30/3/22

1) Program to perform simple operations on if-else statement and while statement To check variables are same or not, greater variable, the no is even or odd.

```
#!/bin/bash
m=1
n=2
if [$n -eq $m]
then
    echo" Both variables are the same"
else
    echo Both variables are different"
fi
```

```
#!/bin/bash
a=2
b=7
if [$a -ge $b]
then
    echo" The variable 'a' is greater than variable
else
    echo" The variable 'b' is greater than variable
fi
```

```
#!bin/bash
n=10
if [$(n%2)) ==0]
then
    echo" The number is even"
else
```

```
    echo "The number is odd."
fi
```

while

```
# /bin/sh
  a=0
  while [$a - le 10]
  do
      echo $a
      a='expr $a +1'
  done
```

```
# ! /user/bin/bash

  a=7
  while [$a -gt 4];
  do
      echo $a
      ((a--))
  done
     echo "out of the loop"
```

```
# ! /user/bin/bash
  i=1
  while [$ si - le 4];
  do
       echo $i
       ((i++))
  done
```

if else statement

```
Syntax [Expression]
    then
        Statement 1
    else
        Statement 2
    fi
```

2) What are the looping statement in shell script explain?

1) while Statement

```
Syntax while [condition]
        do
          Statement 1
          Statement 2
          done
     ex. i=1
          while [$i -le 10]
          do
             echo $i
                i = exist $i+1'
          done
```

2) for loop

```
Syantax: for [variable-name] in
     do
        Statement 1
              Statement 2
              Statement 3
        done
```

ex:
```
for no in [1 10}
do
    echo $10
done.
```

Switch statement

Syntax:
```
Case in
    patern 1) Statement 1;;
    Pattern 2) statement 2;;
    pattern 3) statement 3;;
    pattern 4) statement 4;;
  esac
```

Ex: ~~Cake = "Audi"~~
~~Case "$ Car" in~~
~~# case 1"~~

ex:
```
# !/bin/sh
FRUIT= "kiwi"
Case " $ FRUIT" in
    "apple") echo " Apple pie is quite "
    ;;
    "banana") echo " I like banana nut berad."
    ;;
    "Kiwi") echo "New Zeeland is famous for
                     kiwi
    ;;
    esac.
```

Countinue statement : Continue statement skips the remaining commands inside the body of the enclosing loop for the current and passes program control

```
i=0
while ([ $i -lt 5 ]); do
  ((i++))
  if [[ "$i == '2' ]) ; then
    continue
  fi
    echo "Number: $i"
  done
     echo "All done'
```

output

```
Number: 1
Number: 5
Numbe: 4
Number: 5
All done.
```