



---

# SHELL SCRIPTING

---

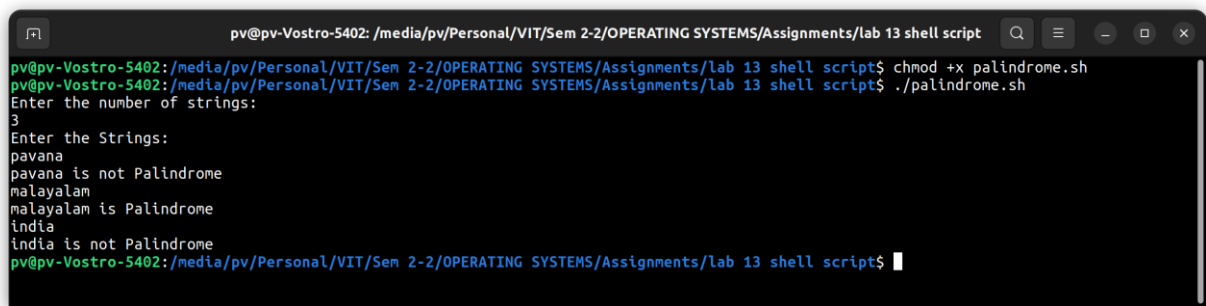
BCSE303P Operating Systems



*Name: Chillara V L N S Pavana Vamsi*  
*Register Number: 21BCE5095*

1. Read n strings and display the strings which are palindrome.

```
#!/bin/bash
echo "Enter the number of strings:"
read n
echo "Enter the Strings:"
for ((i=0;i<n;i++))
do
read string
reverse=$(echo $string | rev)
if [[ $string == $reverse ]]
then
echo "$string is Palindrome"
else
echo "$string is not Palindrome"
fi
done
```

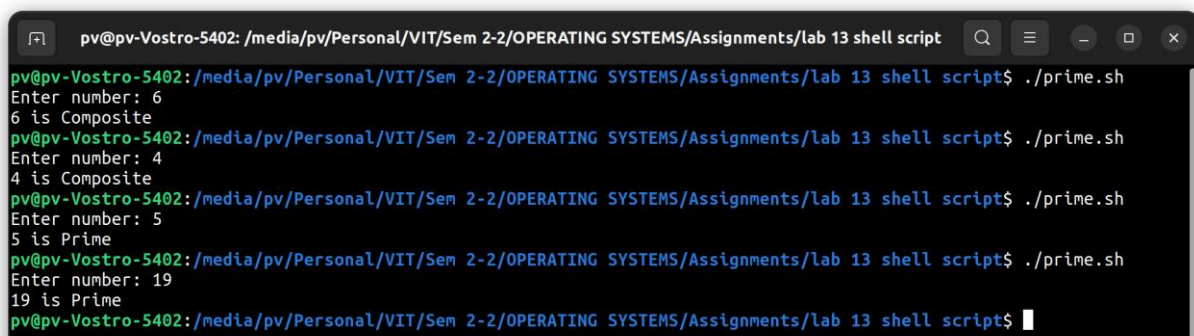


A terminal window titled "pv@pv-Vostro-5402: /media/pv/Personal/VIT/Sem 2-2/OPERATING SYSTEMS/Assignments/lab 13 shell script". The user runs "chmod +x palindrome.sh" and then "./palindrome.sh". The script prompts for the number of strings (3) and then for each string. It outputs "pavana is not Palindrome", "malayalam is Palindrome", and "india is not Palindrome".

```
pv@pv-Vostro-5402: /media/pv/Personal/VIT/Sem 2-2/OPERATING SYSTEMS/Assignments/lab 13 shell script$ chmod +x palindrome.sh
pv@pv-Vostro-5402: /media/pv/Personal/VIT/Sem 2-2/OPERATING SYSTEMS/Assignments/lab 13 shell script$ ./palindrome.sh
Enter the number of strings:
3
Enter the Strings:
pavana
pavana is not Palindrome
malayalam
malayalam is Palindrome
india
india is not Palindrome
pv@pv-Vostro-5402: /media/pv/Personal/VIT/Sem 2-2/OPERATING SYSTEMS/Assignments/lab 13 shell script$
```

2. Check the given number is prime or not.

```
#!/bin/bash
echo -e "Enter number: \c"
read n
for((i=2;i<n;i++))
do
if [ $((n % i)) -eq 0 ];
then
echo "$n is Composite"
exit 0
fi
done
echo "$n is Prime"
```

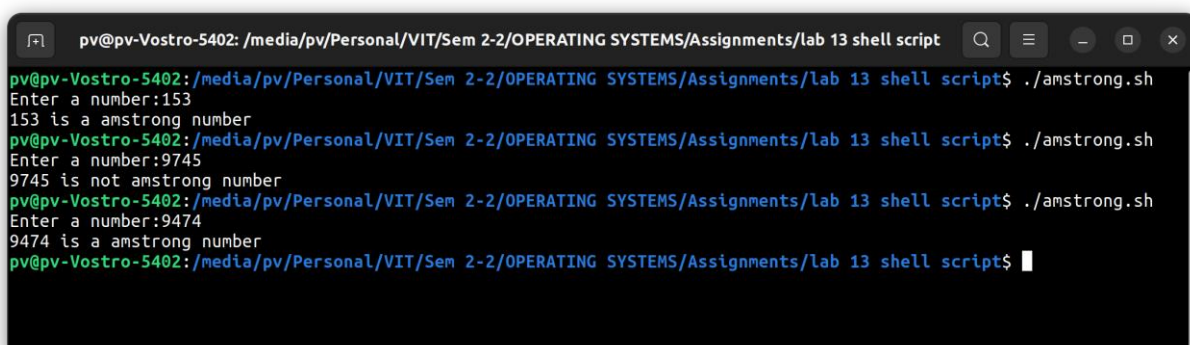


A terminal window titled "pv@pv-Vostro-5402: /media/pv/Personal/VIT/Sem 2-2/OPERATING SYSTEMS/Assignments/lab 13 shell script". The user runs "./prime.sh" four times with inputs 6, 4, 5, and 19. The script outputs "6 is Composite", "4 is Composite", "5 is Prime", and "19 is Prime".

```
pv@pv-Vostro-5402: /media/pv/Personal/VIT/Sem 2-2/OPERATING SYSTEMS/Assignments/lab 13 shell script$ ./prime.sh
Enter number: 6
6 is Composite
pv@pv-Vostro-5402: /media/pv/Personal/VIT/Sem 2-2/OPERATING SYSTEMS/Assignments/lab 13 shell script$ ./prime.sh
Enter number: 4
4 is Composite
pv@pv-Vostro-5402: /media/pv/Personal/VIT/Sem 2-2/OPERATING SYSTEMS/Assignments/lab 13 shell script$ ./prime.sh
Enter number: 5
5 is Prime
pv@pv-Vostro-5402: /media/pv/Personal/VIT/Sem 2-2/OPERATING SYSTEMS/Assignments/lab 13 shell script$ ./prime.sh
Enter number: 19
19 is Prime
pv@pv-Vostro-5402: /media/pv/Personal/VIT/Sem 2-2/OPERATING SYSTEMS/Assignments/lab 13 shell script$
```

3. Check the given number is Armstrong or not for a 4 digit number. (Ex. Armstrong numbers with 4 digits are 1634, 8208, and 9474)

```
#!/bin/bash
echo -e "Enter a number:\c"
read n
length=${#n}
sum=0
for ((i=0;i<length;i++))
do
digit=${n:$i:1}
sum=$((echo $sum + $digit^$length | bc))
done
if [ $sum -eq $n ];then
echo "$n is a armstrong number"
else
echo "$n is not armstrong number"
fi
```



The screenshot shows a terminal window titled "pv@pv-Vostro-5402: /media/pv/Personal/VIT/Sem 2-2/OPERATING SYSTEMS/Assignments/lab 13 shell script". The user runs the script `./amstrong.sh` three times. The first time, they enter 153, and the script outputs "153 is a armstrong number". The second time, they enter 9745, and the script outputs "9745 is not armstrong number". The third time, they enter 9474, and the script outputs "9474 is a armstrong number".

4. Display the factors of the given number

```
#!/bin/bash
echo -e "Enter the value of n:\c"
read n
for ((i=1;i<=n;i++))
do
if [ $((n%i)) -eq 0 ];then
echo -e "$i "
fi
done
```

```

pv@pv-Vostro-5402: /media/pv/Personal/VIT/Sem 2-2/OPERATING SYSTEMS/Assignments/lab 13 shell script
pv@pv-Vostro-5402:/media/pv/Personal/VIT/Sem 2-2/OPERATING SYSTEMS/Assignments/lab 13 shell script$ ./factors.sh
Enter the value of n:10
1
2
5
10
pv@pv-Vostro-5402:/media/pv/Personal/VIT/Sem 2-2/OPERATING SYSTEMS/Assignments/lab 13 shell script$ ./factors.sh
Enter the value of n:90
1
2
3
5
6
9
10
15
18
30
45
90
pv@pv-Vostro-5402:/media/pv/Personal/VIT/Sem 2-2/OPERATING SYSTEMS/Assignments/lab 13 shell script$ ./factors.sh
Enter the value of n:100
1
2
4
5
10
20
25
50
100
pv@pv-Vostro-5402:/media/pv/Personal/VIT/Sem 2-2/OPERATING SYSTEMS/Assignments/lab 13 shell script$

```

5. Assuming a 1 KB page size, what are the page numbers and offsets for the following logical addresses references. Develop a shell script for the same.

- 21942
- 5500850

```

echo -e "Enter the number of inputs:\c"
read n
echo -e "enter the page_size:\c"
read m
a=$((m*1024))
for ((i=0;i<n;i++))
do
echo -e "Enter the logical addr:\c"
read x
pgno=$((x/a))
offset=$((x%a))
echo "Page no:$pgno"
echo "Offset:$offset"
done

```

```

pv@pv-Vostro-5402: /media/pv/Personal/VIT/Sem 2-2/OPERATING SYSTEMS/Assignments/lab 13 shell script
pv@pv-Vostro-5402:/media/pv/Personal/VIT/Sem 2-2/OPERATING SYSTEMS/Assignments/lab 13 shell script$ gedit memory.sh
^C
pv@pv-Vostro-5402:/media/pv/Personal/VIT/Sem 2-2/OPERATING SYSTEMS/Assignments/lab 13 shell script$ chmod +x memory.sh
pv@pv-Vostro-5402:/media/pv/Personal/VIT/Sem 2-2/OPERATING SYSTEMS/Assignments/lab 13 shell script$ ./memory.sh
Enter the number of inputs:2
enter the page_size:1
Enter the logical addr:21942
Page no:21
Offset:438
Enter the logical addr:5500850
Page no:5371
Offset:946
pv@pv-Vostro-5402:/media/pv/Personal/VIT/Sem 2-2/OPERATING SYSTEMS/Assignments/lab 13 shell script$

```

6. Assume that a system has a 32 bit virtual address with a 4kb page. Write a C program that is passed a virtual address (in decimal number) on the command line and have it output the page number and offset for the given address. As an example your program should run as following

•./a.out 19986

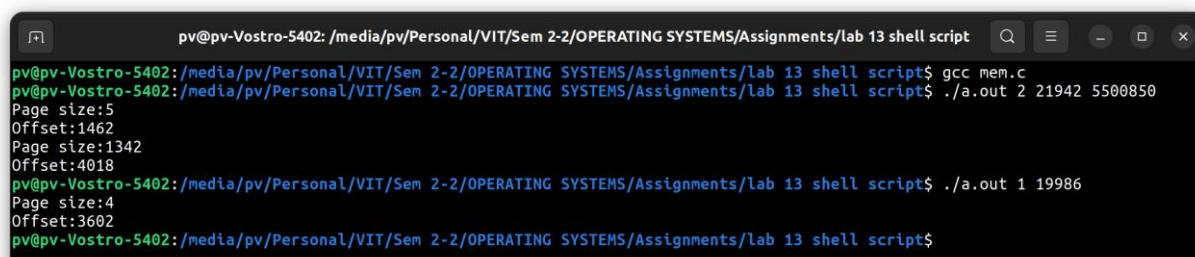
The program output should be

The address 19986 contains:

Page number = 4

Offset = 3602

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{
    int n, pgsz = 4, addr, pgno, offset;
    pgsz *= 1024;
    for (int i = 0; i < atoi(argv[1]); i++)
    {
        addr = atoi(argv[i + 2]);
        pgno = addr / pgsz;
        offset = addr % pgsz;
        printf("Page size:%d\nOffset:%d\n", pgno, offset);
    }
    return 0;
}
```



```
pv@pv-Vostro-5402: /media/pv/Personal/VIT/Sem 2-2/OPERATING SYSTEMS/Assignments/lab 13 shell script
pv@pv-Vostro-5402: /media/pv/Personal/VIT/Sem 2-2/OPERATING SYSTEMS/Assignments/lab 13 shell script$ gcc mem.c
pv@pv-Vostro-5402: /media/pv/Personal/VIT/Sem 2-2/OPERATING SYSTEMS/Assignments/lab 13 shell script$ ./a.out 2 21942 5500850
Page size:5
Offset:1462
pv@pv-Vostro-5402: /media/pv/Personal/VIT/Sem 2-2/OPERATING SYSTEMS/Assignments/lab 13 shell script$ ./a.out 1 19986
Page size:4
Offset:3602
pv@pv-Vostro-5402: /media/pv/Personal/VIT/Sem 2-2/OPERATING SYSTEMS/Assignments/lab 13 shell script$
```