

OS Lab Assignment 4 -Shell Script

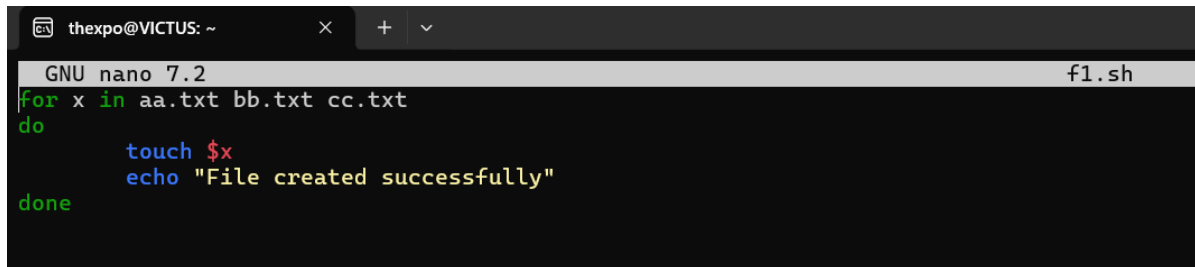
Name- Karnajeet Gosavi

Rollno- 71

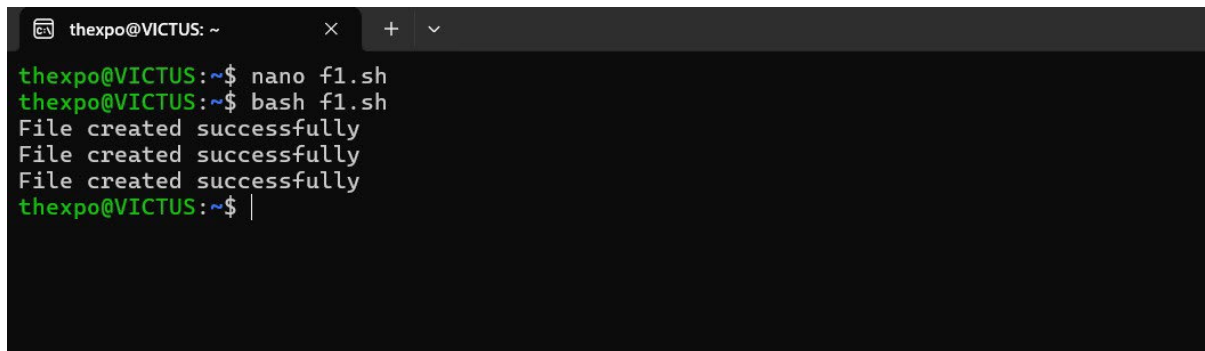
PRNno- 12311431

DIV- SY_CS_B

1. Write a shell script to create three files.

A terminal window titled 'thexpo@VICTUS: ~' with a tab for 'f1.sh'. The window shows the GNU nano 7.2 editor. The script content is:

```
for x in aa.txt bb.txt cc.txt
do
    touch $x
    echo "File created successfully"
done
```

A terminal window titled 'thexpo@VICTUS: ~' showing the execution of the script. The commands and output are:

```
thexpo@VICTUS:~$ nano f1.sh
thexpo@VICTUS:~$ bash f1.sh
File created successfully
File created successfully
File created successfully
thexpo@VICTUS:~$ |
```

2. Write a shell script to print date and time 10 times, once after each second.

A terminal window titled 'thexpo@VICTUS: ~' with a tab for 'f2.sh'. The window shows the GNU nano 7.2 editor. The script content is:

```
for((i = 0; i < 11; i++))
do
    date
    sleep 1
done
```

```
thexpo@VICTUS: ~  
thexpo@VICTUS:~$ nano f2.sh  
thexpo@VICTUS:~$ bash f2.sh  
Tue Feb 11 12:38:16 UTC 2025  
Tue Feb 11 12:38:17 UTC 2025  
Tue Feb 11 12:38:18 UTC 2025  
Tue Feb 11 12:38:19 UTC 2025  
Tue Feb 11 12:38:20 UTC 2025  
Tue Feb 11 12:38:21 UTC 2025  
Tue Feb 11 12:38:22 UTC 2025  
Tue Feb 11 12:38:23 UTC 2025  
Tue Feb 11 12:38:24 UTC 2025  
Tue Feb 11 12:38:25 UTC 2025  
Tue Feb 11 12:38:26 UTC 2025
```

3 Write a shell script to create five directories. Accept the name of directories from command line.

```
GNU nano 7.2 f3.sh  
for y in $1 $2 $3 $4 $5  
do  
    mkdir $y  
    echo "$y directory is created successfully"  
done
```

```
thexpo@VICTUS: ~  
thexpo@VICTUS:~$ nano f3.sh  
thexpo@VICTUS:~$ bash f3.sh game1 game2 game3 game4 game5  
game1 directory is created successfully  
game2 directory is created successfully  
game3 directory is created successfully  
game4 directory is created successfully  
game5 directory is created successfully  
thexpo@VICTUS:~$ |
```

4 Write a shell script to print long list of all file's names passed at command line.

```
GNU nano 7.2 f4.sh  
for z in $*  
do  
    ls -l $z  
done
```

```
thexpo@VICTUS: ~  
thexpo@VICTUS:~$ nano f4.sh  
thexpo@VICTUS:~$ bash f4.sh game88 game99 Student3 Student4  
-rw-r--r-- 1 thexpo thexpo 17 Feb 11 12:45 game88  
-rw-r--r-- 1 thexpo thexpo 17 Feb 11 12:46 game99  
-rw-r--r-- 1 thexpo thexpo 1280 Jan 30 13:16 Student3  
-rw-r--r-- 1 thexpo thexpo 1080 Jan 30 13:19 Student4
```

5 Write a shell script to count the number of file names passed at command line.

```
thexpo@VICTUS: ~  
GNU nano 7.2 f5.sh  
for x in $#  
do  
    echo "$x files were passed"  
done
```

```
thexpo@VICTUS: ~  
thexpo@VICTUS:~$ nano f5.sh  
thexpo@VICTUS:~$ bash f5.sh game1 game2 game3 game4 game88 game99  
6 files were passed
```

6 Write a shell script to accept and display array. Assume array consist 5 number.

```
thexpo@VICTUS: ~  
GNU nano 7.2 f6.sh  
echo "Enter array"  
for((i = 0; i < 5; i++))  
do  
    read a[i]  
done  
echo "Entered array"  
for((i = 0; i < 5; i++))  
do  
    echo "${a[i]}"  
done
```

```
thexpo@VICTUS: ~  
thexpo@VICTUS:~$ nano f6.sh  
thexpo@VICTUS:~$ bash f6.sh  
Enter array  
10  
20  
30  
40  
50  
Entered array  
10  
20  
30  
40  
50
```

7 Write Shell script to find out even and odd numbers from accepted array. Assume Array consists of 5 numbers. Also accept arrays from users.

```
thexpo@VICTUS: ~  
GNU nano 7.2 f7.sh  
echo "Enter array:"  
read -a nums  
  
even_nums=()  
odd_nums=()  
  
for num in "${nums[@]"; do  
    if (( num % 2 == 0 )); then  
        even_nums+=("$num")  
    else  
        odd_nums+=("$num")  
    fi  
done  
  
echo "Even numbers: ${even_nums[@]}"  
echo "Odd numbers: ${odd_nums[@]}"
```

```
thexpo@VICTUS: ~  
thexpo@VICTUS:~$ nano f7.sh  
thexpo@VICTUS:~$ bash f7.sh  
Enter array:  
10 11 20 13 50 17  
Even numbers: 10 20 50  
Odd numbers: 11 13 17
```

8 Write shell script to find out the reverse number of a given number.

```
thexpo@VICTUS: ~  
GNU nano 7.2 f8.sh  
echo "Enter a number:"  
read num  
  
num_str="$num"  
length=${#num_str}  
reverse=""  
  
for (( i=length-1; i>=0; i-- ))  
do  
    reverse="$reverse${num_str:$i:1}"  
done  
  
echo "Reversed number: $reverse"
```

```
thexpo@VICTUS: ~
thexpo@VICTUS:~$ nano f8.sh
thexpo@VICTUS:~$ bash f8.sh
Enter a number:
69
Reversed number: 96
thexpo@VICTUS:~$ bash f8.sh
Enter a number:
578
Reversed number: 875
```

9 Write Shell script to sort array numbers ascending and descending order. Assume Array consists of 5 numbers. Also accept arrays from users.

```
GNU nano 7.2 f9.sh
echo "Enter an array"
read -a numbers

for ((i = 0; i < 5; i++)); do
    for ((j = i + 1; j < 5; j++)); do
        if (( numbers[i] > numbers[j] )); then
            temp=${numbers[i]}
            numbers[i]=${numbers[j]}
            numbers[j]=$temp
        fi
    done
done

echo "Ascending order: ${numbers[@]}"

for ((i = 0; i < 5; i++)); do
    for ((j = i + 1; j < 5; j++)); do
        if (( numbers[i] < numbers[j] )); then
            temp=${numbers[i]}
            numbers[i]=${numbers[j]}
            numbers[j]=$temp
        fi
    done
done

echo "Descending order: ${numbers[@]}"
```

```
thexpo@VICTUS: ~
thexpo@VICTUS:~$ nano f9.sh
thexpo@VICTUS:~$ bash f9.sh
Enter an array
10 29 82 43 12
Ascending order: 10 12 29 43 82
Descending order: 82 43 29 12 10
```

10 Write Shell script to find out smallest number and largest number of given arrays. Assume Array consists of 5 numbers. Also accept arrays from users.

```
thexpo@VICTUS: ~  
GNU nano 7.2 f10.sh  
echo "Enter an array"  
read -a numbers  
  
for ((i = 0; i < 5; i++)); do  
    for ((j = i + 1; j < 5; j++)); do  
        if (( numbers[i] > numbers[j] )); then  
            temp=${numbers[i]}  
            numbers[i]=${numbers[j]}  
            numbers[j]=$temp  
        fi  
    done  
done  
  
echo "Smallest number is : ${numbers[1]}"  
echo "Largest number is : ${numbers[4]}"
```

```
thexpo@VICTUS: ~  
thexpo@VICTUS:~$ nano f10.sh  
thexpo@VICTUS:~$ bash f10.sh  
Enter an array  
23 48 35 93 52  
Smallest number is : 35  
Largest number is : 93
```