```
#!/bin/bash
echo "Enter a number:"
read n
c=0
for (( i=2; i<=n/2; i++ ))
do
    if [ $((n%i)) -eq 0 ]
    then
        c=$((c+1))
        break
    fi
done
if [ $c -eq 0 ] && [ $n -gt 1 ]
then
    echo "Number is prime"
else
    echo "Number is not prime"
fi
output:
Enter a number:
2
Number is prime
```

```
#!/bin/bash
function digital_clock
{
clear
      while true
      do
      date +%T
      sleep 1
      clear
      done
}
digital_clock
output:
11:02:22
```

```
#!/bin/bash
echo "Enter Starting Number:"
read f
echo "Enter Ending Number:"
read e
for (( i=f;i<=e;i++ ))
do
c=0
      for (( j=2;j<=i/2;j++))
      do
      if [ $(($i%$j)) -eq 0 ]
      then
      c=$((c+1))
      break
      fi
      done
      if [ $c -eq 0 ] && [ $i -gt 1 ]
      then
      echo $i
      fi
done
output:
Enter Starting Number:
1
Enter Ending Number:
10

2
3
5
7
```

```
#!/bin/bash
echo "Enter First Number:"
read f
echo "Enter Second Number:"
read s
echo "Enter Third Number:"
read t
tot=`expr $f + $s + $t`
avg=`expr $tot / 3`
echo "average is : $avg"
output:
Enter First Number:
10
Enter Second Number:
20
Enter Third Number:
30
average is : 20
```

```
#!/bin/bash
for (( i=5;i>=1;i-- ))
do
      for (( j=5;j>=i;j-- ))
      do
      echo -ne $j
      done
      echo;
done
output:
5
54
543
5432
54321
```

```
#!/bin/bash
arr=(10 8 20 100 12)
echo "Array in original order:"
echo ${arr[*]}
for ((i=0;i<5;i++))
do
for ((j=0;j<5-i-1;j++))
do
      if [ ${arr[j]} -gt ${arr[ $((j+1)) ] } ]
      then
      temp=${arr[j]}
      arr[$j]=${arr[ $((j+1)) ] }
      arr[$((j+1))]=$temp
      fi
done
done
echo "Array in sorted order:"
echo ${arr[*]}
#ouput:
#Array in original order:
#10 8 20 100 12
#Array in sorted order:
#8 10 12 20 100
```

```
#!/bin/bash
echo "Enter Year To Check"
read y
if [ $((y%4)) -eq 0 ] && [ $((y%100)) -ne 0 ] || [ $((y%400)) -eq 0 ]
then
echo "Year is Leap"
else
echo "Year is not Leap"
fi
output:
Enter Year To Check
2024
Year is Leap
```

```
#!/bin/bash

echo "Enter directory path:"
read directory

# Check if directory exists
if [ -d "$directory" ]; then
    echo "Directory '$directory' exists."

    # Check if directory is empty
    if [ -z "$(ls -A $directory)" ]; then
        echo "Directory '$directory' is empty."
    else
        echo "Directory '$directory' is not empty."
    fi

    # Check if directory is a file
    if [ -f "$directory" ]; then
        echo "'$directory' is a file, not a directory."
    else
        echo "'$directory' is a directory."
    fi

else
    echo "Directory '$directory' does not exist."
fi
output:
Enter directory path:
/home/kali/impsh/test
Directory '/home/kali/impsh/test' exists.
Directory '/home/kali/impsh/test' is empty.
'/home/kali/impsh/test' is a directory.
```

```
#!/bin/bash

echo "Enter the number:"
read n

temp=$n
sum=0

while [ $n -gt 0 ]
do
    r=$((n % 10))
    sum=$((sum + r * r * r))
    n=$((n / 10))
done

if [ $temp -eq $sum ]; then
    echo "Armstrong number"
else
    echo "Not Armstrong number"
fi
output:
Enter the number:
153
Armstrong number
```

```
#!/bin/bash

echo "Enter the number:"
read n

temp=$n
reverse=0

while [ $n -gt 0 ]
do
    remainder=$((n % 10))
    reverse=$((reverse * 10 + remainder))
    n=$((n / 10))
done

if [ $temp -eq $reverse ]; then
    echo "Palindrome number"
else
    echo "Not palindrome"
fi
output:
Enter the number:
121
Palindrome number
```