

Q-1 Write a Bash script that uses a for loop to iterate over a range of numbers from 1 to 20 and prints only the odd numbers.

```
echo "Printing odd numbers from 1-20"
for i in {1..20}
do
    if (( i % 2 != 0 ))
    then
        echo $i
    fi
done
```

Output:
Printing odd numbers from 1-20
1
3
5
7
9
11
13
15
17
19

Q-2 Write a shell script to print 1 to 20 numbers in reverse order and also calculate the sum of odd numbers.

```
echo "printing numbers from 20-1 in reverse order"
for (( i=20; i>=1; i-- ))
do
    echo $i
done

echo "sum of odd numbers"
sum=0
for i in {1..20}
do
    if (( i % 2 != 0 ))
    then
        sum=$((sum + i))
    fi
done
echo $sum
```

Output:
printing numbers from 20-1 in reverse order
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3

```

2
1
sum of odd numbers
100

```

Q-3 Write a script to count total number of vowels from inputted string. (Upper and Lower Both)

```

echo "Enter String : "
read str
count=0
for (( i=0; i<${#str}; i++ ))
do
    ch=${str:$i:1}
    case $ch in
        [aeiouAEIOU]) ((count++)) ;;
    esac
done
echo "Total Vowels : $count"

```

```

Output:
Enter String :
Amit
Total Vowels : 3

```

Q-4 Write a shell script to convert lowercase to uppercase and uppercase to lowercase by using menu.

```

while true
do
    echo
    echo "----Menu----"
    echo "1] Convert to upper case"
    echo "2] Convert to lower case"
    echo "3] Exit"
    echo "-----"
    read -p "Enter your Choice: " choice

    case $choice in
        1)
            read -p "Enter Text: " text
            echo
            echo "Result: $(echo "$text" | tr 'a-z' 'A-Z')"
            ;;
        2)
            read -p "Enter Text: " text
            echo
            echo "Result: $(echo "$text" | tr 'A-Z' 'a-z')"
            ;;
        3)
            echo "Exiting..."
            exit
            ;;
        *)
            echo "Invalid choice, try again."
            ;;
    esac
done

```

```

Output:
----Menu----
1] Convert to upper case
2] Convert to lower case
3] Exit
-----
Enter your Choice: 1

```

Enter Text: amit

Result: AMIT

Q-5 Write a shell script to check inputted number is palindrome or not.

```
read -p "Enter number: " num
rev=0
temp=$num
while [ $num -gt 0 ]
do
    rem=$(( num % 10 ))
    rev=$(( rev * 10 + rem ))
    num=$(( num / 10 ))
done
if [ $temp -eq $rev ]
then
    echo "$temp is a palindrome number."
else
    echo "$temp is not a palindrome number."
fi
```

Output:
Enter number: 1001
1001 is a palindrome number.

Q-6 Write a shell script to check whether a number is between 10 and 20 (Using AND operator)

```
read -p "Enter number: " num
if [ $num -ge 10 -a $num -le 20 ]
then
    echo "$num is between 10 and 20"
else
    echo "$num is not between 10 and 20"
fi
```

Output:
Enter number: 11
11 is between 10 and 20
Enter number: 29
29 is not between 10 and 20

Q-7 Write a shell script to input a number and check whether a number is positive, negative or zero.

```
read -p "Enter number: " num
if [ $num -gt 0 ]
then
    echo "$num is positive."
elif [ $num -lt 0 ]
then
    echo "$num is negative."
else
    echo "zero."
fi
```

Output:
Enter number: -67
-67 is negative.

Q-8 Write a shell script to input a character and check inputted character is vowel or not. (for both Upper and lower using switch case)

```

while true
do
    echo
    echo "Enter String in lowercase : "
    echo "Enter String in uppercase : "
    echo "Exit"
    echo
    read -p "Enter choice : " choice
case $choice in
1)
    read -p "Enter String: " str
    count=0
    for ((i=0; i<${#str}; i++))
    do
        ch=${str:$i:1}
        case $ch in
        [aeiou]) ((count++));;
        esac
    done
    echo "Total vowels: $count"
    ;;
2)
    read -p "Enter String: " str
    count=0
    for((i=0; i<${#str}; i++))
    do
        ch=${str:$i:1}
        case $ch in
        [AEIOU]) ((count++));;
        esac
    done
    echo "Total vowels : $count"
    ;;
3)
    echo "Exiting..."
    exit
    ;;
*)
    echo "Invalid choice,try again."
    ;;
esac
done

```

Output:

```

Enter String in lowercase :
Enter String in uppercase :
Exit

```

```

Enter choice : 1
Enter String:amit
Total vowels: 3

```

Q-9 Write a shell script to input two strings and check both strings are matched or not.

```

read -p "String 1: " s1
read -p "String 2: " s2
if [ "$s1" = "$s2" ]
then
    echo "Both strings are matched."
else
    echo "Strings don't match."
fi

```

Output:

```

String 1: Amit
String 2:Amit
Both strings are matched.

```

Q-10 Write a shell script to input a number and print sum of up to that number from 1.

```

read -p "Enter number: " num
sum=0
c=1
while [ $c -le $num ]
do
    sum=$(( sum + c ))
    ((c++))
done
echo "Sum : $sum"

```

Output:

```

Enter number: 7
Sum : 28

```

Q-11 Write a script that checks if a file named "test.txt" exists in the current directory, and if it does, prints "File exists", otherwise prints "File does not exist".

```

read -p "Enter file name : " file
if [ -f $file ] #-f options checks if the file exists in current directory
then
    echo "File exists"
else
    echo "File doesn't exists"
fi

```

Output:

```

Enter file name : p2.sh
File exists
Enter file name : p21.sh
File doesn't exists

```

Q-12 Write a script that prompts the user to enter their age, and then checks if the age is greater than or equal to 18. If it is, print "You are an adult", otherwise print "You are a minor".

```

read -p "Enter Age : " age
if [ $age -ge 18 ]
then
    echo "You are an adult"
else
    echo "You are a minor"
fi

```

Output:

```

Enter Age : 20
You are an adult

```

Q-13 Write a script that checks if a given string is empty and prints a message accordingly.

```

read -p "Enter String : " str
l=${#str}
if [ $l -le 0 ]
then
    echo "String is empty."
else
    echo "String's length: $l"
fi

```

Output:

```

Enter String :
String is empty.

```

Q-14 Write a Bash script that checks if a given string is a palindrome and prints a message accordingly.

```

read -p "Enter String : " str
for((i=${#str}; i>=0; i--))
do
    ch=${str:$i:1}
    rev=$rev$ch
done
if [ $rev = $str ]

```

```

then
    echo "Given string is palindrome"
else
    echo " Given string is not palindrome"
fi

```

Output:

```

Enter String : mam
Given string is palindrome

```

Q-15 Write a Bash script that checks if a file named "abc.sh" is executable and if it is, runs the script, otherwise prints "Script is not executable".

```

read -p "Enter file you want to check: " file
if [ -x $file ]
then
    echo "Running $file ..."
    ./$file
else
    echo "Script is not executable"
fi

```

Output:

```

Enter file you want to check: p12.sh
Running p12.sh ...
Enter Age : 14
You are a minor

```

Q-16 . Write a Bash script that prompts the user to input a number and then uses a for loop to print the multiplication table of that number up to 10.

```

read -p "Enter number: " num
echo "--Multipliacion table of $num --"
for((i=1; i<=10; i++))
do
    ans=$((num * i ))
    echo "$num * $i = $ans "
    ((c++))
done

```

Output:

```

Enter number: 2
--Multipliacion table of 5 --
2 * 1 = 2
2* 2 = 4
2* 3 =6
2* 4 = 8
2* 5 = 10
2* 6 = 12
2* 7 = 14
2* 8 = 16
2* 9 = 18
2* 10 = 20

```

Q-17 Write a Bash script that utilizes a for loop to iterate through a list of files in the current directory and print each filename.

```

echo "All files in current directory : "
for file in *
do
    echo $file
done

```

Output:

```

All files in current directory :
p10.sh
p11.sh
p12.sh
p13.sh
p14.sh
p15.sh

```

p16.sh
p17.sh
p18.sh
p19.sh
p1.sh
p20.sh
p2.sh
p3.sh
p4.sh
p5.sh
p6.sh
p7.sh
p8.sh
p9.sh

Q-18 Write a Bash script that prompts the user to input a number and then uses a while loop to print the factorial of that number.

```
read -p "Enter number: " num
fact=1
i=1
while((i <= $num))
do
    fact=$(( fact * i ))
    ((i++))
done
echo "Factorial of $num : $fact"
```

Output:

```
Enter number: 5
Factorial of 5 : 120
```

Q-19 . Write a Bash script that utilizes a while loop to continuously generate a random number between 1 and 50 until it generates a number divisible by 3.

```
while true
do
    num=$(( RANDOM % 50 + 1 ))
    echo "$num"
    if(( $num % 3 == 0 ))
    then
        echo "Generated number divisible by 3: $num"
        break
    fi
done
#RANDOM → built-in Bash variable for random numbers.
#% 50 + 1 → makes it fall in range 1 to 50.
```

Output:

```
23
19
44
49
21
Generated number divisible by 3: 21
```

Q-20 . Write a Bash script that uses a for loop to iterate over a list of names and prints a greeting message for each name

```
names=("amit " "anshul" "hariom" "altaf")
for n in "${names[@]}" # iterates over all elements in the array.
do
    echo "Hello $n,Welcome!!!"
done
```

Output:

```
Hello amit,Welcome!!!
Hello anshul ,Welcome!!!
Hello hariom,Welcome!!!
Hello altaf,Welcome!!!
```

Practical Assignment – 2

1. WRITE A UNIX SHELL SCRIPT WHICH TAKES INPUT A FILENAME AND CHECKS IT TO SEE: IF IT IS A ORDINARY FILE AND IS READABLE, ITS CONTENTS SHOULD BE DISPLAYED.

```
echo "Enter filename:"
read fname

if [ -f "$fname" ] && [ -r "$fname" ]
then
    echo "File is ordinary and readable. Contents:"
    cat "$fname"
else
    echo "File is either not ordinary or not readable."
fi
```

2. WRITE A UNIX SHELL SCRIPT TO FIND THE SUM OF FIRST N NUMBERS WHERE N IS ACCEPTED FROM USER.

```
echo "Enter value of N:"
read n

sum=0
for ((i=1; i<=n; i++))
do
    sum=$((sum + i))
done

echo "Sum of first $n numbers = $sum"
```

3. WRITE A UNIX SHELL SCRIPT TO ACCEPT MARKS OF Unix, PHP, ASP AND CALCULATE THE AVERAGE.

```
echo "Enter marks of Unix:"
read u
echo "Enter marks of PHP:"
read p
echo "Enter marks of ASP:"
read a

total=$((u+p+a))
avg=$((total/3))

echo "Average = $avg"

if [ $avg -ge 60 ]; then
    echo "First Class"
elif [ $avg -ge 50 ]; then
    echo "Second Class"
elif [ $avg -ge 40 ]; then
    echo "Third Class"
else
    echo "FAIL"
fi
```

4. WRITE A UNIX SHELL SCRIPT TO ACCEPT +VE NUMBER N AND GENERATE SERIES 1,8,27,...N³

```
echo "Enter a positive number N:"
read n

for ((i=1; i<=n; i++))
```

```
do
    echo -n "${i*i*i} "
done
echo
```

5. WRITE A UNIX SHELL SCRIPT TO ACCEPT +VE NUMBER N AND GENERATE SERIES 1,4,9,...N²

```
echo "Enter a positive number N:"
read n

for ((i=1; i<=n; i++))
do
    echo -n "${i*i} "
done
echo
```

6. WRITE A UNIX SHELL SCRIPT TO FIND ARMSTRONG NUMBERS (1 TO 999).

```
for num in $(seq 1 999)
do
    sum=0
    temp=$num
    while [ $temp -gt 0 ]
    do
        digit=$((temp % 10))
        sum=$((sum + digit*digit*digit))
        temp=$((temp / 10))
    done

    if [ $sum -eq $num ]; then
        echo "$num"
    fi
done
```

7. WRITE A UNIX SHELL SCRIPT TO ACCEPT A 5 DIGIT NUMBER AND PRINT SUM OF DIGITS.

```
echo "Enter a 5-digit number:"
read num

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

echo "Sum of digits = $sum"
```

8. WRITE A SHELL SCRIPT TO COUNT NUMBER OF VOWELS IN FILE (IGNORE CASE).

```
echo "Enter filename:"
read fname

if [ -f "$fname" ]
then
    count=$(grep -o -i "[aeiou]" "$fname" | wc -l)
    echo "Number of vowels in $fname = $count"
else
```

```
    echo "File not found."
fi
```

9. WRITE A SHELL SCRIPT TO PRINT ALL PRIME NUMBERS FROM 1 TO N.

```
echo "Enter value of N:"
read n

for ((i=2; i<=n; i++))
do
    prime=1
    for ((j=2; j<=i/2; j++))
    do
        if [  $((i \% j)) -eq 0$  ]; then
            prime=0
            break
        fi
    done
    if [ $prime -eq 1 ]; then
        echo -n "$i "
    fi
done
echo
```

10. WRITE A PROGRAM TO DISPLAY DATE IN DIFFERENT FORMAT ALONG WITH TIME.

```
echo "Default format:  $$(date)$ "
echo "DD/MM/YYYY format:  $$(date +"\%d/\%m/\%Y")$ "
echo "MM-DD-YYYY format:  $$(date +"\%m-\%d-\%Y")$ "
echo "Time (HH:MM:SS):  $$(date +"\%H:\%M:\%S")$ "
```

11. WRITE A UNIX SHELL SCRIPT TO FIND SUM OF DIGITS AND CHECK THE NUMBER IS PALINDROME OR NOT.

```
echo "Enter a number:"
read num

# Sum of digits
sum=0
temp=$num
while [ $temp -gt 0 ]
do
    digit=$((temp % 10))
    sum=$((sum + digit))
    temp=$((temp / 10))
done
echo "Sum of digits = $sum"

# Palindrome check
rev=$(echo $num | rev)
if [ $num -eq $rev ]; then
    echo "$num is Palindrome"
else
    echo "$num is Not Palindrome"
fi
```

12. WRITE A SHELL SCRIPT TO CALCULATE THE GROSS SALARY.

```
echo "Enter basic salary:"
read basic

da=$((basic * 40 / 100))
```

```
hra=$((basic * 20 / 100))
pf=$((basic * 20 / 100))
gross=$((basic + da + hra - pf))
```

```
echo "Basic = $basic"
echo "DA = $da"
echo "HRA = $hra"
echo "PF = $pf"
echo "Gross Salary = $gross"
```

13. TAKE 4 NUMERIC VALUES FROM THE USER & FIND MAXIMUM VALUE.

```
echo "Enter 4 numbers:"
read a b c d
```

```
max=$a
if [ $b -gt $max ]; then max=$b; fi
if [ $c -gt $max ]; then max=$c; fi
if [ $d -gt $max ]; then max=$d; fi
```

```
echo "Maximum = $max"
```

14. WRITE A SHELL SCRIPT TO FIND THE SQUARE ROOT OF A NUMBER.

```
echo "Enter a number:"
read n
```

```
sqrt=$(echo "scale=2; sqrt($n)" | bc -l)
echo "Square root of $n = $sqrt"
```

15. WRITE A SHELL SCRIPT TO DISPLAY SQUARES FROM 1 TO N (CONTINUE AS LONG AS USER WISHES).

```
choice="y"
while [ "$choice" = "y" ]
do
    echo "Enter a number N:"
    read n
    for ((i=1; i<=n; i++))
    do
        echo "$i^2 = $((i*i))"
    done
    echo "Do you want to continue? (y/n)"
    read choice
done
```

16. WRITE A SHELL PROGRAM TO LIST THE ODD NUMBERS FROM 1 TO N.

```
echo "Enter value of N:"
read n
```

```
for ((i=1; i<=n; i+=2))
do
    echo -n "$i "
done
echo
```

17. WRITE A SHELL SCRIPT TO READ A NUMBER AND REVERSE IT.

```
echo "Enter a number:"
read num
```

```

rev=0
temp=$num
while [ $temp -gt 0 ]
do
    digit=$((temp % 10))
    rev=$((rev*10 + digit))
    temp=$((temp / 10))
done

echo "Reverse of $num = $rev"

```

18. WRITE SHELL SCRIPT TO INPUT 20 NUMBERS AND COUNT HOW MANY +VE AND -VE.

```

pos=0
neg=0

echo "Enter 20 numbers:"
for ((i=1; i<=20; i++))
do
    read num
    if [ $num -ge 0 ]; then
        pos=$((pos+1))
    else
        neg=$((neg+1))
    fi
done

echo "Positive numbers = $pos"
echo "Negative numbers = $neg"

```

19. WRITE A SHELL SCRIPT TO REVERSE ALL WORDS FROM A SENTENCE.

```

echo "Enter a sentence:"
read sentence

for word in $sentence
do
    echo -n "$(echo $word | rev) "
done
echo

```

20. WRITE A SHELL SCRIPT TO INPUT A STRING AND:

```

1. CONVERT TO UPPERCASE
2. CONVERT TO LOWERCASE
3. EXIT
choice=0
while [ $choice -ne 3 ]
do
    echo "Enter a string:"
    read str

    echo "1. Convert to UPPERCASE"
    echo "2. Convert to lowercase"
    echo "3. Exit"
    echo "Enter your choice:"
    read choice

    case $choice in

```

```

        1) echo "Uppercase: $(echo $str | tr 'a-z' 'A-Z')" ;;
        2) echo "Lowercase: $(echo $str | tr 'A-Z' 'a-z')" ;;
        3) echo "Exiting..." ;;
        *) echo "Invalid choice" ;;
    esac
done

```

21. Write a shell script to display content of all the regular files of current directory.

```

for file in *
do
    if [ -f "$file" ]; then
        echo "Contents of $file:"
        cat "$file"
        echo "-----"
    fi
done

```

22. Write grep command for the following:

a) Display lines that have a 5 letters string that starts with 'T' and ends with 'e'.

```
grep -E '\bT...e\b' filename
```

b) Display lines that have numbers only.

```
grep -E '[0-9]+$' filename
```

c) Display lines that start with the character a.

```
grep '^a' filename
```

23. Do the following commands.

a) Link three files

```
ln file1 file2
```

```
ln file1 file3
```

b) List attributes of the current directory

```
ls -l
```

c) To print the files in the current directory and one level down

```
ls
```

```
ls */
```

d) Display inodes of all files of current directory

```
ls -li
```

e) To locate all the lines where 1st and last characters are same

```
grep -E '^(.)*\1$' filename
```

24. Write a shell script to check the file is directory or a file.

```

if [ -z "$1" ]; then
    echo "Usage: $0 filename"
    exit 1
fi

```

```

if [ -f "$1" ]; then
    echo "$1 is a regular file."
elif [ -d "$1" ]; then
    echo "$1 is a directory."
else
    echo "$1 is neither a file nor a directory."
fi

```

25. Write a script to count the number of vowels in the entered string.

```

echo "Enter a string:"
read str

```

```
count=$(echo "$str" | grep -o -i "[aeiou]" | wc -l)
echo "Number of vowels = $count"
```

26. Write a shell script that accepts five numbers and displays the second highest number.

```
echo "Enter 5 numbers:"
read a b c d e

arr=($a $b $c $d $e)
sorted=$(printf "%s\n" "${arr[@]}" | sort -n)
second_highest=${sorted[3]}

echo "Second highest number = $second_highest"
```

27. Write a script to perform following operations using menu:

```
i. count number of vowels
ii. count digits
iii. count special characters
echo "Enter a string:"
read str

choice=0
while [ $choice -ne 4 ]
do
    echo "Menu:"
    echo "1. Count vowels"
    echo "2. Count digits"
    echo "3. Count special characters"
    echo "4. Exit"
    echo "Enter your choice:"
    read choice

    case $choice in
        1) echo "Vowels = $(echo "$str" | grep -o -i "[aeiou]" | wc -l)" ;;
        2) echo "Digits = $(echo "$str" | grep -o "[0-9]" | wc -l)" ;;
        3) echo "Special Characters = $(echo "$str" | grep -o "[^a-zA-Z0-9 ]" | wc -l)" ;;
        4) echo "Exiting..." ;;
        *) echo "Invalid choice" ;;
    esac
done
```