

**Devin Young**

**Bowie State University**

**CTEC 435**

**Professor Anthony**

Complete all exercises within the 10 pdf files in a linux environment.

Screenshot each exercise working successfully in a word document with the chapter number as the header and exercise number as the sub-header.

Submit any written answer questions in the word document.

Submit files that were created within each exercise as

'Chapter#\_exercise#\_FirstInitial\_Lastname' ex - Chapter10\_exercise1\_A\_Adedoyin

## Chapter 10\_exercise1\_D\_Young

### Automating User Management on Servers

Creating a user on multiple servers maybe something that you would do on a daily basis as a sysadmin. It is a tedious a task. Write a bash script that automates the entire process.

```

GNU nano 7.2                                chapt10exercise1.sh
#!/bin/bash

# Check if the username is provided
if [ "$#" -ne 1 ]; then
    echo "Usage: $0 <username>"
    exit 1
fi

USERNAME=$1
SERVERS=("server1" "server2" "server3") # Replace with your server names or IPs

# Loop through each server and create the user
for server in "${SERVERS[@]"; do
    echo "Creating user '$USERNAME' on $server..."

    # Create user with default password and add to sudo group
    ssh "$server" "sudo useradd -m $USERNAME && echo '$USERNAME:password' | sudo chpasswd && sudo usermod -aG sudo $USERNAME"

    if [ $? -eq 0 ]; then
        echo "User '$USERNAME' created successfully on $server."
    else
        echo "Failed to create user '$USERNAME' on $server."
    fi
done

echo "User creation process completed."

```

^G Help    ^O Write Out    ^W Where Is    ^K Cut    ^T Execute    ^C Location    M-U Undo    M-A Set Mark  
 ^X Exit    ^R Read File    ^\ Replace    ^U Paste    ^J Justify    ^/\_ Go To Line    M-E Redo    M-6 Copy

```
devin-young-vb1@devin-young-vb1-VirtualBox:~$ sudo chmod +x chapt10exercise1.sh
devin-young-vb1@devin-young-vb1-VirtualBox:~$ sudo ./chapt10exercise1.sh
Usage: ./chapt10exercise1.sh <username>
devin-young-vb1@devin-young-vb1-VirtualBox:~$ sudo ./chapt10exercise1.sh John
Creating user 'John' on server1...
```

## Chapter 10\_exercise2\_D\_Young

Write an automated script to backup archives of your files on a separate server.

```
GNU nano 7.2                                chapt10exercise2.sh
#!/bin/bash
backup_dirs=("/etc" "/home" "/boot")
dest_dir="/backup"
dest_server="server1"
backup_date=$(date +%b-%d-%y)
echo "Starting backup of: ${backup_dirs[@]}"
for i in "${backup_dirs[@]"; do
sudo tar -Pczf /tmp/${i}-${backup_date}.tar.gz ${i}
if [ $? -eq 0 ]; then
echo "${i} backup succeeded."
else
echo "${i} backup failed."
fi
scp /tmp/${i}-${backup_date}.tar.gz $dest_server:$dest_dir
if [ $? -eq 0 ]; then
echo "${i} transfer succeeded."
else
echo "${i} transfer failed."
fi
done
sudo rm /tmp/*.gz echo "Backup is done."
```

[ Read 21 lines ]

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location	M-U Undo	M-A Set Mark
^X Exit	^R Read File	^_ Replace	^U Paste	^J Justify	^_ Go To Line	M-E Redo	M-6 Copy

```
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# nano chapt10exercise2.sh
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# nano chapt10exercise2.sh
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# ./chapt10exercise2.sh
Starting backup of: /etc /home /boot
/etc backup succeeded.
```

## Chapter 10\_exercise3\_D\_Young

Take backups based on exercise 2 every day at midnight on the separate backup server.

```

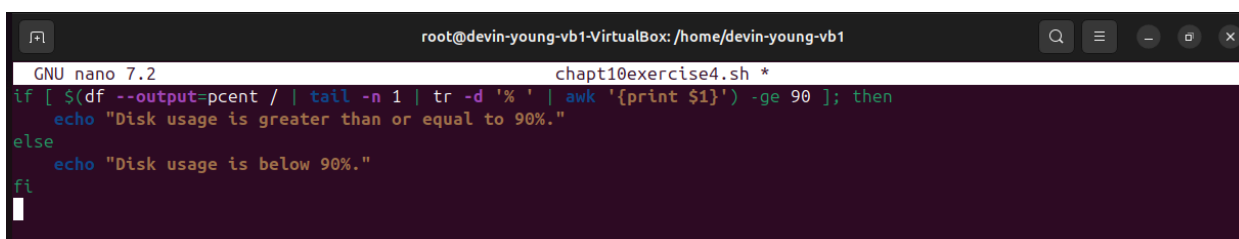
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# crontab -e
crontab: installing new crontab
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# crontab -l

0 0 * * * /path/to/your/chapt10exercise2.sh
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1#

```

## Chapter 10\_exercise4\_D\_Young

Use the commands `df`, `tail`, `awk` and `cut` to grab the disk usage percentage of your system and check to see if it's bigger than or equal to 90%.



```

GNU nano 7.2 chapt10exercise4.sh *
if [ $(df --output=pcent / | tail -n 1 | tr -d '%' | awk '{print $1}') -ge 90 ]; then
    echo "Disk usage is greater than or equal to 90%."
else
    echo "Disk usage is below 90%."
fi

```

```

root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# nano chapt10exercise4.sh
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1#
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# chmod +x chapt10exercise4.sh
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# ./chapt10exercise4.sh
Disk usage is below 90%.
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1#

```

## Chapter 10\_exercise5\_D\_Young

Check disk usage based on exercise 4 every six hours on your system

```

root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# nano chapt10exercise4.sh
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1#
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# chmod +x chapt10exercise4.sh
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# ./chapt10exercise4.sh
Disk usage is below 90%.
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# crontab -e
crontab: installing new crontab
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# crontab -l

0 0 * * * /path/to/your/chapt10exercise2.sh
0 */6 * * * /path/to/your/chapt10exercise4.sh

```

## Chapter 10\_exercise6\_D\_Young

Use Cron to take backups based on exercise 2 every day at midnight on the separate backup server. Also check disk usage based on exercise 4 every six hours on your system.

```
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# crontab -e
crontab: installing new crontab
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# crontab -l

0 0 * * * /path/to/your/chapt10exercise2.sh
0 */6 * * * /path/to/your/chapt10exercise4.sh
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1#
```

## Chapter 9\_exercise1\_D\_Young

Let's start with something simple. Write a bash shell script that checks if a given number is +ve or not. Use a function in your script.

```
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1
GNU nano 7.2 chapt9exercise1.sh *
#!/bin/bash

# Function to check if the number is positive
check_positive() {
    if [ "$1" -gt 0 ]; then
        echo "$1 is positive."
    elif [ "$1" -eq 0 ]; then
        echo "$1 is zero."
    else
        echo "$1 is negative."
    fi
}

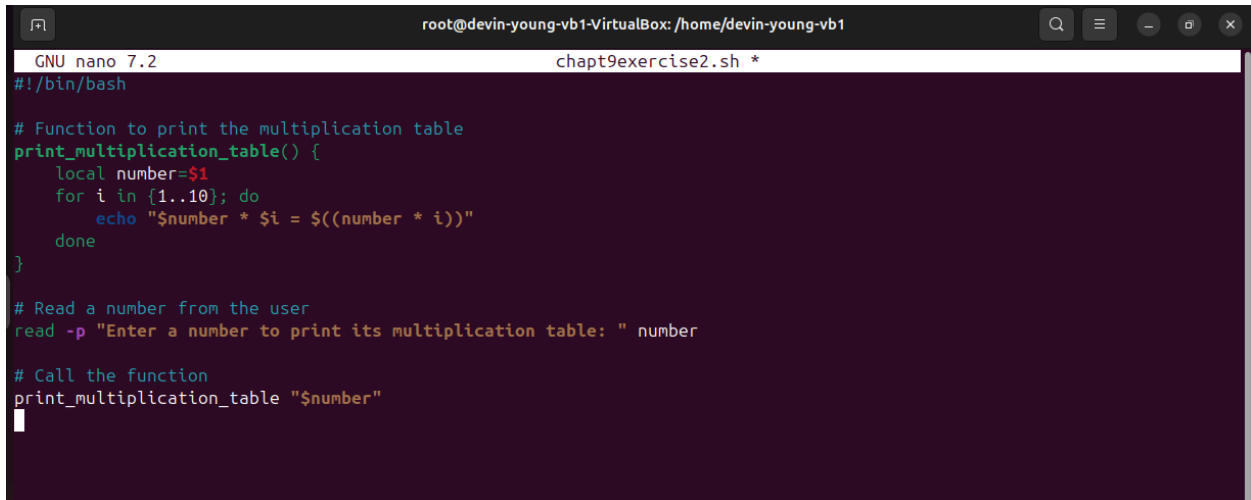
# Read a number from the user
read -p "Enter a number: " number

# Call the function
check_positive "$number"
```

```
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# nano chapt9exercise1.sh
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# ./chapt9exercise1.sh
bash: ./chapt9exercise1.sh: Permission denied
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# chmod +x chapt9exercise1.sh
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# ./chapt9exercise1.sh
Enter a number: 12
12 is positive.
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1#
```

## Chapter 9\_exercise2\_D\_Young

Write a bash shell script that prints the multiplication table of a given number. Using functions is a must.



```

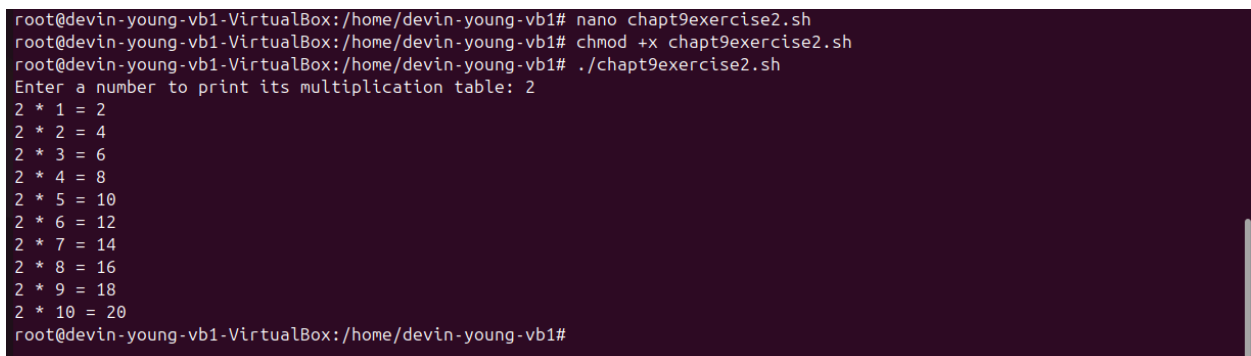
GNU nano 7.2 chapt9exercise2.sh *
#!/bin/bash

# Function to print the multiplication table
print_multiplication_table() {
    local number=$1
    for i in {1..10}; do
        echo "$number * $i = $((number * i))"
    done
}

# Read a number from the user
read -p "Enter a number to print its multiplication table: " number

# Call the function
print_multiplication_table "$number"

```



```

root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# nano chapt9exercise2.sh
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# chmod +x chapt9exercise2.sh
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# ./chapt9exercise2.sh
Enter a number to print its multiplication table: 2
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
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1#

```

## Chapter 9\_exercise3\_D\_Young

Write a shell script that accepts a number and check if it is prime or not. Using function is must here

```

root@devin-young-vb1-VirtualBox: /home/devin-young-vb1
GNU nano 7.2 chapt9exercise3.sh *
#!/bin/bash

# Function to check if a number is prime
is_prime() {
    local number=$1

    # Handle special cases
    if (( number <= 1 )); then
        echo "$number is not prime."
        return
    fi

    for ((i = 2; i * i <= number; i++)); do
        if (( number % i == 0 )); then
            echo "$number is not prime."
            return
        fi
    done

    echo "$number is prime."
}

# Read a number from the user
read -p "Enter a number to check if it is prime: " number

# Call the function
is_prime "$number"

```

^G Help    ^O Write Out    ^W Where Is    ^K Cut    ^T Execute    ^C Location    M-U Undo    M-A Set Mark  
 ^X Exit    ^R Read File    ^\ Replace    ^U Paste    ^J Justify    ^\_ Go To Line    M-E Redo    M-6 Copy

```

root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# nano chapt9exercise3.sh
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# chmod +x chapt9exercise3.sh
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# ./chapt9exercise3.sh
Enter a number to check if it is prime: 3
3 is prime.
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1#

```

## Chapter 9\_exercise4\_D\_Young

**Write a shell script that accepts a number greater than 3 and prints the Fibonacci sequence up to that term. DO NOT USE RECURSIVE FUNCTION.**

```

root@devin-young-vb1-VirtualBox: /home/devin-young-vb1
GNU nano 7.2                                chapt9exercise4.sh *
done
echo
}

# Read a number from the user
read -p "Enter a number greater than 3: " number

# Check if the number is greater than 3
if (( number > 3 )); then
    # Call the function
    print_fibonacci "$number"
else
    echo "Please enter a number greater than 3."
fi

```

```

root@devin-young-vb1-VirtualBox: /home/devin-young-vb1# nano chapt9exercise4.sh
root@devin-young-vb1-VirtualBox: /home/devin-young-vb1# chmod +x chapt9exercise4.sh
root@devin-young-vb1-VirtualBox: /home/devin-young-vb1# ./chapt9exercise4.sh
Enter a number greater than 3: 5
Fibonacci sequence up to 5 terms:
0 1 1 2 3
root@devin-young-vb1-VirtualBox: /home/devin-young-vb1#

```

## Chapter 9\_exercise5\_D\_Young

Write a shell script that accepts a number greater than 3 and prints the Fibonacci sequence up to that term. But you **MUST USE RECURSIVE FUNCTION** here.

```

root@devin-young-vb1-VirtualBox: /home/devin-young-vb1
GNU nano 7.2                                chapt9exercise5.sh *
done
echo
}

# Read a number from the user
read -p "Enter a number greater than 3: " number

# Check if the number is greater than 3
if (( number > 3 )); then
    # Call the function
    print_fibonacci "$number"
else
    echo "Please enter a number greater than 3."
fi

```



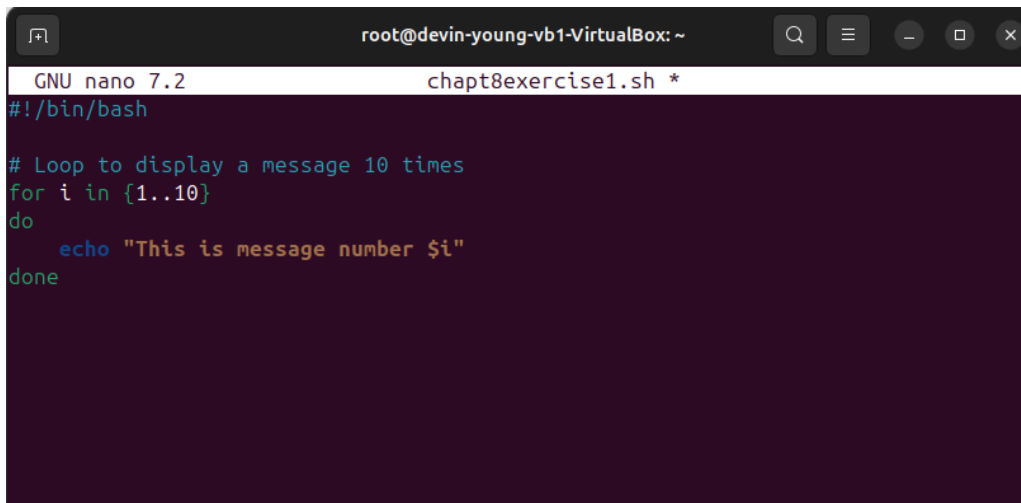
```

root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# nano chapt9exercise5.sh
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# chmod +x chapt9exercise5.sh
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# ./chapter9exercise5.sh
bash: ./chapter9exercise5.sh: No such file or directory
root@devin-young-vb1-VirtualBox:/home/devin-young-vb1# ./chapt9exercise5.sh
Enter a number greater than 3: 6
Fibonacci sequence up to 6 terms:
0 1 1 1 ./chapt9exercise5.sh: line 14: 1 + 1 1 : syntax error in expression (error token is "1 ")
1 1 ./chapt9exercise5.sh: line 14: 1 + 1 1 : syntax error in expression (error token is "1 ")
1 1 ./chapt9exercise5.sh: line 14: 1 + 1 1 : syntax error in expression (error token is "1 ")
1 1

```

## Chapter 8\_exercise1\_D\_Young

Create a for loop to display a message 10 times on Bash. Execute it as a Bash script.

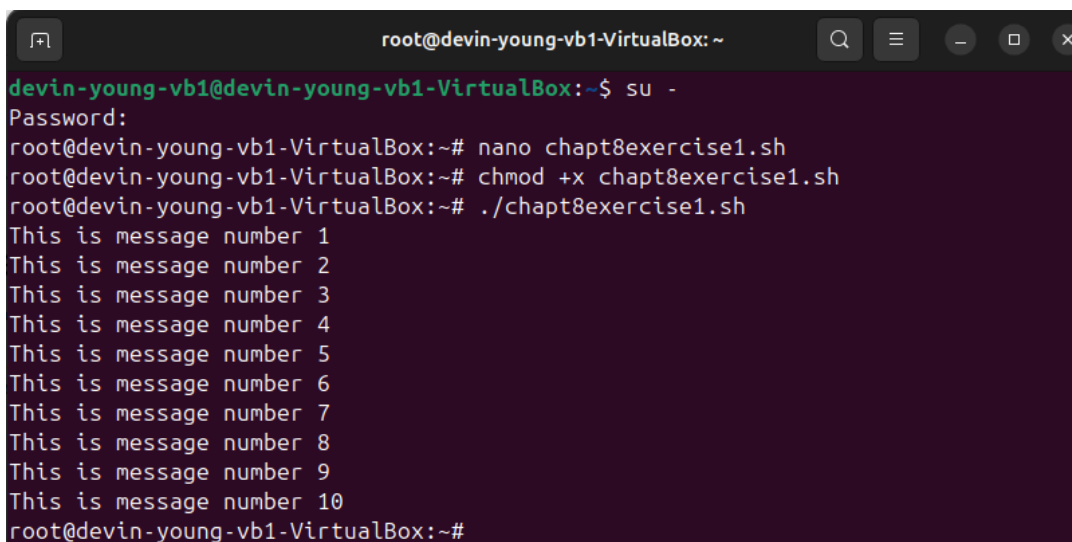


```

root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2 chapt8exercise1.sh *
#!/bin/bash

# Loop to display a message 10 times
for i in {1..10}
do
    echo "This is message number $i"
done

```



```

root@devin-young-vb1-VirtualBox: ~
devin-young-vb1@devin-young-vb1-VirtualBox:~$ su -
Password:
root@devin-young-vb1-VirtualBox:~# nano chapt8exercise1.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt8exercise1.sh
root@devin-young-vb1-VirtualBox:~# ./chapt8exercise1.sh
This is message number 1
This is message number 2
This is message number 3
This is message number 4
This is message number 5
This is message number 6
This is message number 7
This is message number 8
This is message number 9
This is message number 10
root@devin-young-vb1-VirtualBox:~#

```

## Chapter 8\_exercise2\_D\_Young

Output all the files and directory that exists under the /var directory.

The -a includes any hidden files as well.

```

root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2 chapt8exercise2.sh *
ls -al /var

```

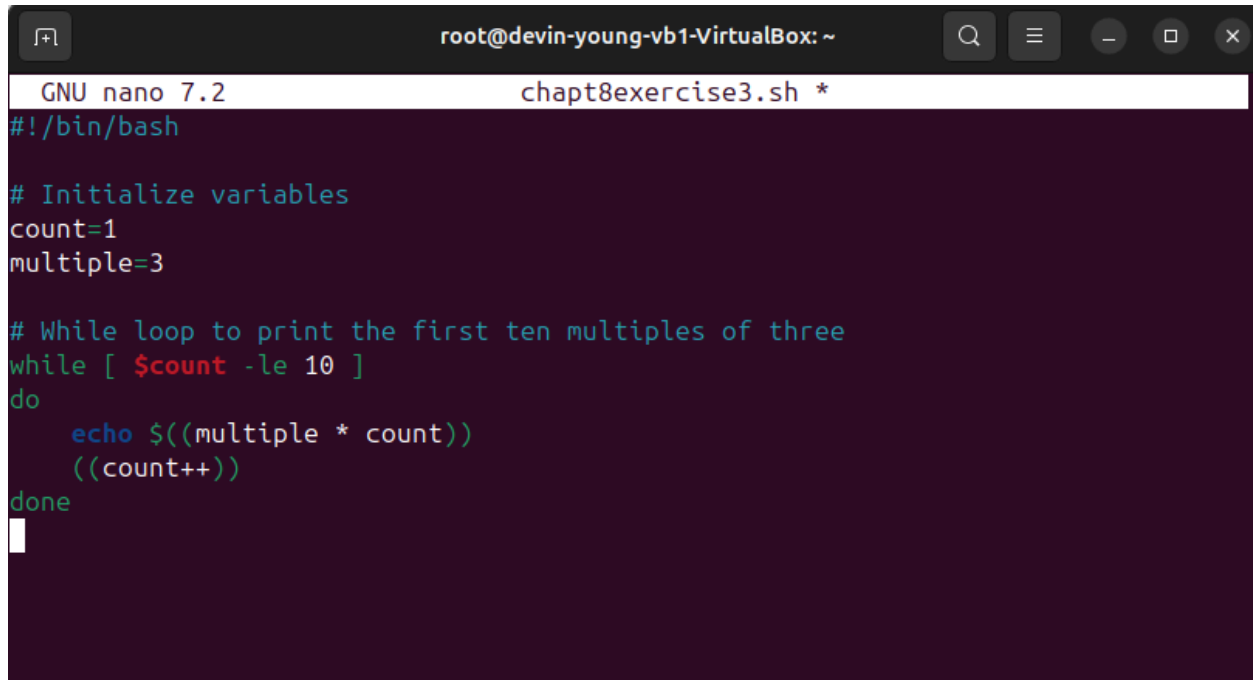
```

root@devin-young-vb1-VirtualBox:~# nano chapt8exercise2.sh
root@devin-young-vb1-VirtualBox:~#
root@devin-young-vb1-VirtualBox:~# chmod +x chapt8exercise2.sh
root@devin-young-vb1-VirtualBox:~# ./chapt8exercise2.sh
total 60
drwxr-xr-x 14 root root    4096 Sep  1 17:24 .
drwxr-xr-x 30 root root    4096 Sep 16 13:51 ..
drwxr-xr-x  2 root root    4096 Sep 25 00:00 backups
drwxr-xr-x 21 root root    4096 Sep  8 15:38 cache
drwxrwsrwt  2 root whoopsie 4096 Sep  9 04:04 crash
drwxr-xr-x 69 root root    4096 Sep  9 13:42 lib
drwxrwsr-x  2 root staff    4096 Apr 22 09:08 local
lrwxrwxrwx  1 root root         9 Aug 27 11:37 lock -> /run/lock
drwxrwxr-x 16 root syslog   4096 Sep 26 15:57 log
drwxrwsr-x  2 root mail     4096 Aug 27 11:37 mail
drwxrwsrwt  2 root whoopsie 4096 Aug 27 11:39 metrics
drwxr-xr-x  2 root root     4096 Aug 27 11:37 opt
lrwxrwxrwx  1 root root         4 Aug 27 11:37 run -> /run
drwxr-xr-x 11 root root     4096 Aug 27 11:42 snap
drwxr-xr-x  6 root root     4096 Aug 27 11:38 spool
drwxrwxrwt 13 root root     4096 Sep 26 16:08 tmp
-rw-r--r--  1 root root      208 Aug 27 11:37 .updated

```

## Chapter 8\_exercise3\_D\_Young

Use a while loop that will print the first ten multiples of the number three.



A screenshot of a terminal window titled 'root@devin-young-vb1-VirtualBox: ~'. The window shows the GNU nano 7.2 text editor editing a file named 'chapt8exercise3.sh'. The script content is as follows:

```
#!/bin/bash

# Initialize variables
count=1
multiple=3

# While loop to print the first ten multiples of three
while [ $count -le 10 ]
do
    echo $((multiple * count))
    ((count++))
done
```

The cursor is positioned at the end of the 'done' line.

```
root@devin-young-vb1-VirtualBox:~# nano chapt8exercise3.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt8exercise3.sh
root@devin-young-vb1-VirtualBox:~# ./chapt8exercise3.sh
3
6
9
12
15
18
21
24
27
30
root@devin-young-vb1-VirtualBox:~#
```

### Chapter 8\_exercise4\_D\_Young

Use an until loop that will print the first ten multiples of the number three.

```
root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2 chapt8exercise4.sh *
#!/bin/bash

# Initialize variables
count=1
multiple=3

# Until loop to print the first ten multiples of three
until [ $count -gt 10 ]
do
    echo $((multiple * count))
    ((count++))
done
```

```
root@devin-young-vb1-VirtualBox:~# nano chapt8exercise4.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt8exercise4.sh
root@devin-young-vb1-VirtualBox:~# ./chapt8exercise4.sh
3
6
9
12
15
18
21
24
27
30
root@devin-young-vb1-VirtualBox:~#
```

### Chapter 8\_exercise5\_D\_Young

Create an array that stores the first ten prime numbers. Iterate over the array and print out each element inside it.

```
root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2 chapt8exercise5.sh *
#!/bin/bash

# Create an array with the first ten prime numbers
primes=(2 3 5 7 11 13 17 19 23 29)

# Iterate over the array and print each element
for prime in "${primes[@]}"
do
    echo "$prime"
done
```

```
root@devin-young-vb1-VirtualBox:~# nano chapt8exercise5.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt8exercise5.sh
root@devin-young-vb1-VirtualBox:~# ./chapt8exercise5.sh
2
3
5
7
11
13
17
19
23
29
root@devin-young-vb1-VirtualBox:~#
```

### Chapter 7\_exercise1\_D\_Young

Write a bash shell script that checks whether the provided number is +ve or -ve.

```

root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2 chapt7exercise1.sh *
#!/bin/bash

# Check if a number is provided as an argument
if [ $# -eq 0 ]; then
    echo "Please provide a number."
    exit 1
fi

# Read the input number
number=$1

# Check if the number is positive, negative, or zero
if [ $number -gt 0 ]; then
    echo "$number is positive."
elif [ $number -lt 0 ]; then
    echo "$number is negative."
else
    echo "$number is zero."
fi

```

```

root@devin-young-vb1-VirtualBox:~# nano chapt7exercise1.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt7exercise1.sh
root@devin-young-vb1-VirtualBox:~# ./chapt7exercise1.sh
Please provide a number.
root@devin-young-vb1-VirtualBox:~# 2
2: command not found
root@devin-young-vb1-VirtualBox:~# ./chapt7exercise1.sh 2
2 is positive.
root@devin-young-vb1-VirtualBox:~#

```

### Chapter 7\_exercise2\_D\_Young

Write a shell script that accepts a string and a substring from the user (use read command). Then check if the string contains that substring and print a message accordingly.

```
root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2 chapt7exercise2.sh *
#!/bin/bash

# Prompt the user for a string
echo "Please enter a string:"
read main_string

# Prompt the user for a substring
echo "Please enter a substring:"
read substring

# Check if the main string contains the substring
if [[ $main_string == *"$substring"* ]]; then
    echo "The string contains the substring."
else
    echo "The string does not contain the substring."
fi

root@devin-young-vb1-VirtualBox:~# nano chapt7exercise2.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt7exercise2.sh
root@devin-young-vb1-VirtualBox:~# ./chapt7exercise2.sh
Please enter a string:
Hello James
Please enter a substring:
dd
The string does not contain the substring.
root@devin-young-vb1-VirtualBox:~#
```

### Chapter 7\_exercise3\_D\_Young

Write a shell script that checks whether a file exists or not. For the simplicity, keep the full file path in the script directly.

```
root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2 chapt7exercise3.sh *
#!/bin/bash

# Specify the full file path
file_path="/path/to/your/home/root@devin-young-vb1-VirtualBox"

# Check if the file exists
if [ -e "$file_path" ]; then
    echo "The file '$file_path' exists."
else
    echo "The file '$file_path' does not exist."
fi
```

```
root@devin-young-vb1-VirtualBox:~# nano chapt7exercise3.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt7exercise3.sh
root@devin-young-vb1-VirtualBox:~# ./chapt7exercise3.sh
The file '/path/to/your/home/root@devin-young-vb1-VirtualBox' does not exist.
root@devin-young-vb1-VirtualBox:~#
```

### Chapter 7\_exercise4\_D\_Young

Write a shell script that prompts user to enter a string (if user wants) and then checks if the input string is empty or not.



```

root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2 chapt7exercise4.sh *
#!/bin/bash

# Prompt the user to enter a string
echo "Please enter a string (press Enter to skip):"
read input_string

# Check if the input string is empty
if [ -z "$input_string" ]; then
    echo "The input string is empty."
else
    echo "The input string is: '$input_string'"
fi

root@devin-young-vb1-VirtualBox:~# nano chapt7exercise4.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt7exercise4.sh
root@devin-young-vb1-VirtualBox:~# ./chapt7exercise4.sh
Please enter a string (press Enter to skip):
Hi
The input string is: 'Hi'
root@devin-young-vb1-VirtualBox:~#

```

### Chapter 7\_exercise5\_D\_Young

Write a shell script that accepts a positive integer as argument and checks if the number is odd or even and prints 'Number X is odd' or 'Number X is even' accordingly.

If there is no argument or more than two arguments or a negative number provided, display an error message: "You have to provide one non negative number".

```
root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2                                chapt7exercise5.sh *

# Get the input number
number=$1

# Check if the number is odd or even
if (( number % 2 == 0 )); then
    echo "Number $number is even."
else
    echo "Number $number is odd."
fi
```

```
root@devin-young-vb1-VirtualBox:~# nano chapt7exercise5.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt7exercise5.sh
root@devin-young-vb1-VirtualBox:~# ./chapt7exercise5.sh
You have to provide one non-negative number.
root@devin-young-vb1-VirtualBox:~# ./chapt7exercise5.sh 10
Number 10 is even.
root@devin-young-vb1-VirtualBox:~#
```

## Chapter 6\_exercise1\_D\_Young

Create a string variable named `distro` and initialize its value to “Rocky Linux is CoolRocky Linux is Cool”. Display the string.

```
root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2                                chapt6exercise1.sh *

#!/bin/bash

# Initialize the string variable
distro="Rocky Linux is CoolRocky Linux is Cool"

# Display the string
echo "$distro"
```

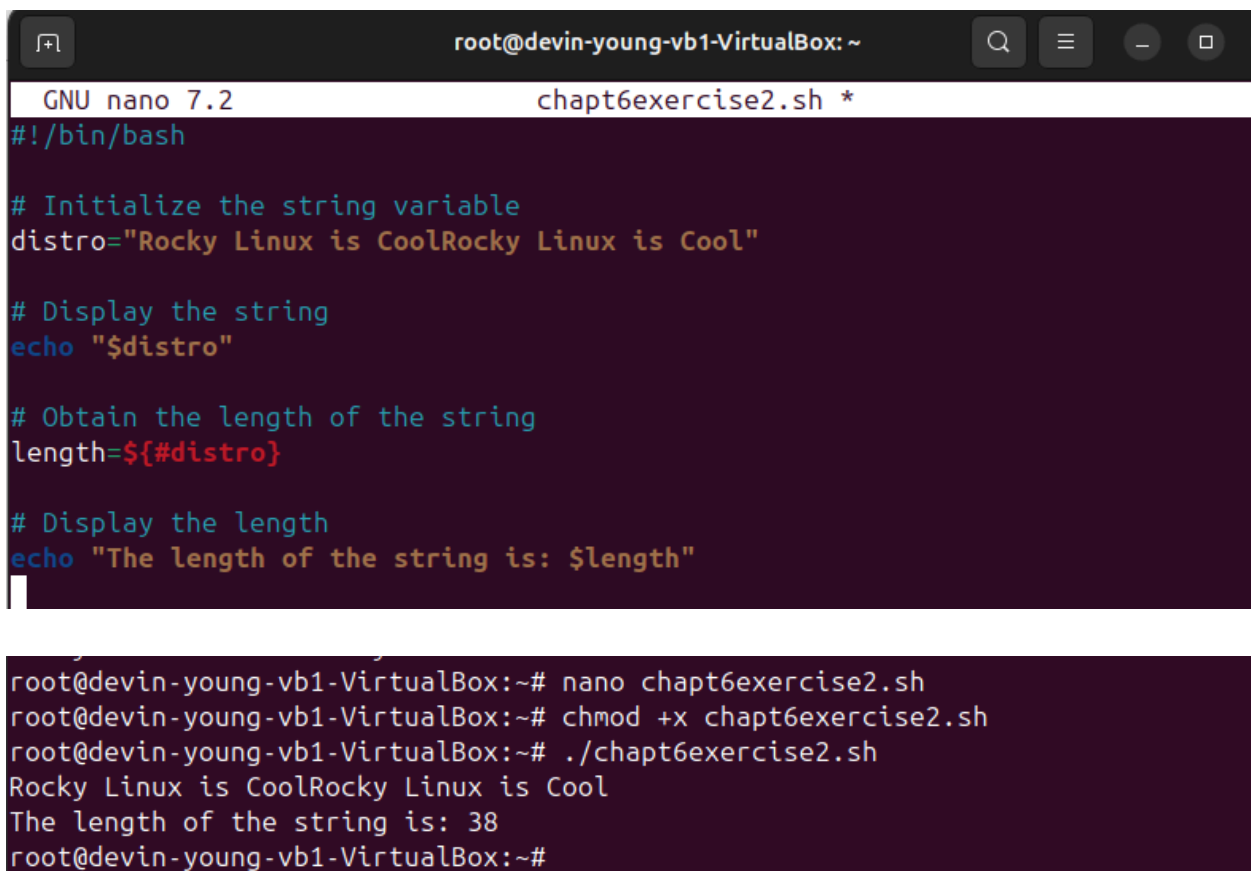
```

root@devin-young-vb1-VirtualBox:~# nano chapt6exercise1.sh
root@devin-young-vb1-VirtualBox:~# chmod chapt6exercise1.sh
chmod: missing operand after 'chapt6exercise1.sh'
Try 'chmod --help' for more information.
root@devin-young-vb1-VirtualBox:~# chmod +x chapt6exercise1.sh
root@devin-young-vb1-VirtualBox:~# ./chapt6exercise1.sh
Rocky Linux is CoolRocky Linux is Cool
root@devin-young-vb1-VirtualBox:~#

```

## Chapter 6\_exercise2\_D\_Young

Obtain the length of the distro string from exercise 1.



```

GNU nano 7.2 chapt6exercise2.sh *
#!/bin/bash

# Initialize the string variable
distro="Rocky Linux is CoolRocky Linux is Cool"

# Display the string
echo "$distro"

# Obtain the length of the string
length=${#distro}

# Display the length
echo "The length of the string is: $length"

root@devin-young-vb1-VirtualBox:~# nano chapt6exercise2.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt6exercise2.sh
root@devin-young-vb1-VirtualBox:~# ./chapt6exercise2.sh
Rocky Linux is CoolRocky Linux is Cool
The length of the string is: 38
root@devin-young-vb1-VirtualBox:~#

```

## Chapter 6\_exercise3\_D\_Young

Create a string variable called rockyrocky and initialize its value to “and Awesome!and Awesome!”. Now join strings distro and rockyrocky inside a third variable. Print it

```

root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2                                chapt6exercise3.sh *
#!/bin/bash

# Initialize the first string variable
distro="Rocky Linux is CoolRocky Linux is Cool"

# Initialize the second string variable
rocky="and Awesome!and Awesome!"

# Join the strings in a third variable
combined="${distro}${rocky}"

# Print the combined string
echo "$combined"

```

```

root@devin-young-vb1-VirtualBox:~# nano chapt6exercise3.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt6exercise3.sh
root@devin-young-vb1-VirtualBox:~# ./chapt6exercise3.sh
Rocky Linux is CoolRocky Linux is Cooland Awesome!and Awesome!
root@devin-young-vb1-VirtualBox:~#

```

## Chapter 6\_exercise4\_D\_Young

```

root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2                                chapt6exercise4.sh *

# Find the position of the first occurrence of "oo"
position=$(echo "$merge" | grep -b -o "oo" | head -n 1 | cut -d: -f1)

# Check if "oo" was found and print the position
if [ -n "$position" ]; then
    echo "The substring 'oo' first occurs at index: $position"
else
    echo "'oo' not found in the string."
fi

```

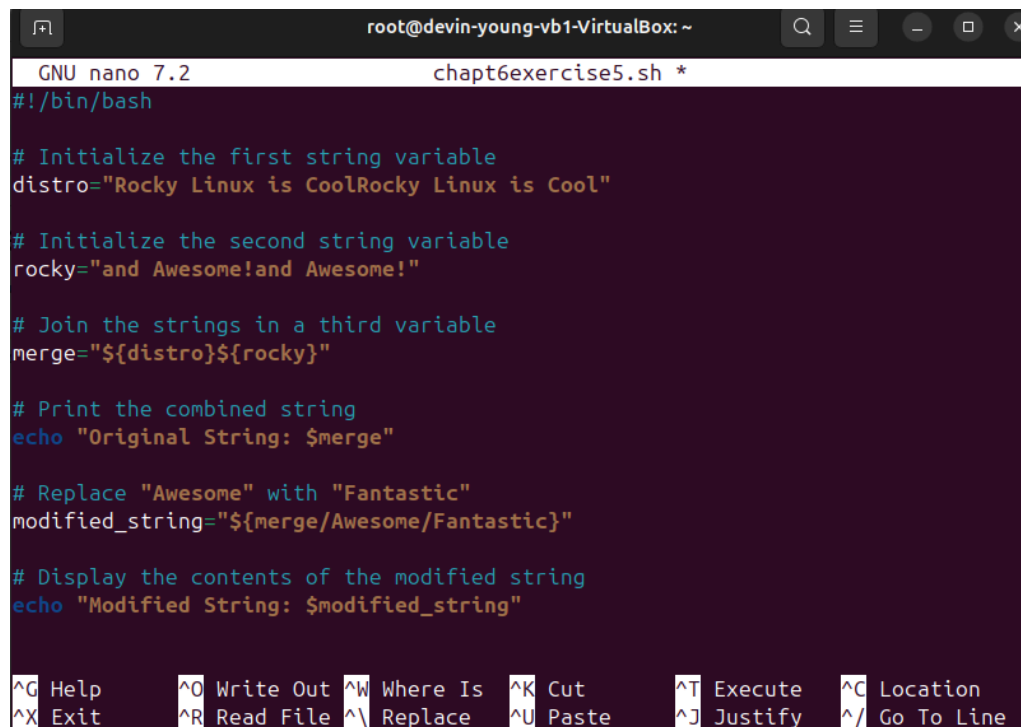
```

root@devin-young-vb1-VirtualBox:~# nano chapt6exercise4.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt6exercise4.sh
root@devin-young-vb1-VirtualBox:~# ./chapt6exercise4.sh
Combined String: Rocky Linux is CoolRocky Linux is Cooland Awesome!and Awesome!
The substring 'oo' first occurs at index: 16
root@devin-young-vb1-VirtualBox:~#

```

## Chapter 6\_exercise5\_D\_Young

Using the example from exercise 3, replace the word Awesome with Fantastic. Display the contents of the modified string.



```

root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2 chapt6exercise5.sh *
#!/bin/bash

# Initialize the first string variable
distro="Rocky Linux is CoolRocky Linux is Cool"

# Initialize the second string variable
rocky="and Awesome!and Awesome!"

# Join the strings in a third variable
merge="${distro}${rocky}"

# Print the combined string
echo "Original String: $merge"

# Replace "Awesome" with "Fantastic"
modified_string="${merge/Awesome/Fantastic}"

# Display the contents of the modified string
echo "Modified String: $modified_string"

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line

```

```

root@devin-young-vb1-VirtualBox:~# nano chapt6exercise5.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt6exercise5.sh
root@devin-young-vb1-VirtualBox:~# ./chapt6exercise5.sh
Original String: Rocky Linux is CoolRocky Linux is Cooland Awesome!and Awesome!
Modified String: Rocky Linux is CoolRocky Linux is Cooland Fantastic!and Awesome!
!
root@devin-young-vb1-VirtualBox:~#

```

## Chapter 5\_exercise1\_D\_Young

Write a shell script that accepts two integers and prints the quotient and remainder for their division.

```

root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2                                chapt5exercise1.sh *
#!/bin/bash

# Check if exactly two arguments are provided
if [ $# -ne 2 ]; then
    echo "You must provide exactly two integers."
    exit 1
fi

# Check if both arguments are integers
if ! [[ $1 =~ ^-[0-9]+$ ]] || ! [[ $2 =~ ^-[0-9]+$ ]]; then
    echo "Both arguments must be integers."
    exit 1
fi

# Assign arguments to variables
num1=$1
num2=$2

# Check for division by zero
if [ $num2 -eq 0 ]; then
    echo "Division by zero is not allowed."
    exit 1
fi

# Calculate quotient and remainder
quotient=$((num1 / num2))
remainder=$((num1 % num2))

```

```

# Check for division by zero
if [ $num2 -eq 0 ]; then
    echo "Division by zero is not allowed."
    exit 1
fi

# Calculate quotient and remainder
quotient=$((num1 / num2))
remainder=$((num1 % num2))

# Print the results
echo "Quotient: $quotient"
echo "Remainder: $remainder"

```

```

root@devin-young-vb1-VirtualBox:~# nano chapt5exercise1.sh
root@devin-young-vb1-VirtualBox:~# chmod chapt5exercise1.sh
chmod: missing operand after 'chapt5exercise1.sh'
Try 'chmod --help' for more information.
root@devin-young-vb1-VirtualBox:~# chmod +x chapt5exercise1.sh
root@devin-young-vb1-VirtualBox:~# ./chapt5exercise1.sh
You must provide exactly two integers.
root@devin-young-vb1-VirtualBox:~# ./chapt5exercise1.sh 10 2
Quotient: 5
Remainder: 0
root@devin-young-vb1-VirtualBox:~#

```

## Chapter 5\_exercise2\_D\_Young

Write a shell script that accepts an integer and prints its multiplication table.

```

root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2 chapt5exercise2.sh *
#!/bin/bash

# Check if an argument is provided
if [ $# -ne 1 ]; then
    echo "Please provide a single integer."
    exit 1
fi

# Check if the argument is an integer
if ! [[ $1 =~ ^-?[0-9]+$ ]]; then
    echo "The input must be an integer."
    exit 1
fi

# Assign the argument to a variable
number=$1

# Print the multiplication table
echo "Multiplication Table for $number:"
for i in {1..10}; do
    result=$((number * i))
    echo "$number x $i = $result"
done

```

```

root@devin-young-vb1-VirtualBox:~# nano chapt5exercise2.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt5exercise2.sh
root@devin-young-vb1-VirtualBox:~# ./chapt5exercise2.sh
Please provide a single integer.
root@devin-young-vb1-VirtualBox:~# ./chapt5exercise2.sh 10
Multiplication Table for 10:
10 x 1 = 10
10 x 2 = 20
10 x 3 = 30
10 x 4 = 40
10 x 5 = 50
10 x 6 = 60
10 x 7 = 70
10 x 8 = 80
10 x 9 = 90
10 x 10 = 100
root@devin-young-vb1-VirtualBox:~#

```

## Chapter 5\_exercise3\_D\_Young

Write a shell script that accepts the sides of the rectangle and prints its area and perimeter.

```

root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2                                chapt5exercise3.sh *
#!/bin/bash

# Check if exactly two arguments are provided
if [ $# -ne 2 ]; then
    echo "Please provide the length and width of the rectangle."
    exit 1
fi

# Check if both arguments are positive numbers
if ! [[ $1 =~ ^[0-9]+(\.[0-9]+)?$ ]] || ! [[ $2 =~ ^[0-9]+(\.[0-9]+)?$ ]]; then
    echo "Both arguments must be positive numbers."
    exit 1
fi

# Assign arguments to variables
length=$1
width=$2

# Calculate area and perimeter
area=$(echo "$length * $width" | bc)
perimeter=$(echo "2 * ($length + $width)" | bc)

# Print the results
echo "Area of the rectangle: $area"
echo "Perimeter of the rectangle: $perimeter"

root@devin-young-vb1-VirtualBox:~# nano chapt5exercise3.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt5exercise3.sh
root@devin-young-vb1-VirtualBox:~# ./chapt5exercise3.sh 5 10
Area of the rectangle: 50
Perimeter of the rectangle: 30
root@devin-young-vb1-VirtualBox:~#

```

## Chapter 5\_exercise4\_D\_Young

Write a shell script that takes the base and height as argument and prints its area.



```

root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2                                chapt5exercise4.sh *
#!/bin/bash

# Check if exactly two arguments are provided
if [ $# -ne 2 ]; then
    echo "Please provide the base and height of the triangle."
    exit 1
fi

# Check if both arguments are positive numbers
if ! [[ $1 =~ ^[0-9]+(\.[0-9]+)?$ ]] || ! [[ $2 =~ ^[0-9]+(\.[0-9]+)?$ ]]; then
    echo "Both arguments must be positive numbers."
    exit 1
fi

# Assign arguments to variables
base=$1
height=$2

# Calculate area
area=$(echo "0.5 * $base * $height" | bc)

# Print the result
echo "Area of the triangle: $area"

root@devin-young-vb1-VirtualBox:~# nano chapt5exercise4.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt5exercise4.sh
root@devin-young-vb1-VirtualBox:~# ./chapt5exercise4.sh 20 25
Area of the triangle: 250.0
root@devin-young-vb1-VirtualBox:~#

```

## Chapter 5\_exercise5\_D\_Young

Write a shell script that takes the three sides of a triangle as argument and prints its area using the Heron's formula.

```

root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2                                chapt5exercise5.sh *

# Assign arguments to variables
a=$1
b=$2
c=$3

# Calculate the semi-perimeter
s=$(echo "scale=2; ($a + $b + $c) / 2" | bc)

# Calculate the area using Heron's formula
area=$(echo "scale=2; sqrt($s * ($s - $a) * ($s - $b) * ($s - $c))" | bc -l)

# Print the result
echo "Area of the triangle: $area"

```

```

root@devin-young-vb1-VirtualBox:~# nano chapt5exercise5.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt5exercise5.sh
root@devin-young-vb1-VirtualBox:~# ./chapt5exercise5.sh 5 8 12
Area of the triangle: 14.52
root@devin-young-vb1-VirtualBox:~#

```

## Chapter 4\_exercise1\_D\_Young

Write a shell script in that uses an array to store the names of the seven colors of the rainbow and prints it.



```

root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2 chapt4exercise1.sh *
#!/bin/bash

# Define an array with the seven colors of the rainbow
colors=("Red" "Orange" "Yellow" "Green" "Blue" "Indigo" "Violet")

# Print the colors
echo "The colors of the rainbow are:"
for color in "${colors[@]}; do
    echo "$color"
done

```

```

root@devin-young-vb1-VirtualBox:~# nano chapt4exercise1.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt4exercise1.sh
root@devin-young-vb1-VirtualBox:~# ./chapt4exercise1.sh
The colors of the rainbow are:
Red
Orange
Yellow
Green
Blue
Indigo
Violet
root@devin-young-vb1-VirtualBox:~#

```

## Chapter 4\_exercise2\_D\_Young

Write a shell script in that asks you to pick a color from exercise 1 and print it.

```

GNU nano 7.2                                chapt4exercise2.sh *
#!/bin/bash

# Define an array with the seven colors of the rainbow
colors=("Red" "Orange" "Yellow" "Green" "Blue" "Indigo" "Violet")

# Print the available colors
echo "The colors of the rainbow are:"
for i in "${!colors[@]}"; do
    echo "${(i + 1)}. ${colors[$i]}"
done

# Ask the user to pick a color by number
read -p "Please pick a color by entering the corresponding number (1-7): " choice

# Validate the input
if [[ $choice -ge 1 && $choice -le 7 ]]; then
    # Print the selected color
    selected_color=${colors[$((choice - 1))]}
    echo "You picked: $selected_color"
else
    echo "Invalid choice. Please enter a number between 1 and 7."
fi

```

```

root@devin-young-vb1-VirtualBox:~# nano chapt4exercise2.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt4exercise2.sh
root@devin-young-vb1-VirtualBox:~# ./chapt4exercise2.sh
The colors of the rainbow are:
1. Red
2. Orange
3. Yellow
4. Green
5. Blue
6. Indigo
7. Violet
Please pick a color by entering the corresponding number (1-7): 4
You picked: Green
root@devin-young-vb1-VirtualBox:~#

```

## Chapter 4\_exercise3\_D\_Young

Write a shell script in that prints the number of colors of the rainbow from exercise 1.

```

GNU nano 7.2                                chapt4exercise3.sh *
#!/bin/bash

# Define an array with the seven colors of the rainbow
colors=("Red" "Orange" "Yellow" "Green" "Blue" "Indigo" "Violet")

# Calculate the number of colors
number_of_colors=${#colors[@]}

# Print the number of colors
echo "The number of colors in the rainbow is: $number_of_colors"

```

```

root@devin-young-vb1-VirtualBox:~# nano chapt4exercise3.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt4exercise3.sh
root@devin-young-vb1-VirtualBox:~# ./chapt4exercise3.sh
The number of colors in the rainbow is: 7
root@devin-young-vb1-VirtualBox:~#

```

## Chapter 4\_exercise4\_D\_Young

Write a shell script to swap any two colors of the rainbow from the current order as in exercise 1.

```

GNU nano 7.2 chapt4exercise4.sh *
#!/bin/bash

# Define an array with the seven colors of the rainbow
colors=("Red" "Orange" "Yellow" "Green" "Blue" "Indigo" "Violet")

# Print the available colors
echo "The colors of the rainbow are:"
for i in "${!colors[@]}; do
    echo "${(i + 1)). ${colors[$i]}"
done

# Ask the user to pick two colors by number
read -p "Enter the number of the first color to swap: " first_choice
read -p "Enter the number of the second color to swap: " second_choice

# Validate the input
if [[ $first_choice -ge 1 && $first_choice -le 7 ]] && [[ $second_choice -ge 1 && $second_choice -le 7 ]]; then
    # Adjust for zero-based indexing
    first_index=$((first_choice - 1))
    second_index=$((second_choice - 1))

    # Swap the colors
    temp=${colors[$first_index]}
    colors[$first_index]=${colors[$second_index]}
    colors[$second_index]=$temp

    # Print the updated colors
    echo "The colors after swapping are:"

    # Print the updated colors
    echo "The colors after swapping are:"
    for i in "${!colors[@]}; do
        echo "${(i + 1)). ${colors[$i]}"
    done
else
    echo "Invalid choice. Please enter numbers between 1 and 7."
fi

```

```

root@devin-young-vb1-VirtualBox:~# nano chapt4exercise4.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt4exercise4.sh
root@devin-young-vb1-VirtualBox:~# ./chapt4exercise4.sh
The colors of the rainbow are:
1. Red
2. Orange
3. Yellow
4. Green
5. Blue
6. Indigo
7. Violet
Enter the number of the first color to swap: 2
Enter the number of the second color to swap: 4
The colors after swapping are:
1. Red
2. Green
3. Yellow
4. Orange
5. Blue
6. Indigo
7. Violet
root@devin-young-vb1-VirtualBox:~# █

```

## Chapter 4\_exercise5\_D\_Young

Write a shell script to add an 8th color to the rainbow array described in exercise 1. Let's say cyan.

```

root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2 chapt4exercise5.sh *
#!/bin/bash

# Define an array with the seven colors of the rainbow
colors=("Red" "Orange" "Yellow" "Green" "Blue" "Indigo" "Violet")

# Print the original colors
echo "Original colors of the rainbow are:"
for color in "${colors[@]}; do
    echo "$color"
done

# Add an 8th color
colors+=("Cyan")

# Print the updated colors
echo "Updated colors of the rainbow are:"
for color in "${colors[@]}; do
    echo "$color"
done
█

```

```

root@devin-young-vb1-VirtualBox:~# nano chapt4exercise5.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt4exercise5.sh
root@devin-young-vb1-VirtualBox:~# ./chapt4exercise5.sh
Original colors of the rainbow are:
Red
Orange
Yellow
Green
Blue
Indigo
Violet
Updated colors of the rainbow are:
Red
Orange
Yellow
Green
Blue
Indigo
Violet
Cyan
root@devin-young-vb1-VirtualBox:~# █

```

### Chapter 3\_exercise1\_D\_Young

Write a shell script in that prints the name of the script.

```

root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2 chapt3exercise1.sh *
#!/bin/bash

# Print the name of the script
echo "The name of the script is: $0"
█

root@devin-young-vb1-VirtualBox:~# nano chapt3exercise1.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt3exercise1.sh
root@devin-young-vb1-VirtualBox:~# ./chapter3exercise1.sh
-bash: ./chapter3exercise1.sh: No such file or directory
root@devin-young-vb1-VirtualBox:~# ./chapt3exercise1.sh
The name of the script is: ./chapt3exercise1.sh
root@devin-young-vb1-VirtualBox:~#

```

### Chapter 3\_exercise2\_D\_Young

Write a shell script that takes 3 arguments and prints them in the reverse order.

```

root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2                                chapt3exercise2.sh *
#!/bin/bash

# Check if exactly three arguments are provided
if [ $# -ne 3 ]; then
    echo "Please provide exactly three arguments."
    exit 1
fi

# Print the arguments in reverse order
echo "Arguments in reverse order:"
echo "$3"
echo "$2"
echo "$1"

```

```

root@devin-young-vb1-VirtualBox:~# nano chapt3exercise2.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt3exercise2.sh
root@devin-young-vb1-VirtualBox:~# ./chapt3exercise2.sh bang chow pow
Arguments in reverse order:
pow
chow
bang
root@devin-young-vb1-VirtualBox:~#

```

### Chapter 3\_exercise3\_D\_Young

Write a shell script that accepts a directory name called hello and creates it if it doesn't exist already. Switch to this directory and create a new file named new.txt.

```

root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2                                chapt3exercise3.sh *
#!/bin/bash

# Define the directory name
dir_name="hello"

# Check if the directory exists
if [ ! -d "$dir_name" ]; then
    # Create the directory
    mkdir "$dir_name"
    echo "Directory '$dir_name' created."
else
    echo "Directory '$dir_name' already exists."
fi

# Switch to the directory
cd "$dir_name"

# Create a new file named new.txt
touch new.txt
echo "File 'new.txt' created in directory '$dir_name'."

```

```

root@devin-young-vb1-VirtualBox:~# nano chapt3exercise3.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt3exercise3.sh
root@devin-young-vb1-VirtualBox:~# ./chapt3exercise3.sh
Directory 'hello' created.
File 'new.txt' created in directory 'hello'.
root@devin-young-vb1-VirtualBox:~# █

```

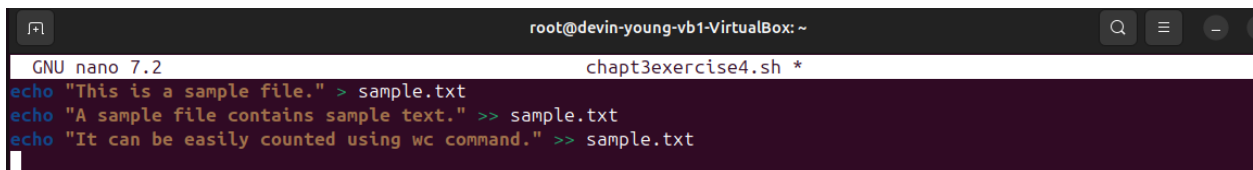
### Chapter 3\_exercise4\_D\_Young

Create a text file named sample.txt and add the following text to it:

This is a sample file.

A sample file contains sample text.

It can be easily counted using wc command.



```

root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2 chapt3exercise4.sh *
echo "This is a sample file." > sample.txt
echo "A sample file contains sample text." >> sample.txt
echo "It can be easily counted using wc command." >> sample.txt
█

```

```

root@devin-young-vb1-VirtualBox:~# nano chapt3exercise4.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt3exercise4.sh
root@devin-young-vb1-VirtualBox:~# ./chapt3exercise4.sh
root@devin-young-vb1-VirtualBox:~# cat sample.txt
This is a sample file.
A sample file contains sample text.
It can be easily counted using wc command.
root@devin-young-vb1-VirtualBox:~# █

```

### Chapter 3\_exercise5\_D\_Young

Write a shell script in that accepts a command as argument and gives the location of its binary file.



```

root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2                                chapt3exercise5.sh *
#!/bin/bash

# Check if a command is provided as an argument
if [ $# -ne 1 ]; then
    echo "Usage: $0 <command>"
    exit 1
fi

# Get the command from the argument
command=$1

# Find the location of the binary file
location=$(which "$command")

# Check if the command was found
if [ -z "$location" ]; then
    echo "Command '$command' not found."
else
    echo "The location of the command '$command' is: $location"
fi

```

```

root@devin-young-vb1-VirtualBox:~# nano chapt3exercise5.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt3exercise5.sh
root@devin-young-vb1-VirtualBox:~# ./chapt3exercise5.sh ls
The location of the command 'ls' is: /usr/bin/ls
root@devin-young-vb1-VirtualBox:~#

```

## Chapter 2\_exercise1\_D\_Young

Write a shell script in that asks for your full name and prints it.

```

root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2                                chapt2exercise1.sh *
#!/bin/bash

# Prompt the user for their full name
read -p "Please enter your full name: " full_name

# Print the full name
echo "Your full name is: $full_name"

```

```

root@devin-young-vb1-VirtualBox:~# nano chapt2exercise1.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt2exercise1.sh
root@devin-young-vb1-VirtualBox:~# ./chapt2exercise1.sh
Please enter your full name: Devin Young
Your full name is: Devin Young
root@devin-young-vb1-VirtualBox:~#

```

## Chapter 2\_exercise2\_D\_Young

Write a shell script in that asks for your age and prints it.

```

root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2                                chapt2exercise2.sh *
#!/bin/bash

# Prompt the user for their age
read -p "Please enter your age: " age

# Print the age
echo "Your age is: $age"

root@devin-young-vb1-VirtualBox:~# nano chapt2exercise2.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt2exercise2.sh
root@devin-young-vb1-VirtualBox:~# ./chapt2exercise2.sh
Please enter your age: 19
Your age is: 19
root@devin-young-vb1-VirtualBox:~#

```

## Chapter 2\_exercise3\_D\_Young

Write a command that prints the value of Pi. Demonstrate that it should not be changeable.

```

root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2                                chapt2exercise3.sh *
#!/bin/bash

# Print the value of Pi
pi=$(echo "scale=10; 4*a(1)" | bc -l)
echo "The value of Pi is: $pi"

# Attempt to change Pi
pi=3.14
echo "Attempting to change Pi: $pi"

# Print again to show it hasn't changed
echo "The value of Pi remains: $(echo "scale=10; 4*a(1)" | bc -l)"

root@devin-young-vb1-VirtualBox:~# nano chapt2exercise3.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt2exercise3.sh
root@devin-young-vb1-VirtualBox:~# ./chapt2exercise3.sh
The value of Pi is: 3.1415926532
Attempting to change Pi: 3.14
The value of Pi remains: 3.1415926532
root@devin-young-vb1-VirtualBox:~#

```

## Chapter 2\_exercise4\_D\_Young

Write a shell script in that stores the current date in a variable and prints it.

```

root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2 chapt2exercise4.sh *
#!/bin/bash

# Store the current date in a variable
current_date=$(date)

# Print the current date
echo "The current date is: $current_date"

```

```

root@devin-young-vb1-VirtualBox:~# nano chapt2exercise4.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt2exercise4.sh
root@devin-young-vb1-VirtualBox:~# ./chapt2exercise4.sh
The current date is: Fri Sep 27 01:43:53 AM EDT 2024
root@devin-young-vb1-VirtualBox:~#

```

## Chapter 2\_exercise5\_D\_Young

Write a shell script in that prints all the outputs for exercises 1, 2 and 4. Print all the answers in a separate line.

```

root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2 chapt2exercise5.sh *
#!/bin/bash

# Prompt for the user's name
read -p "What is your name? " name
echo "Hello $name"

# Prompt for the user's age
read -p "What is your age? " age
echo "So you are $age."

# Print the current date in the specified format
current_date=$(date +"%A %d %B %Y %I:%M:%S %p %Z")
echo "Today's date is $current_date"

```

```

root@devin-young-vb1-VirtualBox:~# nano chapt2exercise5.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt2exercise5.sh
root@devin-young-vb1-VirtualBox:~# ./chapt2exercise5.sh
What is your name? Devin Young
Hello Devin Young
What is your age? 20
So you are 20.
Today's date is Friday 27 September 2024 01:49:58 AM EDT
root@devin-young-vb1-VirtualBox:~#

```

## Chapter 1\_exercise1\_D\_Young

Write a shell script in that prints your user name.

```

root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2                                chapt1exercise1.sh *
#!/bin/bash

# Print the username
echo "Your username is: $USER"

root@devin-young-vb1-VirtualBox:~# nano chapt1exercise1.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt1exercise1.sh
root@devin-young-vb1-VirtualBox:~# ./chapt1exercise1.sh
Your username is: root
root@devin-young-vb1-VirtualBox:~#

```

## Chapter 1\_exercise2\_D\_Young

Write a shell script in that prints your current working directory.

```

root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2                                chapt1exercise2.sh *
#!/bin/bash

# Print the current working directory
echo "Your current working directory is: $(pwd)"

root@devin-young-vb1-VirtualBox:~# nano chapt1exercise2.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt1exercise2.sh
root@devin-young-vb1-VirtualBox:~# ./chapt1exercise2.sh
Your current working directory is: /root
root@devin-young-vb1-VirtualBox:~#

```

## Chapter 1\_exercise3\_D\_Young

Write a shell script in that prints the absolute path to your home directory.

```

root@devin-young-vb1-VirtualBox: ~
GNU nano 7.2                                chapt1exercise3.sh *
#!/bin/bash

# Print the absolute path to the home directory
echo "Your home directory is: $HOME"

root@devin-young-vb1-VirtualBox:~# nano chapt1exercise3.sh
root@devin-young-vb1-VirtualBox:~# chmod +x chapt1exercise3.sh
root@devin-young-vb1-VirtualBox:~# ./chapt1exercise3.sh
Your home directory is: /root
root@devin-young-vb1-VirtualBox:~#

```

## Chapter 1\_exercise4\_D\_Young

Write a shell script in that prints the default shell.

```
root@devin-young-vb1-VirtualBox: ~  
GNU nano 7.2 chapt1exercise4.sh *  
#!/bin/bash  
  
# Print the default shell  
echo "Your default shell is: $SHELL"
```

```
root@devin-young-vb1-VirtualBox:~# nano chapt1exercise4.sh  
root@devin-young-vb1-VirtualBox:~# chmod +x chapt1exercise4.sh  
root@devin-young-vb1-VirtualBox:~# ./chapt1exercise4.sh  
Your default shell is: /bin/bash  
root@devin-young-vb1-VirtualBox:~#
```

## Chapter 1\_exercise5\_D\_Young

Write a shell script in that prints all of the above four questions, i.e. your user name, current working directory, home directory and default shell. Print all the answers in a separate line.

```
root@devin-young-vb1-VirtualBox: ~  
GNU nano 7.2 chapt1exercise5.sh *  
#!/bin/bash  
  
# Print the username  
echo "Your username is: $USER"  
  
# Print the current working directory  
echo "Your current working directory is: $(pwd)"  
  
# Print the home directory  
echo "Your home directory is: $HOME"  
  
# Print the default shell  
echo "Your default shell is: $SHELL"
```

```
root@devin-young-vb1-VirtualBox:~# nano chapt1exercise5.sh  
root@devin-young-vb1-VirtualBox:~# chmod +x chapt1exercise5.sh  
root@devin-young-vb1-VirtualBox:~# ./chapt1exercise5.sh  
Your username is: root  
Your current working directory is: /root  
Your home directory is: /root  
Your default shell is: /bin/bash  
root@devin-young-vb1-VirtualBox:~#
```



