Week 1: Assignment

Name: Mukesh Kumar

1. **Create a Bash script that installs at least 12 programs.**
2. **Create a Bash script that provides the status of the system.**
3. **Create a Bash script that removes at least 6 programs.**
4. **Create a Bash script that allows and if the condition.**

1. Create a Bash script that installs at least 12 programs.

```bash
#!/bin/bash
clear
    echo -e "Installing 12 programs\n"
    echo "\t\t\n[1] Installing Wireshark"
    sudo apt intall wireshark
    echo "[2] intalling tree"
    sudo apt intall tree
    echo "[3] intalling nmap"
    sudo apt intall nmap
    echo "[4] intalling vim"
    sudo apt intallvim
    echo "[5] intalling curl"
    sudo apt intall curl
    echo "[6] intalling lolcat"
    sudo apt intall lolcat
    echo "[7] intalling espeak"
    sudo apt intall espeak
    echo "[8] intalling pv"
    sudo apt intall pv
    echo "[9] intalling jq"
    sudo apt intall jq
    echo "[10] intalling dirsearch"
    sudo apt intall dirsearch
    echo "[11] intalling git"
    sudo apt intall git
    echo "[12] intalling metasploit"
    sudo apt intall metasploit
```

2. Create a Bash script that provides the status of the system

```bash
#!/bin/bash

clear
    echo -e "-------------------------------System
Information--------------------------$GREEN"
    echo -e "Hostname:\t\t"`hostname`
    echo -e "uptime:\t\t\t"`uptime | awk '{print $3,$4}' | sed 's/,//'`
    echo -e "Manufacturer:\t\t"`cat /sys/class/dmi/id/chassis_vendor`
    echo -e "Product Name:\t\t"`cat /sys/class/dmi/id/product_name`
    echo -e "Version:\t\t"`cat /sys/class/dmi/id/product_version`
    echo -e "Serial Number:\t\t"`cat /sys/class/dmi/id/product_serial`
    echo -e "Machine Type:\t\t"`vserver=$(lscpu | grep Hypervisor | wc
-l); if [ $vserver -gt 0 ]; then echo "VM"; else echo "Physical"; fi`
    echo -e "Operating System:\t"`hostnamectl | grep "Operating System" |
cut -d ' ' -f5-`
    echo -e "Kernel:\t\t\t"`uname -r`
    echo -e "Architecture:\t\t"`arch`
    echo -e "Processor Name:\t\t"`awk -F':' '/^model name/ {print $2}'
/proc/cpuinfo | uniq | sed -e 's/^[ \t]*//'`
    echo -e "Active User:\t\t"`w | cut -d ' ' -f1 | grep -v USER | xargs
-n1`
    echo -e "System Main IP:\t\t"`hostname -I`
    echo ""
    echo -e "$RESETBG------------------------------CPU/Memory
Usage----------------------------$GREEN"
    echo -e "Memory Usage:\t"`free | awk '/Mem/{printf("%.2f%"),
$3/$2*100}'`
    echo -e "Swap Usage:\t"`free | awk '/Swap/{printf("%.2f%"),
$3/$2*100}'`
    echo -e "CPU Usage:\t"`cat /proc/stat | awk '/cpu/{printf("%.2f%\n"),
($2+$4)*100/($2+$4+$5)}' | awk '{print $0}' | head -1`
    echo ""
    echo -e "$RESETBG----------------------------Disk Usage
>80%---------------------------$GREEN"
    df -Ph
    echo
"$RESETBG------------------------------------------------------------------
----------"
```

3. Create a Bash script that removes at least 6 programs

```bash
#!/bin/bash
clear
      echo -e "Installing 12 programs\n"
      echo "\t\t\n[1] Installing Wireshark"
      sudo apt intall wireshark
      echo "[2] intalling tree"
      sudo apt intall tree
      echo "[3] intalling nmap"
      sudo apt intall nmap
      echo "[4] intalling vim"
      sudo apt intallvim
      echo "[5] intalling curl"
      sudo apt intall curl
      echo "[6] intalling lolcat"
      sudo apt intall lolcat
      echo "[7] intalling espeak"
      sudo apt intall espeak
      echo "[8] intalling pv"
      sudo apt intall pv
      echo "[9] intalling jq"
      sudo apt intall jq
      echo "[10] intalling dirsearch"
      sudo apt intall dirsearch
      echo "[11] intalling git"
      sudo apt intall git
      echo "[12] intalling metasploit"
      sudo apt intall metasploit
```

4. Create a Bash script that allows and if the condition

```bash
#!/bin/bash
echo "Enter Two Numbers: "
    read n1 n2
    if [[ "$n1" -gt "$n2" ]]; then
        echo "First Number is Larger"
    elif [[ "$n1" -eq "$n2" ]]; then
        echo "Both Numbers are equal"
    elif [[ "$n2" -gt "$n1" ]]; then
            echo "Second Number is Larger"
            else
                echo "Error"
    fi
```

Consolidate Code all scripts merged in a single file and works well

```bash
#!/usr/bin/env bash

## Create a bash script that does the following while demonstrating use.
    # intalling Wireshark and demonstrate use.
    # intalling Gephi or any other data science tool and demonstrate use.
    # intalling Nmap and perform a network mapping with Gephi or any other
data science tool captured. Data will come from the initial Wireshark
capture.
    # intalling Vim and demonstrate use.
    # Create a Bash script that intallings at least 12 programs.
    # Create a Bash script that provides the status of the system.
    # Create a Bash script that removes at least 6 programs. - not
required, advanced completion
    # Create a Bash script that allows and if the condition. - not
required, advanced completion

##Colors
RED="$(printf '\033[31m')"  GREEN="$(printf '\033[32m')"  ORANGE="$(printf
'\033[33m')"  BLUE="$(printf '\033[34m')"
MAGENTA="$(printf '\033[35m')"  CYAN="$(printf '\033[36m')"
WHITE="$(printf '\033[37m')"  BLACK="$(printf '\033[30m')"
REDBG="$(printf '\033[41m')"  GREENBG="$(printf '\033[42m')"
ORANGEBG="$(printf '\033[43m')"  BLUEBG="$(printf '\033[44m')"
MAGENTABG="$(printf '\033[45m')"  CYANBG="$(printf '\033[46m')"
WHITEBG="$(printf '\033[47m')"  BLACKBG="$(printf '\033[40m')"
RESETBG="$(printf '\e[0m')"  BLINK="$(printf '\033[5;31m')"

installPrograms()
{
    clear
    echo -e "Installing 12 programs\n"
    echo "\t\t\n[1] Installing Wireshark"
    sudo apt intall wireshark
    echo "[2] intalling tree"
    sudo apt intall tree
    echo "[3] intalling nmap"
    sudo apt intall nmap
    echo "[4] intalling vim"
    sudo apt intallvim
    echo "[5] intalling curl"
    sudo apt intall curl
    echo "[6] intalling lolcat"
```

```bash
        sudo apt intall lolcat
        echo "[7] intalling espeak"
        sudo apt intall espeak
        echo "[8] intalling pv"
        sudo apt intall pv
        echo "[9] intalling jq"
        sudo apt intall jq
        echo "[10] intalling dirsearch"
        sudo apt intall dirsearch
        echo "[11] intalling git"
        sudo apt intall git
        echo "[12] intalling metasploit"
        sudo apt intall metasploit
}

getSysStatus()
{
        clear
        echo -e "------------------------------System
Information---------------------------$GREEN"
        echo -e "Hostname:\t\t"`hostname`
        echo -e "uptime:\t\t\t"`uptime | awk '{print $3,$4}' | sed 's/,//'`
        echo -e "Manufacturer:\t\t"`cat /sys/class/dmi/id/chassis_vendor`
        echo -e "Product Name:\t\t"`cat /sys/class/dmi/id/product_name`
        echo -e "Version:\t\t"`cat /sys/class/dmi/id/product_version`
        echo -e "Serial Number:\t\t"`cat /sys/class/dmi/id/product_serial`
        echo -e "Machine Type:\t\t"`vserver=$(lscpu | grep Hypervisor | wc
-l); if [ $vserver -gt 0 ]; then echo "VM"; else echo "Physical"; fi`
        echo -e "Operating System:\t"`hostnamectl | grep "Operating System" |
cut -d ' ' -f5-`
        echo -e "Kernel:\t\t\t"`uname -r`
        echo -e "Architecture:\t\t"`arch`
        echo -e "Processor Name:\t\t"`awk -F':' '/^model name/ {print $2}'
/proc/cpuinfo | uniq | sed -e 's/^[ \t]*//'`
        echo -e "Active User:\t\t"`w | cut -d ' ' -f1 | grep -v USER | xargs
-n1`
        echo -e "System Main IP:\t\t"`hostname -I`
        echo ""
        echo -e "$RESETBG----------------------------CPU/Memory
Usage---------------------------$GREEN"
        echo -e "Memory Usage:\t"`free | awk '/Mem/{printf("%.2f%"),
$3/$2*100}'`
        echo -e "Swap Usage:\t"`free | awk '/Swap/{printf("%.2f%"),
```

```bash
$3/$2*100}'`
        echo -e "CPU Usage:\t"`cat /proc/stat | awk '/cpu/{printf("%.2f%\n"),
($2+$4)*100/($2+$4+$5)}' |  awk '{print $0}' | head -1`
        echo ""
        echo -e "$RESETBG----------------------------Disk Usage
>80%----------------------------$GREEN"
        df -Ph
        echo
"$RESETBG-------------------------------------------------------------
----------"
}

removePrograms()
{
        clear
        echo -e "\t\t\nRemoving any 6 programs\n"
        echo "$GREEN [1] Removing Wireshark"
        sudo apt remove wireshark
        echo "[2] removing tree"
        sudo apt remove tree
        echo "[3] removing nmap"
        sudo apt remove nmap
        echo "[4] removing vim"
        sudo apt remove vim
        echo "[5] removing curl"
        sudo apt remove curl
        echo "[6] removing lolcat"
        sudo apt remove lolcat
        echo "[7] removing espeak"
        sudo apt remove espeak
        echo "[8] removing pv"
        sudo apt remove pv
        echo "[9] removing jq"
        sudo apt remove jq
        echo "[10] removing dirsearch"
        sudo apt remove dirsearch
        echo "[11] removing git"
        sudo apt remove git
        echo "[12] removing metasploit$RESETBG"
        sudo apt remove metasploit
}

getLargerno()
```

```bash
{
    echo "Enter Two Numbers: "
    read n1 n2
    if [[ "$n1" -gt "$n2" ]]; then
            echo "First Number is Larger"
    elif [[ "$n1" -eq "$n2" ]]; then
            echo "Both Numbers are equal"
    elif [[ "$n2" -gt "$n1" ]]; then
                echo "Second Number is Larger"
                else
                        echo "Error"
    fi
}

echo -e "\t\t\tHello this is Mukesh Kumar Submitting the assignment week1
OS Hardening"
echo "[1] Install 12 programms"
echo "[2] Get status of the system"
echo "[3] Remove at least 6 programs"
echo "[4] Using the if condition"
read -p "$GREEN > Choose anyone:$RESETBG " choice

if [[ "$choice" == "1" ]]; then
    installPrograms
elif [[ "$choice" == "2" ]]; then
    getSysStatus
    elif [[ "$choice" == "3" ]]; then
                removePrograms
            elif [[ "$choice" == "4" ]]; then
                        getLargerno
fi
```