**Stage 2\_Assignment\_1**

Sequences Practice Problems

1. Use Random Function (( RANDOM )) to get Single Digit

**Code**

#! /bin/bash -x

x=$RANDOM

echo $x

**Output**

nida@nida-PC MINGW64 ~/stage2/shellcommands/seq

$ ./q1.sh

+ x=8110

+ echo 8110

8110

nida@nida-PC MINGW64 ~/stage2/shellcommands/seq

$ ./q1.sh

+ x=15759

+ echo 15759

15759

1. Use Random to get Dice Number between 1 to 6

**Code**

#! /bin/bash -x

x=$RANDOM

echo $(( (x % 6) + 1 ))

**Output**

nida@nida-PC MINGW64 ~/stage2/shellcommands/seq

$ ./q2.sh

+ x=7883

+ echo 6

6

nida@nida-PC MINGW64 ~/stage2/shellcommands/seq

$ ./q2.sh

+ x=5460

+ echo 1

1

1. Add two Random Dice Number and Print the Result

**Code**

#! /bin/bash -x

x1=$RANDOM

x2=$RANDOM

dice1=$((x1 % 6 + 1 ))

dice2=$((x2 % 6 + 1 ))

sum=`expr $dice1 + $dice2`

echo $sum

**Output**

nida@nida-PC MINGW64 ~/stage2/shellcommands/seq

$ ./q2.sh

+ x1=29917

+ x2=23176

+ dice1=2

+ dice2=5

++ expr 2 + 5

+ sum=7

+ echo 7

7

1. Write a program that reads 5 Random 2 Digit values , then find their sum and the average

**Code**

#! /bin/bash -x

sum=0

for i in `seq 5`

do

n=$((RANDOM % 90 + 10 ))

sum=`expr $sum + $n`

done

avg=`expr $sum / 5`

echo "Sum = " $sum

echo "Average = " $avg

**Output**

nida@nida-PC MINGW64 ~/stage2/shellcommands/seq

$ ./q4.sh

+ sum=0

++ seq 5

+ for i in `seq 5`

+ n=95

++ expr 0 + 95

+ sum=95

+ for i in `seq 5`

+ n=51

++ expr 95 + 51

+ sum=146

+ for i in `seq 5`

+ n=2

++ expr 146 + 2

+ sum=148

+ for i in `seq 5`

+ n=22

++ expr 148 + 22

+ sum=170

+ for i in `seq 5`

+ n=52

++ expr 170 + 52

+ sum=222

++ expr 222 / 5

+ avg=44

+ echo 'Sum = ' 222

Sum = 222

+ echo 'Average = ' 44

Average = 44

5. Unit Conversion

a. 1ft = 12 in then 42 in = ? ft

b. Rectangular Plot of 60 feet x 40 feet in meters

c. Calculate area of 25 such plots in acres

**Code:**

#! /bin/bash -x

feet=$(awk "BEGIN{print 42 / 12}")

echo "42 inches = "$feet "Feet"

Rect\_plot=$(awk "BEGIN{print ( 60 \* 40) / 3.281}")

echo "Rectangular plot inches of 60x40 = " $Rect\_plot "Meter"

acre=$(awk "BEGIN{print ($Rect\_plot \* 25) / 4047}")

echo "Area in acres= "$acre

**Output:**

nida@nida-PC MINGW64 ~/stage2/shellcommands/selection

$ ./Q7.sh

++ awk 'BEGIN{print 42 / 12}'

+ feet=3.5

+ echo '42 inches = 3.5' Feet

42 inches = 3.5 Feet

++ awk 'BEGIN{print ( 60 \* 40) / 3.281}'

+ Rect\_plot=731.484

+ echo 'Rectangular plot inches of 60x40 = ' 731.484 Meter

Rectangular plot inches of 60x40 = 731.484 Meter

++ awk 'BEGIN{print (731.484 \* 25) / 4047}'

+ acre=4.51868

+ echo 'Area in acres= 4.51868'

Area in acres= 4.51868

Selection Practice Problems with if & else

1. Write a program that reads 5 Random 3 Digit values and then outputs the minimum and the maximum value

**Code:**

#! /bin/bash -x

n1=$(( RANDOM % 900 + 100 ))

n2=$(( RANDOM % 900 + 100 ))

n3=$(( RANDOM % 900 + 100 ))

n4=$(( RANDOM % 900 + 100 ))

n5=$(( RANDOM % 900 + 100 ))

if [[ $n1 -gt $n2 && $n1 -gt $n3 && $n1 -gt $n3 && $n1 -gt $n4 && $n1 -gt $n5 ]]

then

echo $n1 "is max"

else

if [[ $n2 -gt $n3 && $n2 -gt $n3 && $n2 -gt $n4 && $n2 -gt $n5 ]]

then

echo $n2 "is max"

else

if [[ $n3 -gt $n4 && $n2 -gt $n5 ]]

then

echo $n3 "is max"

else

if [ $n4 -gt $n5 ]

then

echo $n4 "is max"

else

echo $n5 "is max"

fi

fi

fi

fi

**Output:**

nida@nida-PC MINGW64 ~/stage2/shellcommands/selection

$ ./Q5.sh

+ n1=780

+ n2=345

+ n3=675

+ n4=503

+ n5=874

+ [[ 780 -gt 345 ]]

+ [[ 780 -gt 675 ]]

+ [[ 780 -gt 675 ]]

+ [[ 780 -gt 503 ]]

+ [[ 780 -gt 874 ]]

+ [[ 345 -gt 675 ]]

+ [[ 675 -gt 503 ]]

+ [[ 345 -gt 874 ]]

+ '[' 503 -gt 874 ']'

+ echo 874 'is max'

874 is max

For Minimum:

Code:

#! /bin/bash -x

n1=$(( RANDOM % 900 + 100 ))

n2=$(( RANDOM % 900 + 100 ))

n3=$(( RANDOM % 900 + 100 ))

n4=$(( RANDOM % 900 + 100 ))

n5=$(( RANDOM % 900 + 100 ))

if [[ $n1 -lt $n2 && $n1 -lt $n3 && $n1 -lt $n3 && $n1 -lt $n4 && $n1 -lt $n5 ]]

then

echo $n1 "is min"

else

if [[ $n2 -lt $n3 && $n2 -lt $n3 && $n2 -lt $n4 && $n2 -lt $n5 ]]

then

echo $n2 "is min"

else

if [[ $n3 -lt $n4 && $n2 -lt $n5 ]]

then

echo $n3 "is min"

else

if [ $n4 -lt $n5 ]

then

echo $n4 "is min"

else

echo $n5 "is min"

fi

fi

fi

fi

Output:

nida@nida-PC MINGW64 ~/stage2/shellcommands/selection

$ ./Q5.sh

+ n1=790

+ n2=749

+ n3=898

+ n4=317

+ n5=984

+ [[ 790 -lt 749 ]]

+ [[ 749 -lt 898 ]]

+ [[ 749 -lt 898 ]]

+ [[ 749 -lt 317 ]]

+ [[ 898 -lt 317 ]]

+ '[' 317 -lt 984 ']'

+ echo 317 'is min'

317 is min

1. Write a 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.

**Code:**

#!/bin/bash -x

read -p " Enter Date:-" date

read -p " Enter Month:-" Month

if (( ($Month < 6 & $Month >3 & $date<=30 & $date != 0) ))

then

echo "True"

else

if (( ($Month == 3 & $date >= 20 & $date <= 31 & $date != 0) ))

then

echo "True"

else

if (( ($Month == 6 & $date <= 20 & $date != 0) ))

then

echo "True"

else

if (( ($Month == 5 & $date <= 31 & $date != 0) ))

then

echo "True"

else

echo "False"

fi

fi

fi

fi

**Output:**

nida@nida-PC MINGW64 ~/stage2/shellcommands/selection

$ ./Q6.sh

+ read -p ' Enter Date:-' date

Enter Date:-31

+ read -p ' Enter Month:-' Month

Enter Month:-6

+ (( (6 < 6 & 6 >3 & 31<=30 & 31 != 0) ))

+ (( (6 == 3 & 31 >= 20 & 31 <= 31 & 31 != 0) ))

+ (( (6 == 6 & 31 <= 20 & 31 != 0) ))

+ (( (6 == 5 & 31 <= 31 & 31 != 0) ))

+ echo False

False

nida@nida-PC MINGW64 ~/stage2/shellcommands/selection

$ ./Q6.sh

+ read -p ' Enter Date:-' date

Enter Date:-20

+ read -p ' Enter Month:-' Month

Enter Month:-6

+ (( (6 < 6 & 6 >3 & 20<=30 & 20 != 0) ))

+ (( (6 == 3 & 20 >= 20 & 20 <= 31 & 20 != 0) ))

+ (( (6 == 6 & 20 <= 20 & 20 != 0) ))

+ echo True

True

nida@nida-PC MINGW64 ~/stage2/shellcommands/selection

$ ./Q6.sh

+ read -p ' Enter Date:-' date

Enter Date:-21

+ read -p ' Enter Month:-' Month

Enter Month:-6

+ (( (6 < 6 & 6 >3 & 21<=30 & 21 != 0) ))

+ (( (6 == 3 & 21 >= 20 & 21 <= 31 & 21 != 0) ))

+ (( (6 == 6 & 21 <= 20 & 21 != 0) ))

+ (( (6 == 5 & 21 <= 31 & 21 != 0) ))

+ echo False

False

nida@nida-PC MINGW64 ~/stage2/shellcommands/selection

$ ./q2\_3.sh

+ read -p ' Enter Date:-' date

Enter Date:-31

+ read -p ' Enter Month:-' Month

Enter Month:-5

+ (( (5 < 6 & 5 >3 & 31<=30 & 31 != 0) ))

+ (( (5 == 3 & 31 >= 20 & 31 <= 31 & 31 != 0) ))

+ (( (5 == 6 & 31 <= 20 & 31 != 0) ))

+ (( (5 == 5 & 31 <= 31 & 31 != 0) ))

+ echo True

True

nida@nida-PC MINGW64 ~/stage2/shellcommands/selection

$ ./q2\_3.sh

+ read -p ' Enter Date:-' date

Enter Date:-32

+ read -p ' Enter Month:-' Month

Enter Month:-5

+ (( (5 < 6 & 5 >3 & 32<=30 & 32 != 0) ))

+ (( (5 == 3 & 32 >= 20 & 32 <= 31 & 32 != 0) ))

+ (( (5 == 6 & 32 <= 20 & 32 != 0) ))

+ (( (5 == 5 & 32 <= 31 & 32 != 0) ))

+ echo False

False

1. Write a program that takes a year as input and outputs the Year is a Leap Year or not a Leap Year. A Leap Year checks for 4 Digit Number, Divisible by 4 and not 100 unless divisible by 400.

**Code:**

#! /bin/bash -x

read -p "Enter the year you want to check leap year for: " year

if (( ($year % 4 == 0 & $year % 100 != 0 | $year % 400 == 0) ))

then

echo $year "is leap year"

else

echo $year "is not leap year"

fi

**Output:**

nida@nida-PC MINGW64 ~/stage2/shellcommands/selection

$ ./q3.sh

+ read -p 'Enter the year you want to check leap year for: ' year

Enter the year you want to check leap year for: 1900

+ (( (1900 % 4 == 0 & 1900 % 100 != 0 | 1900 % 400 == 0) ))

+ echo 1900 'is not leap year'

1900 is not leap year

nida@nida-PC MINGW64 ~/stage2/shellcommands/selection

$ ./q3.sh

+ read -p 'Enter the year you want to check leap year for: ' year

Enter the year you want to check leap year for: 2036

+ (( (2036 % 4 == 0 & 2036 % 100 != 0 | 2036 % 400 == 0) ))

+ echo 2036 'is leap year'

2036 is leap year

1. Write a program to simulate a coin flip and print out "Heads" or "Tails" accordingly.

**Code:**

#! /bin/bash -x

flip=$((RANDOM % 2))

if [ $flip -eq 1 ]

then

echo "Its Head"

else

echo "Its Tail"

fi

**Output:**

nida@nida-PC MINGW64 ~/stage2/shellcommands/selection

$ ./q4.sh

+ flip=1

+ '[' 1 -eq 1 ']'

+ echo 'Its Head'

Its Head

nida@nida-PC MINGW64 ~/stage2/shellcommands/selection

$ ./q4.sh

+ flip=0

+ '[' 0 -eq 1 ']'

+ echo 'Its Tail'

Its Tail

Selection Practice Problems with if, elif and else

1. Read a single digit number and write the number in word

**Code:**

#! /bin/bash -x

read -p "Enter number: " num

if [ $num -eq 1 ]

then

echo "ONE"

elif [ $num -eq 2 ]

then

echo "TWO"

elif [ $num -eq 3 ]

then

echo "THREE"

elif [ $num -eq 4 ]

then

echo "FOUR"

elif [ $num -eq 5 ]

then

echo "FIVE"

elif [ $num -eq 6 ]

then

echo "SIX"

elif [ $num -eq 7 ]

then

echo "SEVEN"

elif [ $num -eq 8 ]

then

echo "EIGHT"

elif [ $num -eq 9 ]

then

echo "NINE"

else

echo "enter 1 digit number"

fi

**Output:**

nida@nida-PC MINGW64 ~/stage2/shellcommands/selection2

$ ./q1.sh

+ read -p 'Enter number: ' num

Enter number: 5

+ '[' 5 -eq 1 ']'

+ '[' 5 -eq 2 ']'

+ '[' 5 -eq 3 ']'

+ '[' 5 -eq 4 ']'

+ '[' 5 -eq 5 ']'

+ echo FIVE

FIVE

nida@nida-PC MINGW64 ~/stage2/shellcommands/selection2

$ ./q1.sh

+ read -p 'Enter number: ' num

Enter number: 10

+ '[' 10 -eq 1 ']'

+ '[' 10 -eq 2 ']'

+ '[' 10 -eq 3 ']'

+ '[' 10 -eq 4 ']'

+ '[' 10 -eq 5 ']'

+ '[' 10 -eq 6 ']'

+ '[' 10 -eq 7 ']'

+ '[' 10 -eq 8 ']'

+ '[' 10 -eq 9 ']'

+ echo 'enter 1 digit number'

enter 1 digit number

1. Read a Number and Display the week day (Sunday, Monday,...)

**Code:**

#! /bin/bash -x

read -p "Enter WEEK DAY: " num

if [ $num -eq 1 ]

then

echo "SUNDAY"

elif [ $num -eq 2 ]

then

echo "MONDAY"

elif [ $num -eq 3 ]

then

echo "TUESDAY"

elif [ $num -eq 5 ]

then

echo "THURSDAY"

elif [ $num -eq 4 ]

then

echo "WEDNESDAY"

elif [ $num -eq 6 ]

then

echo "FRIDAY"

elif [ $num -eq 7 ]

then

echo "SATURDAY"

else

echo "Enter proper value"

fi

**Output**

nida@nida-PC MINGW64 ~/stage2/shellcommands/selection2

$ ./q1.sh

+ read -p 'Enter WEEK DAY: ' num

Enter WEEK DAY: 3

+ '[' 3 -eq 1 ']'

+ '[' 3 -eq 2 ']'

+ '[' 3 -eq 3 ']'

+ echo TUESDAY

TUESDAY

nida@nida-PC MINGW64 ~/stage2/shellcommands/selection2

$ ./q1.sh

+ read -p 'Enter WEEK DAY: ' num

Enter WEEK DAY: 8

+ '[' 8 -eq 1 ']'

+ '[' 8 -eq 2 ']'

+ '[' 8 -eq 3 ']'

+ '[' 8 -eq 5 ']'

+ '[' 8 -eq 4 ']'

+ '[' 8 -eq 6 ']'

+ '[' 8 -eq 7 ']'

+ echo 'Enter proper value'

Enter proper value

1. Read a Number 1, 10, 100, 1000, etc and display unit, ten, hundred,...

**Code**:

#! /bin/bash -x

read -p "Enter number: " num

if [ $num -eq 1 ]

then

echo "unit"

elif [ $num -eq 10 ]

then

echo "Ten"

elif [ $num -eq 100 ]

then

echo "Hundred"

elif [ $num -eq 1000 ]

then

echo "Thousand"

else

echo "Enter proper value"

fi

**Output:**

nida@nida-PC MINGW64 ~/stage2/shellcommands/selection2

$ ./q1.sh

+ read -p 'Enter number: ' num

Enter number: 1

+ '[' 1 -eq 1 ']'

+ echo unit

unit

nida@nida-PC MINGW64 ~/stage2/shellcommands/selection2

$ ./q1.sh

+ read -p 'Enter number: ' num

Enter number: 10

+ '[' 10 -eq 1 ']'

+ '[' 10 -eq 10 ']'

+ echo Ten

Ten

4. Enter 3 Numbers do following arithmetic operation and find the one that is maximum and minimum

1. a + b \* c 3. c + a / b

2. a % b + c 4. a \* b + c

**Code:**

#! /bin/bash -x

read -p "Enter value for a " a

read -p "Enter value for b " b

read -p "Enter value for c " c

v1=`expr $a + $b \\* $c`

v2=`expr $a % $b + $c`

v3=`expr $c + $a / $b`

v4=`expr $a \\* $b + $c`

if [[ $v1 -gt $v2 && $v1 -gt $v3 && $v1 -gt $v4 ]]

then

echo $v1 "is max"

elif [[ $v2 -gt $v3 && $v2 -gt $v4 ]]

then

echo $v2 "is max"

elif [ $v3 -gt $v4 ]

then

echo $v3 "is max"

else

echo $v4 "is max"

fi

if [[ $v1 -lt $v2 && $v1 -lt $v3 && $v1 -lt $v4 ]]

then

echo $v1 "is min"

elif [[ $v2 -lt $v3 && $v2 -lt $v4 ]]

then

echo $v2 "is min"

elif [ $v3 -lt $v4 ]

then

echo $v3 "is min"

else

echo $v4 "is min"

fi

**Output:**

nida@nida-PC MINGW64 ~/stage2/shellcommands/selection2

$ ./q2.sh

+ read -p 'Enter value for a ' a

Enter value for a 1

+ read -p 'Enter value for b ' b

Enter value for b 2

+ read -p 'Enter value for c ' c

Enter value for c 3

++ expr 1 + 2 '\*' 3

+ v1=7

++ expr 1 % 2 + 3

+ v2=4

++ expr 3 + 1 / 2

+ v3=3

++ expr 1 '\*' 2 + 3

+ v4=5

+ [[ 7 -gt 4 ]]

+ [[ 7 -gt 3 ]]

+ [[ 7 -gt 5 ]]

+ echo 7 'is max'

7 is max

+ [[ 7 -lt 4 ]]

+ [[ 4 -lt 3 ]]

+ '[' 3 -lt 5 ']'

+ echo 3 'is min'

3 is min

Selection Practice Problems with case statement

1. Read a single digit number and write the number in word using Case

**Code:**

#!/bin/bash -x

read -p "Enter number: " num

case "$num" in

"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 "Enter one digit number"

;;

esac

**Output:**

nida@nida-PC MINGW64 ~/stage2/shellcommands/switch

$ ./q1.sh

+ read -p 'Enter number: ' num

Enter number: 10

+ case "$num" in

+ echo 'Enter one digit number'

Enter one digit number

nida@nida-PC MINGW64 ~/stage2/shellcommands/switch

$ ./q1.sh

+ read -p 'Enter number: ' num

Enter number: 3

+ case "$num" in

+ echo THREE

THREE

1. Read a Number and Display the week day (Sunday, Monday,...)

**Code:**

#!/bin/bash -x

read -p "Enter day of week: " num

case "$num" in

"1") echo "SUNDAY"

;;

"2") echo "MONDAY"

;;

"3") echo "TUESDAY"

;;

"4") echo "WEDNESDAY"

;;

"5") echo "THURSDAY"

;;

"6") echo "FRIDAY"

;;

"7") echo "SATURDAY"

;;

\*) echo "Enter PROPER number within range of 1-7"

;;

esac

**Output:**

nida@nida-PC MINGW64 ~/stage2/shellcommands/switch

$ ./q1.sh

+ read -p 'Enter day of week: ' num

Enter day of week: 1

+ case "$num" in

+ echo SUNDAY

SUNDAY

nida@nida-PC MINGW64 ~/stage2/shellcommands/switch

$ ./q1.sh

+ read -p 'Enter day of week: ' num

Enter day of week: 8

+ case "$num" in

+ echo 'Enter PROPER number within range of 1-7'

Enter PROPER number within range of 1-7

1. Read a Number 1, 10, 100, 1000, etc and display unit, ten, hundred,...

**Code:**

#!/bin/bash -x

read -p "Enter number: " num

case "$num" in

"1") echo "units"

;;

"10") echo "Tens"

;;

"100") echo "Hundred"

;;

"1000") echo "Thousand"

;;

\*) echo "Enter PROPER number"

;;

esac

**Output:**

nida@nida-PC MINGW64 ~/stage2/shellcommands/switch

$ ./q1.sh

+ read -p 'Enter number: ' num

Enter number: 1000

+ case "$num" in

+ echo Thousand

Thousand

nida@nida-PC MINGW64 ~/stage2/shellcommands/switch

$ ./q1.sh

+ read -p 'Enter number: ' num

Enter number: 10000

+ case "$num" in

+ echo 'Enter PROPER number'

Enter PROPER number

4. 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

**Code:**

#!/bin/bash -x

echo "Conversion"

echo "1. Feet to Inch"

echo "2. Inch to Feet"

echo "3. Feet to Meter"

echo "4. Meter to Feet"

read -p "Enter choice: " ch

case "$ch" in

"1") read -p "Enter the value in Feet: " num

inches=$(awk "BEGIN{ print $num \* 12 }")

echo $num "Feet = " $inches "Inches"

;;

"2") read -p "Enter the value in inch: " num

feet=$(awk "BEGIN{print $num / 12}")

echo $num "Inches = " $feet "Feet"

;;

"3") read -p "Enter the value in Feet: " num

meter=$(awk "BEGIN{print $num / 3.281}")

echo $num "Feet = " $meter "Meter"

;;

"4") read -p "Enter the value in Meter: " num

feet=$(awk "BEGIN{print $num \* 3.281}")

echo $num "Meter = " $feet "Feet"

;;

\*) echo "Enter PROPER choice"

;;

esac

**Output:**

nida@nida-PC MINGW64 ~/stage2/shellcommands/switch

$ ./q1.sh

+ echo Conversion

Conversion

+ echo '1. Feet to Inch'

1. Feet to Inch

+ echo '2. Inch to Feet'

2. Inch to Feet

+ echo '3. Feet to Meter'

3. Feet to Meter

+ echo '4. Meter to Feet'

4. Meter to Feet

+ read -p 'Enter choice: ' ch

Enter choice: 4

+ case "$ch" in

+ read -p 'Enter the value in Meter: ' num

Enter the value in Meter: 12

++ awk 'BEGIN{print 12 \* 3.281}'

+ feet=39.372

+ echo 12 'Meter = ' 39.372 Feet

12 Meter = 39.372 Feet