

bash -writeup

so we can type commands in the command line but often we need to repeat to perform a certain task that's when the role of scripts comes in place. So basically a script is a textfile that contains the commands that can be executed when called upon and the interpreter we are currently using is BASH, so just like a coffee machine dispensing coffee automatically.

when we execute a bash file the commands are automatically executed and saves a lot of time

echo \$SHELL- shows the current interpreter that is bash

## CREATING A SCRIPT IN BASH

so to execute a file we need a textfile so in order to create the text editor that I used is vim to create a new file use `vim filename.sh`

to enter something we have 2 modes there append and insert mode. Append mode allows to enter after the cursor whereas insert mode allows us to enter before the cursor. After entering to write the file press the `esc` button :w for writing the file and :q for saving the file. To execute the script we need to enter `./filename.sh` in the terminal

1.) Write a shell script to run the following operations by reading 2 numbers and one choice from the user:

- Addition
- Subtraction
- Multiplication
- Division
- Average

It should be a choice based program i.e. if the input is 1, Addition should be performed

so here we need to receive 3 variables from the user. 2 of them (a and b) is used for receiving 2 numbers and 1 for receiving the user's input.

after that we use the case operator we used case instead of if else because it would help us to make the code small and we do not need to give a break command because as soon as the condition is found it stops working.

then we use the bc command for computing the calculations

then we use esac to exit the structure if there is no input matching with the given condition

code

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

```
echo "Enter Two numbers : "      - input related statement s
read a
read b
```

```
echo "Enter Choice : "
echo "1. Addition"              - PRINTING THE OPTIONS THAT USER HAS
echo "2. Subtraction"
echo "3. Multiplication"
echo "4. Division"
```

```
read ch
```

```
case $ch in
```

```
1)res='echo $a + $b | bc'
```

```
::
```

- USING THE CASE TO CHECK WHETHER THE GIVEN PATTERN IS THERE

```

2)res='echo $a - $b | bc'
;;
3)res='echo $a \* $b | bc'
;;
4)res='echo "scale=2; $a / $b" | bc'
;;
esac
echo "Result : $res"

```

-TO TERMINATE THE PATTERN CHECKING IF NO PATTERN IS FOUND

2.)write a shell script to check whether a pgrm is a palindrome nor not

so for this pgrm we create a new file using the common procedure and with the help of if and else loop and a command rev in bash  
first we recieve the input from the user  
then we store it in the variable rl and in that variable itself we use the rev command what it does is it reverses the characters  
for eg if we type 456 and use rev it would be 654  
now we use the if loop if the entered num is equal to the variable in which rev is formed ,then it is palindrome  
and by using else if the conditon is violated we print the opposite statement

```

code
#!/bin/bash
echo "enter the number "
read n
rl=$(echo $n | rev)
if [ $n -eq $rl ]; then
    echo "Number is palindrome"
else
    echo "Number is not palindrome"
fi

```

3.) write a bash script and according to user input use rot13 decode and rot13 encode  
for this pgrm we create and give the neccesary permissons form the common procedure  
then basically rot13 is a kind of encryption that allow the text to be rotated within 13 spaces as defined by its name  
so here basically anjo becomes nawb as a is shifted 13 times and becomes n and restof he characters also undergoes this process and the vice versa also happens in decoding  
for the scripting part i have used if,elif ,else and also used the tr command  
i first recevied the choice from the user and acoording to that i would receive the text store that in a variable and inside that i would use the tr and encode the entered text

```

CODE
echo "enter the choice"
echo "encode"
echo "2 encode"
read p
if [ $p -eq 1 ] : then
    read c
    g=$(echo "$c" |tr 'A-Za-z' 'N-ZA-Mn-za -m')
    echo "enoded text ; $g "
elif [ $p -eq 2 ] ; then
    read l
    f=$(echo( "$l" \tr 'A-Za-z' N-ZA-Mn-za -m')

```

```
    echo "decoded text ; $f "  
else:  
    echo "pls enter a valid choice"  
fi
```

4.)write a function in bash to check whether a file present in the system

here also i made the script file and then named the function as k then we assign a variable storing the file path and using the if we check whether the file exist in the system

i also referred some yt channels as i was lacking clarity in the functions concept

COde

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

```
k() {  
    local path="$1"  
  
    if [[ -e "$path" ]]; then  
        echo "File '$path' exists."  
    else  
        echo "File '$path' does not exist."  
    fi  
}
```