ASSIGNMENT DAY5 PROBLEM 2 PRACTICE SETS

SELECTION PRACTICE PROBLEM WITH IF & ELSE

Problem 1: Write a program that reads 5 three digit values and then outputs the minimum and maximum value.

Solution: nano five1.sh

```
#!/bin/bash
max=0
min=1000
for (( i=1 ; $i<=5 ; i++ ))
do
        n=$((100+ $RANDOM%100))
        echo $n
        if [[ $max -lt $n ]]
        then
        max=$(($n))
        fi
        if [[ $min -gt $n ]]
        then
        min=$(($n))
        fi
done
echo "Maximum value: " $max
echo "Minimum value: " $min
```

Output : chmod +x five1.sh

```
Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five1.sh
187
167
136
106
149
Maximum value:  187
Minimum value:  106
```

Problem 2: Write the program that takes day and month from the command line and prints true if day of month is between March 20 and June 20 , false otherwise.

Solution: nano five2.sh

```
#!/bin/bash
echo "To display date between March 20 and June 20"
echo "Enter Date"
read date
echo "Enter Month(in words)"
read month
if [[ $month -eq "march" && $date -gt 20 && $date -lt 31 ]]
then
        echo "True"
elif [[ $month -eq "april" && $date -lt 30 ]]
then
        echo "True"
elif [[ $month -eq "may" && $date -lt 31 ]]
then
        echo "True"
elif [[ $month -eq "june" && $date -lt 20 ]]
then
        echo "True"
else
        echo "False"
```

Output : chmod +x five2.sh

```
Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five2.sh
To display date between March 20 and June 20
Enter Date
24
Enter Month(in words)
march
True

Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
```

```
$  ./five2.sh
To display date between March 20 and June 20
Enter Date
32
Enter Month(in words)
july
False
```

Problem 3] Write a program that takes a year as a input and outputs the year is a leap year or not a leap year. A Leap year checks for four digit number, divisible by 4 and not 100 unless divisible by 400.

Solution : nano five3.sh

```
#!/bin/bash
echo "Enter the year:"
read year
if [[ $year%400 -eq 0 || $year%4 -eq 0 && $year%100 -ne 0 ]]
then
        echo "It is a leap year"
else
        echo "It is not a leap year"
fi
```

Output : chmod +x five3.sh

```
Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five3.sh
Enter the year:
2020
It is a leap year

Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five3.sh
Enter the year:
2018
It is not a leap year
```

Problem 4] Write a program to silumate a coin flip and print out “Heads” or “Tails” accordingly.

Solution: nano five4.sh

```
#!/bin/bash
cointoss=$((RANDOM%2))
if [[ $cointoss -eq 1 ]]
then
        echo "HEADS"
else
        echo "TAILS"
fi
```

Output : chmod +x five4.sh

```
Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five4.sh
HEADS

Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five4.sh
HEADS

Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five4.sh
TAILS

Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five4.sh
TAILS

Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five4.sh
HEADS
```

SELECTION PRACTICE PROBLEMS WITH If, elif and else

Problem 5] Read a single digit number and write the number in word.

Solution: nano five5.sh

```
#!/bin/bash
echo "Enter single digit number: "
read a

if [[ $a -eq 0 ]]
then
        echo "0=zero"
elif [[ $a -eq 1 ]]
then
        echo "1=one"
elif [[ $a -eq 2 ]]
then
        echo "2=two"
elif [[ $a -eq 3 ]]
then
        echo "3=three"
elif [[ $a -eq 4 ]]
then
        echo "4=four"
elif [[ $a -eq 5 ]]
then
        echo "5=five"
elif [[ $a -eq 6 ]]
then
        echo "6=six"
elif [[ $a -eq 7 ]]
then
        echo "7=seven"
elif [[ $a -eq 8 ]]
then
        echo "8=eight"
elif [[ $a -eq 9 ]]
then
        echo "9=nine"
else
        echo "Number is not single digit number"
fi
```

Output : chmod five5.sh

```
Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five5.sh
Enter single digit number:
8
8=eight

Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five5.sh
Enter single digit number:
5
5=five

Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five5.sh
Enter single digit number:
10
Number is not single digit number
```

Problem 6] Read a number and display the week day.

Solution: nano five6.sh

```
#!/bin/bash
echo "Enter the number from 1 to 7: "
read a
if [[ $a -eq 1 ]]
then
        echo "$a=Sunday"
elif [[ $a -eq 2 ]]
then
        echo "$a=Monday"
elif [[ $a -eq 3 ]]
then
        echo "$a=Tuesday"
elif [[ $a -eq 4 ]]
then
```

```
        echo "$a=Wednesday"
elif [[ $a -eq 5 ]]
then
        echo "$a=Thursday"
elif [[ $a -eq 6 ]]
then
        echo "$a=Friday"
elif [[ $a -eq 7 ]]
then
        echo "$a=Saturday"
else
        echo "Invalid input"
fi
```

Output: chmod +x five6.sh

```
Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five6.sh
Enter the number from 1 to 7:
1
1=Sunday

Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five6.sh
Enter the number from 1 to 7:
2
2=Monday

Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five6.sh
Enter the number from 1 to 7:
3
3=Tuesday

Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five6.sh
Enter the number from 1 to 7:
4
4=Wednesday

Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five6.sh
Enter the number from 1 to 7:
5
5=Thursday

Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five6.sh
Enter the number from 1 to 7:
6
6=Friday

Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five6.sh
Enter the number from 1 to 7:
7
7=Saturday

Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five6.sh
Enter the number from 1 to 7:
9
Invalid input
```

Problem 7] Read a number 1, 10, 100, 1000 etc and display unit, tens, hundred, thousands, etc.

Solution : nano five7.sh

```
#!/bin/bash
echo "Enter the number containing the place value: "
read a
if [[ $a -eq 1 ]]
then
        echo "Its unit place"
elif [[ $a -eq 10 ]]
then
        echo "Its tens place"
elif [[ $a -eq 100 ]]
then
        echo "Its hundred place"
elif [[ $a -eq 1000 ]]
then
```

```
        echo "Its thousand place"
else
        echo "Provide valid input"
fi
```

Output : chmod +x five7.sh

```
Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five7.sh
Enter the number containing the place value:
100
Its hundred place

Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five7.sh
Enter the number containing the place value:
10
Its tens place

Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five7.sh
Enter the number containing the place value:
1
Its unit place

Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five7.sh
Enter the number containing the place value:
1000
Its thousand place

Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five7.sh
Enter the number containing the place value:
10000
Provide valid input
```

Problem 8] Enter three numbers, do following arithmetic operation and find the one that is maximum and minimum.

i) a+b*c ii) a%b+c iii)c+a/b iv)a*b+c

Solution: nano five8.sh

```
#!/bin/bash
echo "a="
read a
echo "b="
read b
echo "c="
read c

d=$(($a+$b*$c))
echo "a + b * c:" $d
e=$(($a%$b+$c))
echo "a % b + c" $e
f=$(($c+$a/$b))
echo "c + a / b " $f
g=$(($a*$b+$c))
echo "a * b + c" $g

echo "Maximum: " $d
echo "Minimum: " $f
```

Output: chmod +x five8.sh

```
Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five8.sh
a=
15
b=
78
c=
36
a + b * c: 2823
a % b + c 51
c + a / b  36
a * b + c 1206
```

```
Maximum:  2823
Minimum:  36
```

SELECTION PRACTICE PROBLEM WITH CASE STATEMENT

Problem 9] Read a single digit number and write the number in word using case statement.

Solution: nano five9.sh

```
#!/bin/bash

echo "Enter single digit number:"
read num

case "$num" in
        "0") echo "zero"
        ;;
        "1") echo "one"
        ;;
        "2") echo "two"
        ;;
        "3") echo "three"
        ;;
        "4") echo "four"
        ;;
        "5") echo "five"
        ;;
        "6") echo "six"
        ;;
        "7") echo "seven"
        ;;
        "8") echo "eight"
        ;;
        "9") echo "nine"
        ;;
        *)
        echo "Not a single digit number"
esac
```

Output: chmod +x five9.sh

```
Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five9.sh
Enter single digit number:
5
five

Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five9.sh
Enter single digit number:
8
eight

Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five9.sh
Enter single digit number:
1
one

Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five9.sh
Enter single digit number:
6
six

Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five9.sh
Enter single digit number:
45
Not a single digit number
```

Problem 10] Read a number and display the week day.

Solution : nano five10.sh

```
#!/bin/bash
```

```
echo "Enter the numbers between 1 to 7 to get the week days:"
read n
case $n in
        "1") echo "Sunday"
        ;;
        "2") echo "Monday"
        ;;
        "3") echo "Tuesday"
        ;;
        "4") echo "Wednesday"
        ;;
        "5") echo "Thursday"
        ;;
        "6") echo "Friday"
        ;;
        "7") echo "Saturday"
        ;;
        *) echo "Invalid input"
esac
```

Output: chmod +x five10.sh

```
Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five10.sh
Enter the numbers between 1 to 7 to get the week days:
1
Monday

Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five10.sh
Enter the numbers between 1 to 7 to get the week days:
2
Tuesday

Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five10.sh
Enter the numbers between 1 to 7 to get the week days:
3
Wednesday

Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five10.sh
Enter the numbers between 1 to 7 to get the week days:
4
Thursday

Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five10.sh
Enter the numbers between 1 to 7 to get the week days:
5
Friday

Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five10.sh
Enter the numbers between 1 to 7 to get the week days:
6
Saturday

Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five10.sh
Enter the numbers between 1 to 7 to get the week days:
7
Sunday

Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five10.sh
Enter the numbers between 1 to 7 to get the week days:
8
Invalid input
```

Problem 11] Read a number 1,10,100,1000 ,etc and display them as unit, tens, hundred, thousand, etc.

Solution: nano five11.sh

```
#!/bin/bash
echo "Enter the numbers representing the place values:"
read n

case $n in
        "1") echo "unit place"
        ;;
```

```
        "10") echo "tens place"
        ;;
        "100") echo "hundred place"
        ;;
        "1000") echo "thousand place"
        ;;
        "10000") echo "ten thousand place"
        ;;
        *) echo "Please provide the valid input"
esac
```

Output: chmod +x five11.sh

```
Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five11.sh
Enter the numbers representing the place values:
10
tens place

Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five11.sh
Enter the numbers representing the place values:
1
unit place

Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five11.sh
Enter the numbers representing the place values:
100
hundred place

Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five11.sh
Enter the numbers representing the place values:
1000
thousand place

Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five11.sh
Enter the numbers representing the place values:
1000000
Please provide the valid input
```

Problem 12] Write a program that takes User inputs and does unit Conversion of different length units.

1. Feet to inch **3. Inch to feet**
2. Feet to meter **4. Meter to feet**

Solution : nano five12.sh

```
#!/bin/bash
echo "Select the conversion you want to do: 1] Feet to inch, 2] Feet to meter , 3] Inch to feet
, 4] Meter to feet"
read n

case $n in
    "1") read -p "Enter the length:" length
        inch=$(( $length * 12 ))
        echo "$length feet equal to $inch inch"
        ;;
    "2") read -p "Enter the value :" feet
        meter=`(awk "BEGIN{print $feet/3.281}")`
        echo "$feet feet equal to $meter meter"
        ;;
    "3") read -p "Enter the value :" inch
        feet=`(awk "BEGIN{print $inch/12}")`
        echo "$inch inch equal to $feet feet"
        ;;
    "4") read -p "Enter the value :" meter
        feet=`(awk "BEGIN{print $meter*3.281}")`
        echo "$meter meter equal to $feet feet"
        ;;
        *) echo "Invalid input"
esac
```

Output : 1. Feet to inch

```
Hp@DESKTOP-OAFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five12.sh
Select the conversion you want to do: 1] Feet to inch, 2] Feet to meter , 3] Inch to feet , 4]
Meter to feet
1
Enter the length:2
2 feet equal to 24 inch
```

2.feet to meter :

```
Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five12.sh
Select the conversion you want to do: 1] Feet to inch, 2] Feet to meter , 3] Inch to feet , 4]
Meter to feet
2
Enter the value :12
   12  eet equal to 3.65742 meter
```

3.Inch to feet :

```
Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five12.sh
Select the conversion you want to do: 1] Feet to inch, 2] Feet to meter , 3] Inch to feet , 4]
Meter to feet
3
Enter the value :14
14 inch equal to 1.16667 feet
```

4.meter to feet :

```
Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five12.sh
Select the conversion you want to do: 1] Feet to inch, 2] Feet to meter , 3] Inch to feet , 4]
Meter to feet
4
Enter the value :40
40 meter equal to 131.24 feet

Hp@DESKTOP-0AFPT6H MINGW64 ~/Desktop/bridgelabz
$  ./five12.sh
Select the conversion you want to do: 1] Feet to inch, 2] Feet to meter , 3] Inch to feet , 4]
Meter to feet
5
Invalid input
```