# SHELL SCRIPTING

# Exercise 1 :Variables

```
#!/bin/sh
NAME="Anashka"                    # Assigning a string value to a
variable
echo $NAME                        # Displaying the value of the
variable
age=100                           # Assigning integer value
echo "Age is"$age
```

**Output:** Anashka

Age is 100

# Exercise 2 :Readonly variable

```
#!/bin/sh
NAME="Anashka"
readonly NAME                    #readonly variables cannot change
echo $NAME
#NAME="Anu"
unset NAME              # cannot unset a variable which is readonly
echo $NAME


echo $$                 # PID of the current shell
echo $0                 # file name of the current script
echo $n                 # arguement with which the script was invoked
```

**Output:**
```
Anashka
Anashka
17590
main.ksh

main.ksh[6]: unset: warning: NAME: is read only
```

# Exercise 3 : Array

```
#!/bin/sh
# Assigning values to array named DEPARTMENT

DEPARTMENT[0]="CSE"
DEPARTMENT[1]="ME"
DEPARTMENT[2]="ECE"

echo "The first dept is: ${DEPARTMENT[0]}"
# To get all elements of array
echo "Elements are : ${DEPARTMENT[*]}"

# To get all elements of array
echo "Elements are : ${DEPARTMENT[@]}"
```

**Output:** The first dept is : CSE
Elements are : CSE ME ECE
Elements are : CSE ME ECE

# Exercise 4 : Operators

```
#!/bin/sh
n1=4
n2=5

#Usage of Arithmetic operators

sum=`expr $n1 + $n2`
echo "sum : $sum"
pro=`expr $n1 \* $n2`
echo "product : $pro"
diff=`expr $n1 - $n2`
echo "difference : $diff"

# Usage of Relational operators and if..else..fi

if [ $n1 -gt $n2 ]
then
echo "n1 is bigger..."
else
echo "n2 is bigger..."
fi

if [ $n1 -lt $n2 ]
then
echo "n1 is small..."
else
```

```
echo "n2 is bigger..."
fi
```

**Output:**

```
sum : 9
product : 20
difference : -1
n2 is bigger...
n1 is small...
```

# Exercise 5: Decision making

```
#!/bin/sh

# Biggest among 3 numbers using if..else..fi statement

n1=3
n2=0
n3=1

if [ $n1 -gt $n2 ]
then
 if [ $n1 -gt $n3 ]
  then
   echo "biggest number is : $n1"
  else
   echo "biggest number is : $n3"
 fi
else
 if [ $n2 -gt $n3 ]
 then
   echo "biggest number is : $n2"
  else
   echo "biggest number is : $n3"
 fi
fi
```

**Output:** biggest number is : 3

# Exercise 6: Case statement

```
#!/bin/sh
w=6

case $w in

    1) echo "Monday"
    ;;
    2) echo "Tuesday"
    ;;
    3) echo "Wednesday"
    ;;
    4) echo "Thursday"
    ;;
    5) echo "Friday"
    ;;
    6) echo "Saturday"
    ;;
    7) echo "Sunday"
    ;;

esac
```

**Output:**

Saturday

# Exercise 7 : Loops

## a)While loop

### Exercise 1

```
#!/bin/sh

#Multiplication table using while loop

num=6
n=1
while [ $n -le 10 ]
do
   res=`expr $n \* $num`
   echo $n "*" $num "=" $res
   n=`expr $n + 1`
done
```

**Output:**

```
1 * 6 = 6
2 * 6 = 12
3 * 6 = 18
4 * 6 = 24
5 * 6 = 30
6 * 6 = 36
7 * 6 = 42
8 * 6 = 48
9 * 6 = 54
10 * 6 = 60
```

## b)for loops

### Exercise 1

```
#!/bin/sh
for i in {1..10}
do
    echo $i
done
```

**Output:**

```
1
2
3
4
5
6
7
8
9
10
```

## Exercise 2

```
#!/bin/sh
# checking prime number or not

num=13
count=0
for (( i=2 ; i<=`expr $num/2` ; i++ ))
do
    rem=`expr $num % $i`
    if [ $rem -eq 0 ]
    then
        count=`expr $count + 1`
        #echo $count
    fi

done
if [ $count -eq 0 ]
then
    echo $num" is prime"
else
    echo $num" is not prime"
fi
```

**Output:**

13 is prime

## c)until loop

### Exercise 1

```
#!/bin/sh
a=0
until [ $a -gt 10 ]
do
   echo "Welcome $a"
   $a=`expr $a + 1`
done
```

**Output:**

Welcome 0
Welcome 1
Welcome 2
Welcome 3
Welcome 4
Welcome 5
Welcome 6
Welcome 7
Welcome 8
Welcome 9

## d)select loop

### Exercise 1

```
#!/bin/bash
select chocolate in DiaryMilk Kitkat Perk
do
    echo $chocolate" selected"
done
```

**Output:**

1)DiaryMilk

2)Kitkat

3)Perk

```
#? 1
DiaryMilk selected
```

## e)Nested for loop

### Exercise 1

```
#!/bin/sh
for(( i=1 ; i <= 10; i++))
do
   for(( j=1; j<=$i ; j++))
   do
       echo -n "*"
   done

   echo
done
```

**Output:**

```
*
**
***
****
*****
******
*******
********
*********
**********
```

## f)Nested while loop

### Exercise 1

```
#!/bin/sh

i=1
while [ $i -le 10 ]
do
    j=1
    while [ $j -le $i ]
    do
        echo -n "*"
        j=`expr $j + 1`
    done
  echo ""
  i=`expr $i + 1`
  done
```

**Output:**

```
*
**
***
****
*****
******
*******
********
*********
**********
```

# Exercise 8 : Functions

## a)Creating Function

```
#!/bin/sh

# Function definition
wish ()
{
   echo "GOOD MORNING ALL...!"
}
#Function call
wish
```

**Output:**

GOOD MORNING ALL...!

a)Passing parameters

```
#!/bin/sh
# Function definition
wish ()
{
   echo "GOOD MORNING $1 $2"
}
#function call
wish abc def
```

**Output:**

```
GOOD MORNING abc def
```