Practical No. 11: Execute Shell Script by using for statements.

X. Program Code :

1.

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ cat demo.sh

echo "Enter a Number"
read n
  for((i=1; i<=10; i++))
do
  echo " $n x $i = `expr $n \* $i` "
done

onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ bash demo.sh
Enter a Number
4
 4 x 1 = 4
 4 x 2 = 8
 4 x 3 = 12
 4 x 4 = 16
 4 x 5 = 20
 4 x 6 = 24
 4 x 7 = 28
 4 x 8 = 32
 4 x 9 = 36
 4 x 10 = 40
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$
```

2.

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ cat sample.sh

read n

for((i=n; i>=1; i--))
do
  for((j=1; j<=n-i; j++))
  do
   echo -n " " #one space
  done
   for((j=1; j<=i; j++))
   do
     echo -n " * " #one star one space
   done
echo
done
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ bash sample.sh
6
 *  *  *  *  *  *
  *  *  *  *  *
   *  *  *  *
    *  *  *
     *  *
      *
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$
```

XII. Practical Related Questions

1. a.

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ cat read.sh
#!/bin/sh
NUMBERS="1 2 3 4 5 6 7"
for NUM in $NUMS
do
  Q=`expr $NUM % 2`
  if [ $Q -eq 0 ]
  then
     echo "Number is an even number!!"
  continue
  fi
   echo "Found odd number"
done
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ bash read.sh
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$
```

b.

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ cat samp.sh
#!/bin/sh

a=0
while [ $a -lt 10 ]
do
  echo $a
     if [ $a -eq 5 ]
  then
  break
  fi
a=`expr $a + 1`
done
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ bash samp.sh
0
1
2
3
4
5
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$
```

2.

2. State the difference between Iteration and Recursion.

→

Iteration	Recursion
① Time complexity of iteration can be found by finding the number of cycles being repeated inside the loop.	Time complexity of recursion can be found by finding the value of the nth recursive call in terms of the previous calls.
② Iteration is repetition of a block code.	Recursion involves calling the same function again & it has a very small length of code.
③ Larger code size	Smaller code size.
④ For loops.	For functions.
⑤ A set of instructions repeatedly executes.	Function calls itself.

3.

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ cat demo.sh
a=0
b=1
echo "Enter a Number :"
read n
echo "Fibonacci series is :"
for((i=0; i<n; i++))
do
echo -n "$a"
echo " "
fn=$((a+b))
a=$b
b=$fn
done
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ bash demo.sh
Enter a Number :
5
Fibonacci series is :
0
1
1
2
3
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$
```

4.

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ cat table.sh

for((j=2; j<=10; j++))
do
  echo " "
  for((i=1; i<=10; i++))
  do
   echo "$j * $i = `expr $j \* $i` "
  done
done
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ bash table.sh

2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18
2 * 10 = 20
```

```
7 * 1 = 7
7 * 2 = 14
7 * 3 = 21
7 * 4 = 28
7 * 5 = 35
7 * 6 = 42
7 * 7 = 49
7 * 8 = 56
7 * 9 = 63
7 * 10 = 70

8 * 1 = 8
8 * 2 = 16
8 * 3 = 24
8 * 4 = 32
8 * 5 = 40
8 * 6 = 48
8 * 7 = 56
8 * 8 = 64
8 * 9 = 72
8 * 10 = 80
```

```
9 * 1 = 9
9 * 2 = 18
9 * 3 = 27
9 * 4 = 36
9 * 5 = 45
9 * 6 = 54
9 * 7 = 63
9 * 8 = 72
9 * 9 = 81
9 * 10 = 90

10 * 1 = 10
10 * 2 = 20
10 * 3 = 30
10 * 4 = 40
10 * 5 = 50
10 * 6 = 60
10 * 7 = 70
10 * 8 = 80
10 * 9 = 90
10 * 10 = 100
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$
```

5.

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ cat add.sh
num=12345
n=$num
s=0
while [ $num -gt 0 ]
do
k=$(( $num % 10 ))
num=$(( $num / 10 ))
s=$(( $s + $k ))
done
 echo "Addition of five digit numbers is : $s"
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ bash add.sh
Addition of five digit numbers is : 15
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$
```

XIII.

1.

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ cat week.sh

 i=1
for day in Mon Tue Wed Thu Fri Sat
 do
  echo "Weekday $((i++)) : $day"
done

onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ bash week.sh
Weekday 1 : Mon
Weekday 2 : Tue
Weekday 3 : Wed
Weekday 4 : Thu
Weekday 5 : Fri
Weekday 6 : Sat
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$
```

2.

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ cat pattern.sh
read n
for((i=n; i>=1; i--))
do
  for((j=1; j<=n-i; j++))
  do
  echo -n "  "
  done
   for((j=1; j<=i; j++))
   do
   echo -n "$"
   done
echo
done
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ bash pattern.sh
5
$$$$$
  $$$$
    $$$
      $$
        $
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$
```

3.

```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ cat operations.sh
 echo "Enter Two Numbers :"
read a
read b

echo "1.Addition"
echo "2.Subtraction"
echo "3.Multiplication"
echo "4.Division"
echo "Enter a Choice :"
read ch

case $ch in
  1)res=`expr $a + $b`;;
  2)res=`expr $a - $b`;;
  3)res=`expr $a \* $b`;;
  4)res=`expr $a / $b` ;;
esac
 echo "Result is : $res"
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ bash operations.sh
Enter Two Numbers :
10
20
1.Addition
2.Subtraction
3.Multiplication
4.Division
Enter a Choice :
3
Result is : 200
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$
```