Operating System (CS301)

Assignment - 2

U19CS012

(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.

Script:

```
# Path to Intepretor
#! /bin/bash

# Prompt the User to Read a Character
read -p "Enter One Character : " USER_CHAR

case $USER_CHAR in
    "A") echo "$USER_CHAR is UPPER case Vowel Alphabet";;
    "E") echo "$USER_CHAR is UPPER case Vowel Alphabet";;
    "I") echo "$USER_CHAR is UPPER case Vowel Alphabet";;
    "O") echo "$USER_CHAR is UPPER case Vowel Alphabet";;
    "U") echo "$USER_CHAR is UPPER case Vowel Alphabet";;
    "a") echo "$USER_CHAR is lower case Vowel Alphabet";;
    "e") echo "$USER_CHAR is lower case Vowel Alphabet";;
    "i") echo "$USER_CHAR is lower case Vowel Alphabet";;
    "o") echo "$USER_CHAR is lower case Vowel Alphabet";;
    "u") echo "$USER_CHAR is lower case Vowel Alphabet";;
    [[:digit:]]) echo "$USER_CHAR is Digit";;
    [[:upper:]]) echo "$USER_CHAR is UPPER case Alphabet" ;;
    [[:lower:]]) echo "$USER_CHAR is lower case Alphabet" ;;
    *) echo "$USER_CHAR is Special symbol"  ;;
esac
```

Output:

```
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q1.sh
Enter One Character : A
A is UPPER case Vowel Alphabet
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q1.sh
Enter One Character : B
B is UPPER case Alphabet
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q1.sh
Enter One Character : e
e is lower case Vowel Alphabet
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q1.sh
Enter One Character : g
g is lower case Alphabet
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q1.sh
Enter One Character : 4
4 is Digit
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q1.sh
Enter One Character : @
@ is Special symbol
```

(2) Using case .. esac structure

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

Script:

```
#!/bin/bash

echo "Enter n -> Number of Users Logged In"
echo "Enter c -> Calendar of current year"
echo "Enter d -> Date"

read choice
echo ""

case "$choice" in
    'n')
        # Counting unique logins when *who* doesn't have a --count flag.
        users=$(who | sort --key=1,1 --unique | wc --lines)
        echo "Number of Users Logged in: " $users
    ;;
    'c')
        # cal : Shows current month calendar on the terminal with the current date highlighte
d.
        cal -y
    ;;
    'd')
```

```
        # date command is used to display the system date and time
        date
    ;;
    *)
        echo "Please Enter a Valid Character"
    ;;
esac

# References:
# 1.) https://stackoverflow.com/questions/28208069/linux-bash-script-to-determine-number-of-users-logged-in
# 2.) https://www.geeksforgeeks.org/cal-command-in-linux-with-examples/
# 3.) https://www.geeksforgeeks.org/date-command-linux-examples/
```

Output:

```
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q2.sh
Enter n -> Number of Users Logged In
Enter c -> Calendar of current year
Enter d -> Date
n

Number of Users Logged in:  0
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$
```

Continued on Next Page!

```
Enter c -> Calendar of current year
Enter d -> Date
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
25 26 27 28 29 30 31  29 30 31              26 27 28 29 30

      October               November              December
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
 3  4  5  6  7  8  9   7  8  9 10 11 12 13   5  6  7  8  9 10 11
10 11 12 13 14 15 16  14 15 16 17 18 19 20  12 13 14 15 16 17 18
17 18 19 20 21 22 23  21 22 23 24 25 26 27  19 20 21 22 23 24 25
24 25 26 27 28 29 30  28 29 30              26 27 28 29 30 31
31
```

```
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q2.sh
Enter n -> Number of Users Logged In
Enter c -> Calendar of current year
Enter d -> Date
d

Tue Aug 10 20:34:57 IST 2021
```

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

Script:

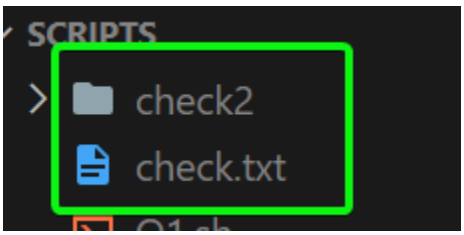
```
#!/bin/bash

FILE_NAME=$1

if [ -d "${FILE_NAME}" ];
then
    echo "${FILE_NAME} is a Directory"
elif [ -f "${FILE_NAME}" ]; then
    echo "${FILE_NAME} is a File"
else
    echo "${FILE_NAME} is Not Valid File/Directory"
    exit 1
fi
```

Output:

```
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q3.sh check.txt
check.txt is a File
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q3.sh check2
check2 is a Directory
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q3.sh abcd.txt
abcd.txt is Not Valid File/Directory
```



(4) Shell Script Program to Count Number of files in a Directory.

Script:

```
#!/bin/bash

printf "Number of Files in Current Directory: "
ls -1q | wc -l

# Reference
# 1.) https://stackoverflow.com/questions/20895290/count-number-of-files-within-a-directory-in-linux
# 2.) https://www.geeksforgeeks.org/practical-applications-ls-command-linux/
```

Output:

```
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q4.sh
Number of Files in Current Directory: 18
```

```
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ls -1
Q1.sh
Q10.sh
Q11.sh
Q12.sh
Q13.sh
Q14.sh
Q15.sh
Q16.sh
Q2.sh
Q3.sh
Q4.sh
Q5.sh
Q6.sh
Q7.sh
Q8.sh
Q9.sh
check.txt
check2
```

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

Script:

```
#!/bin/bash

file1=$1
file2=$2

if [ -f "$file1" ]; then
    cat $1 >>$2
else
    echo "$file1 Does Not Exist."
fi
```

Output:

```
Q9.sh
check.txt
check2
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q5.sh Q5.sh check.txt
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ cat check.txt
# (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
```

file1 file2

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

Script:

```
#!/bin/bash

# Regular Expression for Number
re='^[+-]?[0-9]+?$'

# Check if Input is Valid
if ! [[ $1 =~ $re ]]; then
    echo "Error: Input is Not a Valid Number" >&2
    exit 1
fi

if ! [[ $2 =~ $re ]]; then
    echo "Error: Input is Not a Valid Number" >&2
    exit 1
fi


echo "Sum of Two Given Numbers : " $(($1 + $2))

# References
# 1.) https://stackoverflow.com/questions/806906/how-do-i-test-if-a-variable-is-a-number-in-bash/806923
```

Output:

```
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q6.sh 11 25
Sum of Two Given Numbers :  36
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q6.sh -10 25
Sum of Two Given Numbers :  15
```

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

Script:

```
#!/bin/bash

# Regular Expression for Number
re='^[+-]?[0-9]+?$'

# Input Validation
if ! [[ $1 =~ $re ]]; then
    echo "Error: Input is Not a Valid Number" >&2
    exit 1
fi
if ! [[ $2 =~ $re ]]; then
    echo "Error: Input is Not a Valid Number" >&2
    exit 1
fi
if ! [[ $3 =~ $re ]]; then
    echo "Error: Input is Not a Valid Number" >&2
    exit 1
fi

# Method 1

# Store it in Array for
arr=($1 $2 $3)

# Intializa the First element to be Maximum
max_element=${arr[0]}

# Using ${arr[@]} or ${arr[*]}, we can access the all array elements.
for n in "${arr[@]}";
do
    ((n > max_element)) && max_element=$n
done

echo "Maximum of all 3 Numbers : " $max_element

# Method 2

if [ $1 -gt $2 ] && [ $1 -gt $3 ]
then
    echo "Maximum Element : " $1
elif [ $2 -gt $1 ] && [ $2 -gt $3 ]
then
    echo "Maximum Element : " $2
else
    echo "Maximum Element : " $3
fi
```

Output:

```
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts
$ ./Q7.sh 89 -23 43
Maximum of all 3 Numbers :  89
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts
$ ./Q7.sh 1000 23 -1
Maximum of all 3 Numbers :  1000
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts
$ ./Q7.sh 20 20 20
Maximum of all 3 Numbers :  20
```

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

Script:

```
#!/bin/bash

# Check for Valid Input

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

# Inputs from User
a=$1
b=$2

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

echo "Your Choice : "
```

```
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"

# bc command is used for command line calculator.

# References
# 1.) https://www.geeksforgeeks.org/simple-calculator-bash/
```

Output:

```
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q8.sh 12 7
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
Your Choice :
1
Result : 19
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q8.sh 12 7
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
Your Choice :
2
Result : 5
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q8.sh 12 7
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
Your Choice :
3
Result : 84
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q8.sh 12 7
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
Your Choice :
4
Result : 1.71
```

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

Script:

```
#! /bin/bash

# User Need to Change this Input
arr=(15 71 120 7 3 35)

n=${#arr[@]}
echo "Original Number(s) in Array : "

for x in "${arr[@]}";
```

```
do
    printf "$x "
done

echo ""

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

echo ""

echo "Final Number(s) of Array [Descending] : "
i=0
while [ $i -lt $n ];
do
    printf "${arr[i]} "
    i=$((i + 1))
done

echo ""
```

Output:

```
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q9.sh
Original Number(s) in Array :
15 71 120 7 3 35

Final Number(s) of Array [Descending] :
120 71 35 15 7 3
```

(10) Write a shell script to create a simple calculator using switch-case statement. [Same as Question 8]

Script:

```
#!/bin/bash

# Check for Valid Input
```

```
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

# Inputs from User
a=$1
b=$2

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

echo "Your Choice : "
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"
```

Output:

```
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q8.sh 12 7
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
Your Choice :
1
Result : 19
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q8.sh 12 7
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
Your Choice :
2
Result : 5
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q8.sh 12 7
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
Your Choice :
3
Result : 84
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q8.sh 12 7
Enter Choice :
1. Addition
2. Subtraction
3. Multiplication
4. Division
Your Choice :
4
Result : 1.71
```

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

Script:

```
#! /bin/bash
# Method 1

# Number can be Treated as String
read -p "Enter a Number: " num
echo $num | rev

# Method 2

echo "Enter a Number : "
read n

digit=0
rev=0

while [ $n -gt 0 ]
do
    digit=$(( $n % 10 ))
    # Left Shift Old Answer and Add the Current Digit to it
    rev=`expr $rev \* 10 + $digit`
    n=$(( $n / 10 ))
done

echo "Reverse Number : $rev"

# References
# 1.) https://unix.stackexchange.com/questions/65913/print-given-number-in-reverse-order/65924
```

Output:

```
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q11.sh
Enter a Number: 48729305
50392784
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q11.sh
Enter a Number :
49283057
Reverse Number : 75038294
```

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

Script:

```
#!/bin/bash

read -p "Enter Number : " num

# Intialize the Sum = 0
sum=0

while [ $num -gt 0 ];
do
    digit=$((num % 10))
    sum=$((sum + digit))
    num=$((num / 10))
done

echo "Sum of Digits of Number : $sum"
```

Output:

```
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q12.sh
Enter Number : 12345
Sum of Digits of Number : 15
```

(13) Find the factorial value of given input number.

Script:

```
#! /bin/bash

read -p "Enter Number: " num

re='[0-9]'

if ! [[ $num =~ $re ]]; then
    echo "Error : Not A Valid Number for Factorial" >&2
    exit 1
fi

fact=1

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

```
done

echo "Factorial of Given Number is : $fact"

# Limitation : It can Only Calculate Small Factorial

# For Solving the Limitation :

echo 'define f(x) {if (x>1){return x*f(x-1)};return 1} f(100)' | bc

# References
# 1.) https://stackoverflow.com/questions/3394580/how-do-you-find-the-factorial-of-a-number-in-a-bash-script
```

Output:

```
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q13.sh
Enter Number: 5
Factorial of Given Number is : 120
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q13.sh
Enter Number: 10
Factorial of Given Number is : 3628800
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q13.sh
Enter Number: 20
Factorial of Given Number is : 2432902008176640000
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q13.sh
Enter Number: 50
Factorial of Given Number is : -3258495067890909184
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q13.sh
30414093201713378043612608166064768844377641568960512000000000000
```

(14) Generate and display Fibonacci series.

Script:

```
#! /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 ""
# echo -n : this option is used to omit echoing trailing newline .
```

Output:

```
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q14.sh
Enter number of values to show in Fibonnaci series: 10
The Fibonacci series is :
0 1 1 2 3 5 8 13 21 34
```

(15) Display all even numbers within given range.

Script:

```
#! /bin/bash

read -p "Enter Lower Limit of Range : " first
read -p "Enter Upper Limit of Range : " second

# Approach 1 : Brute Force

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

# Approach 2 : Since Modulus is Costly Operation

rem2=$((first % 2))

if [ "$rem2" -eq "1" ]; then
    ((first++))
fi

for ((i = $first; i <= $second; i+=2));
do
    echo $i
done
```

Output:

```
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q15.sh
Enter Lower Limit of Range : 19
Enter Upper Limit of Range : 32
20
22
24
26
28
30
32
```

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

Script:

```
#! /bin/bash
# Using wc command the number of words, characters and number of lines can be determined.
if [ -f "$1" ]; then
    words=$(cat $1 | wc -w)
    chars=$(cat $1 | wc -c)
    line=$(cat $1 | wc --lines)
    echo Number of Characters in $1 is $chars
    echo Number of Words in $1 is $words
    echo Number of Lines in $1 is $line
else
    echo "Error : $1 File Does Not Exist!"
fi
```

Output:

```
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ ./Q16.sh content.txt
Number of Characters in content.txt is 377
Number of Words in content.txt is 70
Number of Lines in content.txt is 9
bhagya@LAPTOP-1723NVO9:/mnt/c/Users/Admin/Desktop/OS_LAB_2/Scripts$ cat content.txt
It always seems impossible until it's done - Nelson Mandela

Start where you are. Use what you have. Do what you can - Arthur Ashe

Ever tried. Ever failed. No matter. Try Again. Fail again. Fail better -  Samuel Beckett

Don't watch the clock; do what it does. Keep going - Sam Levenson

Life is 10% what happens to you and 90% how you react to it - Charles Swindoll
```

SUBMITTED BY:

U19CS012

BHAGYA VINOD RANA