OS LAB 2

U18CO021: SAHIL BONDRE

1. Shell script Program to accept a character and check whether it is an

- **Lower case alphabet**
- **Upper case alphabet**
- **A digit**
- **Special symbol**
- **Vowel**

Using case control structure

```
#!/bin/bash

printf "Enter A charecter: "
read -n1 ans
echo ""
echo "You entered:" $ans

case $ans in

[A-Z])
  echo "Uppercase"
  ;;
[a-z])
  echo "Lowercase"
  ;;
[0-9])
  echo "Digit"
  ;;
*)
  echo "Special Symbol"
  ;;
esac

case $ans in
[AEIOUaeiou])
  echo "It is a vowel"
  ;;
*)
```

```
  echo "Not a Vowel"
  ;;
esac
```

```
➜  q01 git:(master) ✗ ./script.sh
Enter A charecter: .
You entered: .
Special Symbol
➜  q01 git:(master) ✗ ./script.sh
Enter A charecter: A
You entered: A
Uppercase
It is a vowel
➜  q01 git:(master) ✗
```

2. Using case .. esac structure

- **Find the number of users logged into the system**
- **Print the calendar for current year**
- **Print the date**

```
#!/bin/bash

echo "enter l for Number of users logged in"
echo "enter c for calendar of current year"
echo "enter d for date"
read -n1 ans
echo ""
echo "You entered:" $ans

case "$ans" in
'l')
  users=$(who | sort --key=1,1 --unique | wc --lines)
  echo "Number of users logged in:" $users
  ;;
'c')
  cal -y
  ;;
'd')
  date
  ;;
```

```
*)
  echo "Unknown charecter entered..."
  ;;
esac
```

```
➜  q02 git:(master) ✗ ./script.sh
enter l for Number of users logged in
enter c for calendar of current year
enter d for date
l
You entered: l
Number of users logged in: 1
➜  q02 git:(master) ✗ ./script.sh
enter l for Number of users logged in
enter c for calendar of current year
enter d for date
c
You entered: c
                               2021

       January               February                March
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                1  2       1  2  3  4  5  6       1  2  3  4  5  6
 3  4  5  6  7  8  9    7  8  9 10 11 12 13    7  8  9 10 11 12 13
10 11 12 13 14 15 16   14 15 16 17 18 19 20   14 15 16 17 18 19 20
17 18 19 20 21 22 23   21 22 23 24 25 26 27   21 22 23 24 25 26 27
24 25 26 27 28 29 30   28                     28 29 30 31
31
        April                   May                   June
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
             1  2  3                         1          1  2  3  4  5
 4  5  6  7  8  9 10    2  3  4  5  6  7  8    6  7  8  9 10 11 12
11 12 13 14 15 16 17    9 10 11 12 13 14 15   13 14 15 16 17 18 19
18 19 20 21 22 23 24   16 17 18 19 20 21 22   20 21 22 23 24 25 26
25 26 27 28 29 30      23 24 25 26 27 28 29   27 28 29 30
                       30 31
        July                  August                September
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
             1  2  3    1  2  3  4  5  6  7              1  2  3  4
 4  5  6  7  8  9 10    8  9 10 11 12 13 14    5  6  7  8  9 10 11
11 12 13 14 15 16 17   15 16 17 18 19 20 21   12 13 14 15 16 17 18
18 19 20 21 22 23 24   22 23 24 25 26 27 28   19 20 21 22 23 24 25
```

3. Shell Script Program to check whether a given file is a directory or not.

```
#!/bin/bash

PASSED=$1

if [ -d "${PASSED}" ]; then
```

```
  echo "${PASSED} is a directory"
elif [ -f "${PASSED}" ]; then
  echo "${PASSED} is a file"
else
  echo "${PASSED} is not valid"
  exit 1
fi
```

```
➜  q03 git:(master) ✗ ./script.sh
 is not valid
➜  q03 git:(master) ✗ ./script.sh .
. is a directory
➜  q03 git:(master) ✗ ./script.sh script.sh
script.sh is a file
➜  q03 git:(master) ✗
```

4. Shell Script Program to Count number of files in a Directory.

```
#!/bin/bash

printf "Number of files: "
ls -1q | wc -l
```

```
➜  q04 git:(master) ✗ ./script.sh
Number of files: 2
➜  q04 git:(master) ✗ ls
script.sh  test.log
➜  q04 git:(master) ✗
```

5. Shell Script Program to copy contents of one file to another.

```
#!/bin/bash

file1=$1
file2=$2

if [ -f "$file1" ]; then
  cat $1 >>$2
else
  echo "$file1 does not exist."
fi
```

```
→  q05 git:(master) ✗ ./script.sh script.sh test
→  q05 git:(master) ✗ ls
script.sh  test
→  q05 git:(master) ✗ cat test
#!/bin/bash

file1=$1
file2=$2

if [ -f "$file1" ]; then
  cat $1 >>$2
else
  echo "$file1 does not exist."
fi
→  q05 git:(master) ✗
```

6. Write a shell script to add two numbers supplied by user and supplied as command line argument.

```bash
#!/bin/bash

re='[0-9]'
if ! [[ $1 =~ $re ]]; then
  echo "error: Not a number" >&2
  exit 1
fi

if ! [[ $2 =~ $re ]]; then
  echo "error: Not a number" >&2
  exit 1
fi

echo "Sum:" $(($1 + $2))
```

```
→  q06 git:(master) ✗ ./script.sh 4 s
error: Not a number
→  q06 git:(master) ✗ ./script.sh 4 45
Sum 49
→  q06 git:(master) ✗
```

7. Write a shell script to find out the biggest number form given three numbers. Numbers are supplied by command line argument.

```
#!/bin/bash

re='[0-9]'
if ! [[ $1 =~ $re ]]; then
  echo "error: Not a number" >&2
  exit 1
fi

if ! [[ $2 =~ $re ]]; then
  echo "error: Not a number" >&2
  exit 1
fi

if ! [[ $3 =~ $re ]]; then
  echo "error: Not a number" >&2
  exit 1
fi

arr=($1 $2 $3)
max=${arr[0]}
for n in "${arr[@]}"; do
  ((n > max)) && max=$n
done
echo "Max:" $max
```

```
→  q07 git:(master) ✗ ./script.sh 4 5 6
Max: 6
→  q07 git:(master) ✗ ./script.sh 4 55 7
Max: 55
→  q07 git:(master) ✗ ./script.sh 4 55 -7
Max: 55
→  q07 git:(master) ✗
```

8. Implement simple calculator.Numbers are supplied by command line argument

```
#!/bin/bash

if [ $# -ne 2 ]; then
```

```
  echo "2 command line arguments are required"
  exit 2
fi

if ! [[ $1 =~ $re ]]; then
  echo "error: Not a number" >&2
  exit 1
fi

if ! [[ $2 =~ $re ]]; then
  echo "error: Not a number" >&2
  exit 1
fi

a=$1
b=$2

echo "Enter Choice :"
echo "1. Addition"
echo "2. Subtraction"
echo "3. Multiplication"
echo "4. Division"
read -n1 ch
echo " "
# Switch Case to perform
# calculator operations
case "$ch" in
'1')
  res=$(echo $a + $b | bc)
  ;;
'2')
  res=$(echo $a - $b | bc)
  ;;
'3')
  res=$(echo $a \* $b | bc)
  ;;
'4')
  res=$(echo "scale=2; $a / $b" | bc)
  ;;
*)
  echo "Invalid Choice"
  exit 1
  ;;
esac
echo "Result : $res"
```

```
→  q08 git:(master) ✗ ./script.sh 4 5
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
2
Result : -1
→  q08 git:(master) ✗ ./script.sh 4 5
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
3
Result : 20
→  q08 git:(master) ✗
```

9. Write a shell script to print numbers in descending order using while loop.

```bash
#! /bin/bash

num=(5 17 20 67 1 3)

echo "Original array num:"
for l in "${num[@]}"; do
  printf "$l "
done

echo ""

for ((i = 0; i < ${#num[@]}; i++)); do
  for ((j = 0; j < ${#num[@]}; j++)); do
    if [[ ${num[$j]} -lt ${num[$i]} ]]; then
      tmp=${num[$i]}
      num[$i]=${num[$j]}
      num[$j]=${tmp}
    fi
  done
done
```

```
echo ""
echo "Descending order num:"
for k in "${num[@]}"; do
  printf "$k "
done

echo ""
```

```
→  q09 git:(master) ✗ ./script.sh
Original array num:
5 17 20 67 1 3

Descending order num:
67 20 17 5 3 1
→  q09 git:(master) ✗
```

10. Write a shell script to create a simple calculator using switch-case statements.

```
#!/bin/bash

if [ $# -ne 2 ]; then
  echo "2 command line arguments are required"
  exit 2
fi

if ! [[ $1 =~ $re ]]; then
  echo "error: Not a number" >&2
  exit 1
fi

if ! [[ $2 =~ $re ]]; then
  echo "error: Not a number" >&2
  exit 1
fi

a=$1
b=$2

echo "Enter Choice :"
echo "1. Addition"
echo "2. Subtraction"
```

```
echo "3. Multiplication"
echo "4. Division"
read -n1 ch
echo " "
# Switch Case to perform
# calulator operations
case "$ch" in
'1')
  res=$(echo $a + $b | bc)
  ;;
'2')
  res=$(echo $a - $b | bc)
  ;;
'3')
  res=$(echo $a \* $b | bc)
  ;;
'4')
  res=$(echo "scale=2; $a / $b" | bc)
  ;;
*)
  echo "Invalid Choice"
  exit 1
  ;;
esac
echo "Result : $res"
```

```
➜  q10 git:(master) ✗ ./script.sh 4 5
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
1
Result : 9
➜  q10 git:(master) ✗ ./script.sh 4 5
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
2
Result : -1
➜  q10 git:(master) ✗
```

11. Write a shell script to print a given number in reverse order.

```bash
#! /bin/bash

read -p "Enter a number: " num
echo $num | rev
```

```
➜  q11 git:(master) ✗ ./script.sh
Enter a number: 4567
7654
➜  q11 git:(master) ✗
```

12. Write a shell script to print sum of all digits of a given number

```bash
#!/bin/bash

read -p "Enter number: " num
sum=0
```

```
while [ $num -gt 0 ]; do
  mod=$((num % 10))  # split
  sum=$((sum + mod)) # add
  num=$((num / 10))  # divide num by 10.
done

echo "Sum of digits of number: $sum"
```

```
➜  q12 git:(master) ✗ ./script.sh
Enter number: 457
Sum of digits of number: 16
➜  q12 git:(master) ✗
```

13. Find the factorial value of given input number.

```
#! /bin/bash

read -p "Enter number: " num
re='[0-9]'
if ! [[ $num =~ $re ]]; then
  echo "error: Not a number" >&2
  exit 1
fi

fact=1

while [ $num -gt 1 ]; do
  fact=$((fact * num))
  num=$((num - 1))
done

echo "Factorial of number is: $fact"
```

```
➜  q13 git:(master) ✗ ./script.sh
Enter number: 7
Factorial of number is: 5040
➜  q13 git:(master) ✗
```

14. Generate and display Fibonacci series.

```
#! /bin/bash

read -p "Enter number of values to show in fibonnaci series: " N

echo "The Fibonacci series is : "

a=0
b=1
for ((i = 0; i < N; i++)); do
  echo -n "$a "
  fn=$((a + b))
  a=$b
  b=$fn
done
echo ""
```

```
➜  q14 git:(master) ✗ ./script.sh
Enter number of values to show in fibonnaci series: 5
The Fibonacci series is :
0 1 1 2 3
➜  q14 git:(master) ✗
```

15. Display all even numbers within a given range.

```
#! /bin/bash

read -p "Enter lower range: " first
read -p "Enter upper range: " second

for ((i = $first; i <= $second; ++i)); do
  mod=$((i % 2))
  if [ "$mod" -eq "0" ]; then
    echo $i
  fi
done
```

```
→  q15 git:(master) ✗ ./script.sh
Enter lower range: 1
Enter upper range: 9
2
4
6
8
→  q15 git:(master) ✗
```

16. Find out the number of characters, words and lines from a given file.

```bash
#! /bin/bash


if [ -f "$1" ]; then
  w=$(cat $1 | wc -w)
  c=$(cat $1 | wc -c)
  l=$(grep -c "." $1)
  echo Number of characters in $1 is $c
  echo Number of words in $1 is $w
  echo Number of lines in $1 is $l
else
  echo "$1 does not exist."
fi
```

```
→  q16 git:(master) ✗ ./script.sh script.sh
Number of characters in script.sh is 248
Number of words in script.sh is 53
Number of lines in script.sh is 11
→  q16 git:(master) ✗
```