀kali)-[~/Desktop/1by19is047]
└─$ sh 2
Enter 1st file name:
samplefile0
Enter 2nd file name:
samplefile1
samplefile0 does not exist

┌──(kali㊀kali)-[~/Desktop/1by19is047]
└─$ sh 2
Enter 1st file name:
samplefile1
Enter 2nd file name:
samplefile0
samplefile1 exist
samplefile0 does not exist

┌──(kali㊀kali)-[~/Desktop/1by19is047]
└─$ chmod 777 samplefile1

┌──(kali㊀kali)-[~/Desktop/1by19is047]
└─$ sh 2
Enter 1st file name:
samplefile1
Enter 2nd file name:
samplefile2
samplefile1 exist
samplefile2 exist
Permissions are not Identical
Permission of samplefile1 is rwxrwxrwx
Permission of samplefile2 is rw-r--r--

┌──(kali㊀kali)-[~/Desktop/1by19is047]
└─$
```

3) A shell script to print first 10 numbers (1 to 10).

```
┌──(kali㊀kali)-[~/Desktop/1by19is047]
└─$ cat 3
#!/bin/sh
a=1
while [ $a -lt 11 ]
do
    echo $a
    a=`expr $a + 1`
done

┌──(kali㊀kali)-[~/Desktop/1by19is047]
└─$ sh 3
1
2
3
4
5
6
7
8
9
10

┌──(kali㊀kali)-[~/Desktop/1by19is047]
└─$
```

4) A shell script (program4.sh) to accept any number of arguments and print them in a reverse order. For example if A B C are entered then output is C B A.

```
kali@kali: ~/Desktop/1by19is047

File  Actions  Edit  View  Help

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ cat 4
#!/bin/sh
if [ $# -eq 0 ]
then
    echo "no arguments given"
    exit
fi
echo "Total number of arguments: $#"
echo "The arguments are: $*"
echo "The arguments in reverse order:"
rev=" "
for i in $*
do
  rev=$i" "$rev
done
echo $rev

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ sh 4
no arguments given

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ sh 4 A B C
Total number of arguments: 3
The arguments are: A B C
The arguments in reverse order:
C B A

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ ▯
```

5) A shell script to create a menu, which displays the list of files, process status, current date and current users of the system.

```
kali@kali: ~/Desktop/1by19is047

File  Actions  Edit  View  Help

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ cat 5
#!/bin/sh
while :
do
echo "\nEnter the options between 1 to 5\n"
echo "1 for list of files\n"
echo "2 for process status\n"
echo "3 for current date\n"
echo "4 for current users of the system\n"
echo "5 exit\n"
read choice
case $choice in
1) ls;;
2) ps;;
3) date;;
4) who;;
5) exit;;
*) echo "choose option between 1-4"
esac
done

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ sh 5

Enter the options between 1 to 5

1 for list of files

2 for process status

3 for current date

4 for current users of the system

5 exit

1
1  10  11  12  14  15  2  3  4  5  6  7  8  9  samplefile1  samplefile2
```

File Actions Edit View Help

```
Enter the options between 1 to 5

1 for list of files

2 for process status

3 for current date

4 for current users of the system

5 exit

2
    PID TTY          TIME CMD
   2821 pts/0    00:00:02 zsh
   3253 pts/0    00:00:00 sh
   3286 pts/0    00:00:00 ps
Enter the options between 1 to 5

1 for list of files

2 for process status

3 for current date

4 for current users of the system

5 exit

3
Thu 09 Dec 2021 07:37:39 AM EST
Enter the options between 1 to 5

1 for list of files

2 for process status

3 for current date

4 for current users of the system

5 exit

4
kali     tty7         2021-12-09 04:10 (:0)
```

File Actions Edit View Help

```
Enter the options between 1 to 5

1 for list of files

2 for process status

3 for current date

4 for current users of the system

5 exit

5
┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ 
```

6) A shell script to read a string from terminal and display suitable message if it doesn't have at least 10 characters using expr.

```
┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ cat 6
#!/bin/sh
echo "Enter a string : "
read str
len=`expr length "$str"`
if [ $len -gt 9 ];
then
echo "string has $len characters"
else
echo "string has only $len characters\nenter atleast 10 characters"
fi

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ sh 6
Enter a string :
hello world
string has 11 characters

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ sh 6
Enter a string :
hello
string has only 5 characters
enter atleast 10 characters

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$
```

7) A shell script to read a string from terminal and display suitable message if it doesn't have at least 10 characters using case.

```
┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ cat 7
#!/bin/sh
echo "Enter a string : "
read str
len=`expr length "$str"`
case $len in
1) echo "string contains only $len characters\nenter atleast 10";;
2) echo "string contains only $len characters\nenter atleast 10";;
3) echo "string contains only $len characters\nenter atleast 10";;
4) echo "string contains only $len characters\nenter atleast 10";;
5) echo "string contains only $len characters\nenter atleast 10";;
6) echo "string contains only $len characters\nenter atleast 10";;
7) echo "string contains only $len characters\nenter atleast 10";;
8) echo "string contains only $len characters\nenter atleast 10";;
9) echo "string contains only $len characters\nenter atleast 10";;
*) echo "string contains $len characters";;
esac

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ sh 7
Enter a string :
hello world
string contains 11 characters

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ sh 7
Enter a string :
hello
string contains only 5 characters
enter atleast 10

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ []
```

8) A shell script to check whether a given number is palindrome or not.

File Actions Edit View Help

```
┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ cat 8
#!/bin/sh
echo "Enter a number"
read num
rev=0
temp=$num
while [ $num -gt 0 ]
do
rev=$((rev*10))
rev=$((rev+$((num%10))))
num=$((num/10))
done
if [ $rev -eq $temp ];
then
echo "$temp is palandrome"
else
echo "$temp is not a palandrome"
fi

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ sh 8
Enter a number
121
121 is palandrome

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ sh 8
Enter a number
124
124 is not a palandrome

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ ▊
```

9) A shell script to read a pattern and filename from the terminal. And search for the pattern in the file.

File Actions Edit View Help

```
┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ cat 9
#!/bin/sh
echo "Enter pattern : "
read pattern
echo "Enter file name : "
read filename
a=$(grep -c $pattern $filename)
if [ $a -gt 0 ];
then
echo "$pattern found in $filename"
else
echo "$pattern not found in $filename"
fi

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ sh 9
Enter pattern :
hello
Enter file name :
samplefile1
hello found in samplefile1

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ sh 9
Enter pattern :
hi
Enter file name :
samplefile1
hi not found in samplefile1

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ ▊
```

10) A shell script (program10.sh) to compute sum of numbers passed in command line.

```
┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ cat 10
#!/bin/sh
echo "How many numbers would u like to add"
read n
i=0
sum=0
while [ $i -lt $n ]
do
echo "\nenter number : "
read num
sum=$((sum+num))
i=$((i+1))
done
echo "\nsum of numbers is $sum"

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ sh 10
How many numbers would u like to add
4

enter number :
1

enter number :
2

enter number :
3

enter number :
5

sum of numbers is 11

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ 
```

11) A shell script (program11.sh) to compute length of strings in the file (student.lst).

```
┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ cat 11
#!/bin/bash
echo "enter file name : "
read filename
for i in `cat $filename`
do
len=`expr length "$i"`
echo "string : $i, length : $len"
done

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ sh 11
enter file name :
samplefile1
string : hello, length : 5
string : world, length : 5

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ 
```

12) A shell script to validate the password. Let VALID_PASSWORD="secret";

```
┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ cat 12
#!/bin/sh
echo "Enter your password"
read str
if [ $str = "secret" ];
then
echo "vaild password"
else
echo "invalid password"
fi

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ sh 12
Enter your password
idontknow
invalid password

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ sh 12
Enter your password
secret
vaild password

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ ▯
```

13) A shell script to append doc extension to all filenames.

```
┌──(kali㉿kali)-[~/Desktop/1by19is047/samplefolder]
└─$ cat 13
#!/bin/sh
for i in *
do
if [ $i ≠ 13 ];
then
mv $i $i.doc
fi
done

┌──(kali㉿kali)-[~/Desktop/1by19is047/samplefolder]
└─$ ls
13  file1  samplefile3  Screenshot_2021-12-09_05-02-45.png  textfile

┌──(kali㉿kali)-[~/Desktop/1by19is047/samplefolder]
└─$ sh 13

┌──(kali㉿kali)-[~/Desktop/1by19is047/samplefolder]
└─$ ls
13  file1.doc  samplefile3.doc  Screenshot_2021-12-09_05-02-45.png.doc  textfile.doc

┌──(kali㉿kali)-[~/Desktop/1by19is047/samplefolder]
└─$ ▯
```

14) A shell script to check if the length of the name is greater than 20 characters.

```
┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ cat 14
#!/bin/sh
echo -n "Enter ur name : "
read name
len=`expr length "$name"`
if [ $len -gt 20 ];
then
echo "your name has ${#name} characters"
else
echo "your name has less than 20 characters"
fi

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ sh 14
Enter ur name : ram
your name has less than 20 characters

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ sh 14
Enter ur name : avul pakir jainlabdeen abdul kalam
your name has 34 characters

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$
```

15) Write a menu driven shell script to perform the following operations.

a) List of users   b) Files in a directory

c) Todays date   d) count number of files in a directory

```
┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ cat 15
#!/bin/sh
while :
do
echo "\nEnter the options between to execute commands:\n"
echo "1 for list of users\n"
echo "2 for files in current directory\n"
echo "3 for today's date\n"
echo "4 to count number of files in current directory\n"
echo "5 exit\n"
read choice
case $choice in
1) who;;
2) ls;;
3) date +"%d-%m-%y";;
4) ls | wc -l;;
5) exit;;
*) echo "choose option between 1-4"
esac
done

┌──(kali㉿kali)-[~/Desktop/1by19is047]
└─$ sh 15

Enter the options between to execute commands:

1 for list of users

2 for files in current directory

3 for today's date

4 to count number of files in current directory

5 exit

1
kali      tty7          2021-12-09 04:10 (:0)
```

```
Enter the options between to execute commands:

1 for list of users

2 for files in current directory

3 for today's date

4 to count number of files in current directory

5 exit

2
1  10  11  12  14  15  2  3  4  5  6  7  8  9  samplefile1  samplefile2
Enter the options between to execute commands:

1 for list of users

2 for files in current directory

3 for today's date

4 to count number of files in current directory

5 exit

3
09-12-21
Enter the options between to execute commands:

1 for list of users

2 for files in current directory

3 for today's date

4 to count number of files in current directory

5 exit

4
16
Enter the options between to execute commands:

1 for list of users

2 for files in current directory
```

```
3 for today's date

4 to count number of files in current directory

5 exit

5

┌──(kali㊙kali)-[~/Desktop/1by19is047]
└─$
```