① Explain conditional statement with examples.

Ans: Conditional statements are used to check a particular conditions and if true then the statement of that condition will be executed.

conditional operators:

-eq : equal to.

-lt : less than

-gt : greater than

== : ~~egua~~ two strings are equal.

!= : Not equal to.

~~} -&3~~

⇒) If statement:

"A condition is checked if true then the statement with then condition will be executed.

Syntax:
```
If [condition];
    commands/statements
fi
```

⇒ Ex:
```
echo "Enter a number"
read n
if [ $n -lt 100 ];
then
echo " $n is less than 100"
```

⇒ if-else:

· if condition is written if true then if block will be executed or else block will be performed.

Syntax:

```
if [condition];
then
    command(s)
else
    command(s)
fi
```

Ex:
```
a=10;
b=20;
if [a -gt b]
    echo "a is greater"
else
    echo "b is greater"
```

⇒ if - elif - else statements

· Multiple if statements are written and their conditions are checked and if any of them statement is false then the else block will be performed.

```
Syntax: if [condition]
        then
            commands
        elif [condition]
        then
            commands
        fi.
```

Ex: a=10;
b=20;
c=20;
if [a -gt b & a -gt c]
    then
        echo " a is greater"
    elif [b -gt a & b -gt c]
    then
        echo " b is greater"
    else
        echo "c is greater"
    fi

⇒ Nuted if:

• if condition are written inside another if condition.
    Syntax: if [condition] then
            commands
            if [condition] then
                commands
            fi
        fi.

⇒ Case statements

case statement can be used as an alternative to if statement.
    Syntax:
        case in
            pattern 1) commands;
            pattern n) commands;
        esac.

Ex: N1 = $1
    N2 = $2
    N3 = $3.

    case $op in '+')
    ((Punlt = $N1 + $N2));

    ((Punlt = $N1 - $N2));

    ((Punlt = $N1 * N2));

    ((Punlt = $N1 / $N2));

*)
    echo "wrong number of arguments"
    exit 0;

    esac

    echo "$N1 $op $N2 = $Punlt".

② Explain looping statement with example:
① while statement:
        until
• Here command is evaluated and based on the result loop will be executed. if condition is false then loop will be terminated.

Ex: a = 10
    until [$a -lt 10]
    do
        echo $a
        a = `expr $a + 1`
    done.

② For loop: while:

While loop enables you to execute set of commands repeatedly
until some condition occurs.

Ex: i=1
```
while [$i -le 10]
do
    echo $i
    i=`expr $i + 1`
done.
```

⑧ for loop:

A for loop is a statement which allows code to be repeatedly executed.

```
Ex: for no in {1.. 10}
do
    echo $no
done.
```