-: Basic Shell Scripting :- [Bash]

A shell script is a text file that contains a sequence of commands for a UNIX-based operating system.
It is still just a list of commands that are executed sequentially.

NOTE: Please push all your code to your GIT Repository

Lets start with Hello, World!

NOTE: Add hello_world.sh file for below code and Add, Commit, Push the file to Git repo.

```
#!/usr/local/bin/bash
# Author : Bhaskar
# A Shell program to read username and greet him. [ 01_helloworld.sh ]
echo "Hello, World!"
echo "What is your name?"
read NAME
echo "Hello, $NAME"
```

-: Decision Making :-

NOTE: create a branch <your_first_name>_sh_decision_01 to add below scripts

if..else..fi:

```
#!/usr/local/bin/bash
read -p "Enter number : " n
if test $n -ge 0
then
        echo "$n is positive number."
else
        echo "$n number is negative number."
fi
```

case....esac :

```
#!/usr/local/bin/bash
echo "Enter a tool (jenkins, jira, git, sonarqube): "
read TOOL
case "$TOOL" in
   "jenkins") echo "jenkins is a CI/CD tool."
   ;;
   "jira") echo "jira is a bug reporting and agile tool."
   ;;
   "git") echo "git is one of source-control management tool."
   ;;
   "sonarqube") echo "sonarqube is code coverage and static analysis tool"
   ;;
    *) echo "invalid tool $TOOL. valid tools are: jenkins, git, jira,
sonarqube."
   ;;
esac
```

-: Loops :-

NOTE: create a branch <your_first_name>_sh_loops_01 to add below scripts

for:

```
#!/usr/local/bin/bash
for count in {1..10}
do
   echo "$count"
done
```

while:

```
#!/usr/local/bin/bash
count=1
while [ $count -lt 10 ]
do
   echo $count
   count=`expr $count + 1`
done
```

-: Command line arguments :-

NOTE: create a branch <your_first_name>_sh_cliargs_func_01 to add below scripts

```
#!/usr/local/bin/bash
# A Shell program to check and print cli arguments passed
# Usage: bash 06_cli_args.sh ab cd ef gh
# Try  :
#   bash 06_cli_args.sh
#   bash 06_cli_args.sh ab
#   bash 06_cli_args.sh ab cd
#   bash 06_cli_args.sh ab cd ef gh ij
#
if [ $# -eq 2 ]
then
     echo "Received required arguments"
else
     echo "At least 2 arguments must be passed at cli"
fi

echo "This scripts name is         : $0"
echo "The first argument is        : $1"
echo "The number of arguments is   : $#"
echo "The list of arguments passed : $@"
echo "The exit status of last cmd  : $?"
# NOTE: If CLI arguments are more than 9 then use ${} notation.
# Example : To print 10th cli arguments use ${10} as $10 does not work.
```

-: Functions or Subroutines :-

```
#!/usr/local/bin/bash
factorial ()
{
    num1=$1
    fsum=1
    while [ $num1 -gt 1 ]
    do
        fsum=`expr $fsum \* $num1`
        num1=`expr $num1 - 1`
    done
    return $fsum
}
factorial "5"
echo "The factorial of 5 is: $?"
```

-: Write shell code to print a given string in ascii sorted order, after removing duplicate characters :-

Example: abracadabra --> abcdr

```
echo "abracadabra" | grep -o . | sort -u | tr -d '\n'
```

NOTE: Use of '|' pipe and grouping of commands to complete bigger task.