**NETWORKING LAB PROGRAMS**

**2. Write a Shell program to check the given number is even or odd.**

**Code:**

echo -n "Enter a number:"

read n

if [ `expr $n % 2` == 0 ]

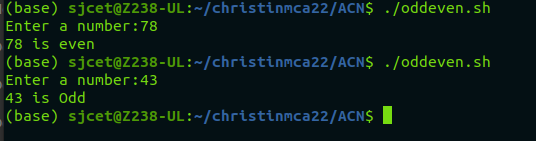
then

echo "$n is even"

else

echo "$n is Odd"

fi



**3. Write a Shell program to check a leap year.**

**Code:**

echo "Enter the year (YYYY)"

read year

if [ $((year % 4)) -eq 0 ]

then

if [ $((year % 100)) -eq 0 ]

then

if [ $((year % 400)) -eq 0 ]

then

echo "$year is a leap year"

else

echo "$year is not a leap year"

fi

else

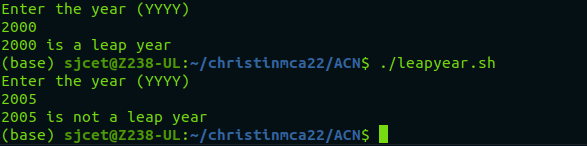
echo "$year is a leap year"

fi

else

echo "$year is not a leap year"

fi



**4. Write a Shell program to find the area and circumference of a circle.**

**Code:**

echo "Enter the radius:"

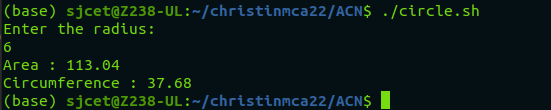
read r

area=`echo 3.14 \\* $r \\* $r| bc `

cir=`echo 2 \\* 3.14 \\* $r| bc `

echo "Area : $area"

echo "Circumference : $cir"



**5. Write a Shell program to check the given number and its reverse are same.**

**Code:**

echo "Enter a number: "

read num

reverse=$(echo "$num" | rev)

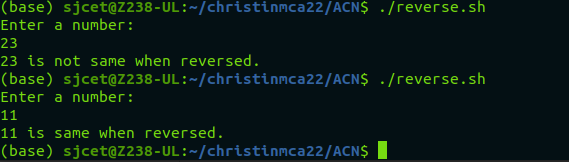
if [ "$num" -eq "$reverse" ]; then

echo "$num is same when reversed."

else

echo "$num is not same when reversed."

fi



**6. Write a Shell program to check the given string is palindrome or not.**

**Code:**

echo Enter the string

read s

echo $s>temp

rvs="$(rev temp)"

if [ $s = $rvs ]

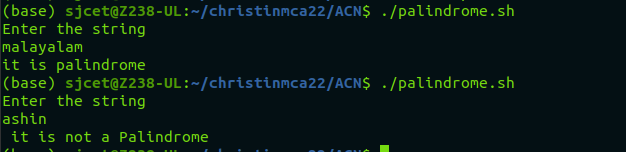
then

echo "it is palindrome"

else

echo " it is not a Palindrome"

fi



**7. Write a Shell program to find the sum of odd and even numbers from a set of numbers.**

**Code:**

echo "Enter a set of numbers (separated by spaces):"

read numbers

IFS=" " read -a array <<< "$numbers"

sum\_even=0

sum\_odd=0

for num in "${array[@]}"

do

if [ $((num % 2)) -eq 0 ]

then

sum\_even=$((sum\_even + num))

else

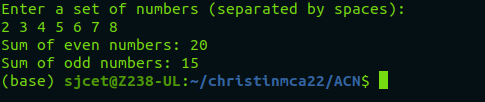
sum\_odd=$((sum\_odd + num))

fi

done

echo "Sum of even numbers: $sum\_even"

echo "Sum of odd numbers: $sum\_odd"



**8. Write a Shell program to find the roots of a quadratic equation**.

**Code:**

echo "Enter the coefficients of the quadratic equation (a, b, c): "

read a b c

discriminant=$((b\*b - 4\*a\*c))

if [ $discriminant -lt 0 ]

then

echo "The quadratic equation has no real roots."

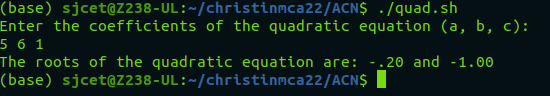
else

root1=$(echo "scale=2; (-$b + sqrt($discriminant)) / (2\*$a)" | bc)

root2=$(echo "scale=2; (-$b - sqrt($discriminant)) / (2\*$a)" | bc)

echo "The roots of the quadratic equation are: $root1 and $root2"

fi



**9. Write a Shell program to check the given integer is Armstrong number or not.**

**Code:**

echo "Enter an integer: "

read number

count=${#number}

sum=0

for (( i=0; i<count; i++ ))

do

digit=${number:i:1}

sum=$((sum + digit\*\*count))

done

if [ "$sum" -eq "$number" ]

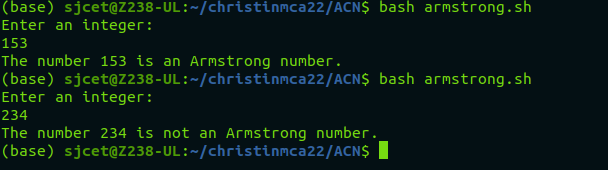
then

echo "The number $number is an Armstrong number."

else

echo "The number $number is not an Armstrong number."

Fi



**10. Write a Shell program to check the given integer is prime or not.**

**Code:**

echo "Enter an integer: "

read number

flag=1

for (( i=2; i<=number/2; i++ ))

do

if [ $((number%i)) -eq 0 ]

then

flag=0

break

fi

done

if [ $number -eq 1 ]

then

echo "1 is neither prime nor composite."

elif [ $flag -eq 1 ]

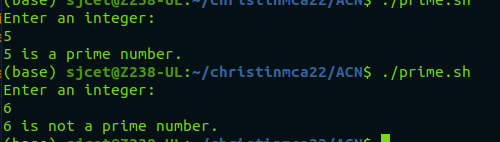
then

echo "$number is a prime number."

else

echo "$number is not a prime number."

fi



**11. Write a Shell program to generate prime numbers between 1 and 50.**

**Code:**

echo "Prime numbers between 1 and 50 are:"

for (( number=2; number<=50; number++ ))

do

flag=1

for (( i=2; i<=number/2; i++ ))

do

if [ $((number%i)) -eq 0 ]

then

flag=0

break

fi

done

if [ $flag -eq 1 ]

then

echo $number

fi

done



**12. Write a Shell program to find the sum of square of individual digits of a number.**

**Code:**

echo "Enter a number: "

read number

sum=0

while [ $number -ne 0 ]

do

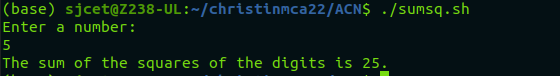
digit=$((number % 10))

sum=$((sum + digit \* digit))

number=$((number / 10))

done

echo "The sum of the squares of the digits is $sum."



**13. Write a Shell program to count the number of vowels in a line of text.**

**Code:**

echo "Enter a line of text: "

read line

count=0

for (( i=0; i<${#line}; i++ ))

do

char=${line:$i:1}

if [[ $char == [aeiouAEIOU] ]]

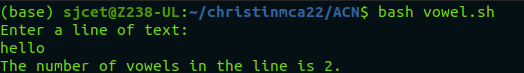
then

count=$((count + 1))

fi

done

echo "The number of vowels in the line is $count."



**14. Write a Shell program to display student grades.**

**Code:**

declare -A grades=(

[Alice]=90

[Bob]=80

[Charlie]=70

[David]=60

[Emma]=50

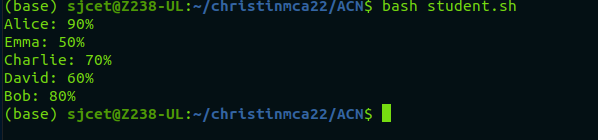
)

for name in "${!grades[@]}"

do

echo "$name: ${grades[$name]}%"

done

****

**15. Write a Shell program to find the smallest and largest numbers from a set of numbers.**

**Code:**

echo "Enter a list of numbers separated by spaces: "

read numbers

IFS=' ' read -ra nums <<< "$numbers"

min=${nums[0]}

max=${nums[0]}

for num in "${nums[@]}"

do

if (( num < min )); then

min=$num

fi

if (( num > max )); then

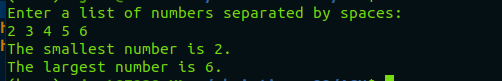
max=$num

fi

done

echo "The smallest number is $min."

echo "The largest number is $max."



**16. Write a Shell program to find the smallest digit from a number**

**Code:**

echo "Enter a number: "

read num

min=${num:0:1}

for (( i=1; i<${#num}; i++ ))

do

digit=${num:$i:1}

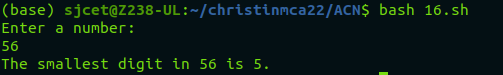
if (( digit < min )); then

min=$digit

fi

done

echo "The smallest digit in $num is $min."



**17. Write a Shell program to find the sum of all numbers between 50 and 100, which are**

**divisible by 3 and not divisible by 5.**

**Code:**

sum=0

for (( num=50; num<=100; num++ ))

do

if (( num % 3 == 0 && num % 5 != 0 )); then

sum=$((sum + num))

fi

done

echo "The sum of all numbers between 50 and 100, which are divisible by 3 and not divisible by 5, is $sum."



**18. Write a Shell program to find the second highest number from a set of numbers.**

**Code:**

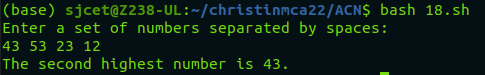
echo "Enter a set of numbers separated by spaces: "

read numbers

arr=($numbers)

sorted\_arr=($(echo "${arr[@]}" | tr " " "\n" | sort -rn))

echo "The second highest number is ${sorted\_arr[1]}."



**19. Write a Shell program to find the sum of digits of a number using function.**

**Code:**

sum\_of\_digits() {

num=$1

sum=0

while [ $num -gt 0 ]

do

digit=$((num % 10))

sum=$((sum + digit))

num=$((num / 10))

done

echo $sum

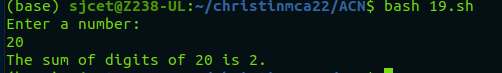
}

echo "Enter a number: "

read num

result=$(sum\_of\_digits $num)

echo "The sum of digits of $num is $result."



**20. Write a Shell program to print the reverse of a number using function.**

**Code:**

reverse\_number() {

num=$1

rev=0

while [ $num -gt 0 ]

do

digit=$((num % 10))

rev=$((rev \* 10 + digit))

num=$((num / 10))

done

echo $rev

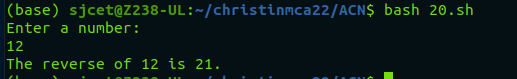
}

echo "Enter a number: "

read num

result=$(reverse\_number $num)

echo "The reverse of $num is $result."



**21. Write a Shell program to find the factorial of a number using for loop.**

**Code:**

echo "Enter a number: "

read num

factorial=1

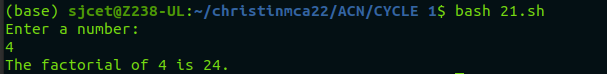
for (( i=1; i<=$num; i++ ))

do

factorial=$((factorial \* i))

done

echo "The factorial of $num is $factorial."



**22. Write a Shell program to generate Fibonacci series.**

**Code:**

echo "Enter the number of terms to generate: "

read num

a=0

b=1

echo -n "$a $b"

for (( i=3; i<=$num; i++ ))

do

c=$((a + b))

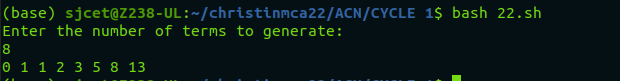
echo -n " $c"

a=$b

b=$c

Done

echo



**23. Write a shell script, which receives two filenames as arguments. It checks whether the two files contents are same or not. If they are same then second file is deleted.**

**Code:**

if [ $# -ne 2 ]; then

echo "Usage: $0 file1 file2"

exit 1

fi

if cmp -s "$1" "$2"; then

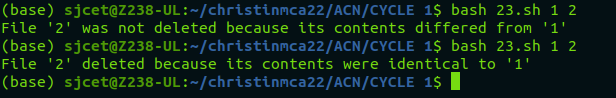
rm "$2"

echo "File '$2' deleted because its contents were identical to '$1'"

else

echo "File '$2' was not deleted because its contents differed from '$1'"

fi



**24. Write a Menu driven Shell script that Lists current directory, Prints Working Directory,**

**displays Date and displays Users logged in.**

**Code:**

while true; do

clear

echo "======================="

echo " MAIN MENU "

echo "======================="

echo "1. List current directory"

echo "2. Print working directory"

echo "3. Display date"

echo "4. Display users logged in"

echo "5. Exit"

echo -n "Enter your choice: "

read choice

case $choice in

1)

ls -la

echo "Press enter to continue"

read

;;

2)

pwd

echo "Press enter to continue"

read

;;

3)

date

echo "Press enter to continue"

read

;;

4)

who

echo "Press enter to continue"

read

;;

5)

echo "Exiting..."

exit 0

;;

\*)

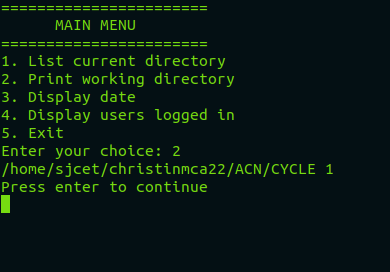
echo "Invalid choice. Press enter to continue"

read

;;

esac

done



**25. Shell script to check executable rights for all files in the current directory, if a file does not have the execute permission then make it executable.**

**Code:**

for file in \*; do

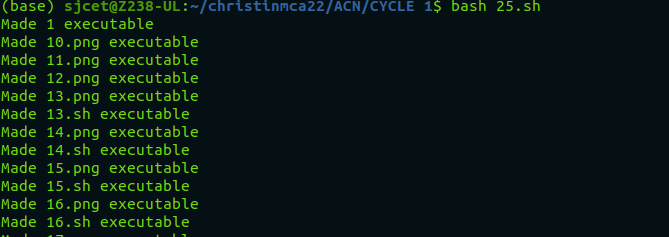
if [[ ! -x "$file" ]]; then

chmod +x "$file"

echo "Made $file executable"

fi

done



**26. Write a Shell program to generate all combinations of 1, 2, and 3 using loop.**

**Code:**

for i in 1 2 3; do

for j in 1 2 3; do

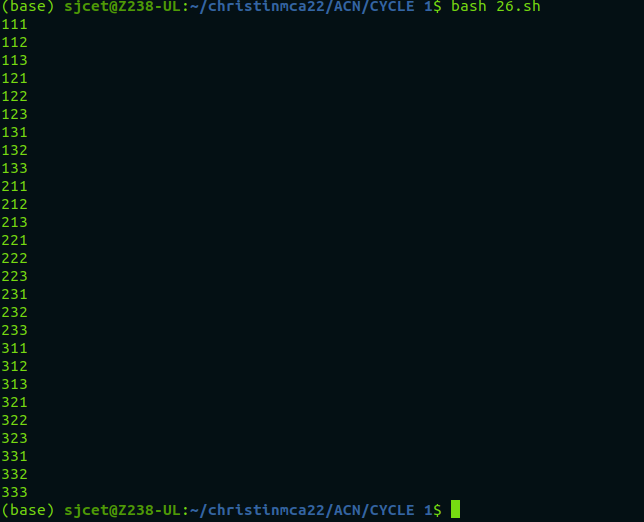
for k in 1 2 3; do

echo "$i$j$k"

done

done

don**e**

****

**27. Write a Shell program to create the number series.**

**1**

**2 3**

**4 5 6**

**7 8 9 10**

**Code:**

rows=4

current=1

for (( i=1; i<=rows; i++ ))

do

for (( j=1; j<=i; j++ ))

do

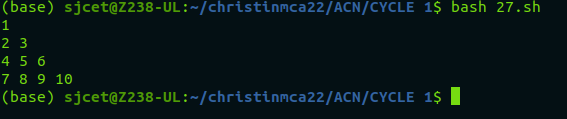
echo -n "$current "

(( current++ ))

done

echo

done

****

**28. Write a Shell program to create Pascal’s triangle.**

**Code:**

function binom {

if [ $2 -eq 0 ] || [ $2 -eq $1 ]; then

echo 1

else

echo $(( $(binom $(($1-1)) $(($2-1))) + $(binom $(($1-1)) $2) ))

fi

}

echo "Enter the number of rows in Pascal's triangle: "

read rows

for (( i=0; i<$rows; i++ )); do

for (( j=0; j<=$i; j++ )); do

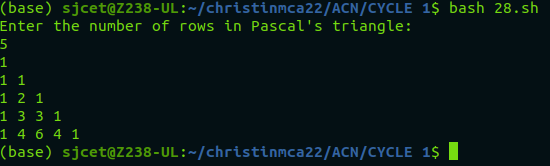
val=$(binom $i $j)

echo -n "$val "

done

echo ""

done

****

**29. Write a Decimal to Binary Conversion Shell Script.**

**Code:**

read -p "Enter decimal number: " decimal

binary=$(echo "obase=2;$decimal" | bc)

echo "Binary equivalent of $decimal is: $binary"

****

**30. Write a Shell Script to Check Whether a String is Palindrome or not.**

**Code:**

echo "Enter a string: "

read string

reverse=$(echo $string | rev)

if [ "$string" == "$reverse" ]

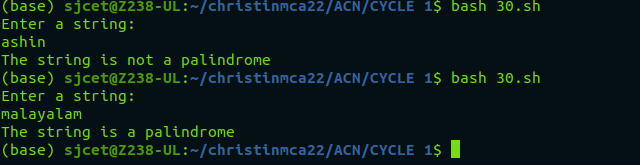
then

echo "The string is a palindrome"

else

echo "The string is not a palindrome"

fi

****

**31. Write a shell script to find out the unique words in a file and also count the occurrence of each of these words.**

**Code:**

echo "Enter the file name: "

read file

if [ ! -f "$file" ]; then

echo "File not found."

exit 1

fi

contents=$(tr '[:upper:]' '[:lower:]' < $file | sed 's/[^a-z0-9]/ /g')

words=($contents)

declare -A count

for word in "${words[@]}"; do

if [ -n "$word" ]; then

((count[$word]++))

fi

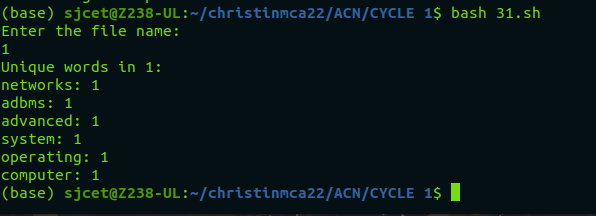
done

echo "Unique words in $file:"

for word in "${!count[@]}"; do

echo "$word: ${count[$word]}"

done

****

**32. Write a shell script to get the total count of the word “Linux” in all the “.txt” files and also across files present in subdirectories.**

**Code:**

search\_dir="."

files=$(find "$search\_dir" -type f -name "\*.txt")

count=0

for file in $files; do

occurrences=$(grep -o "Linux" "$file" | wc -l)

count=$((count + occurrences))

done

echo "Total count of 'Linux' in all .txt files: $count"

****

**33. Write a shell script to validate password strength. Here are a few assumptions for the**

**password string.**

**Length – minimum of 8 characters.**

**Contain both alphabet and number.**

**Include both the small and capital case letters.**

**Code:**

read -p "Enter your password: " password

if [[ ${#password} -lt 8 ]]; then

echo "Password length must be at least 8 characters."

exit 1

fi

if ! [[ "$password" =~ [A-Za-z]+[0-9]+ ]]; then

echo "Password must contain both alphabet and number."

exit 1

fi

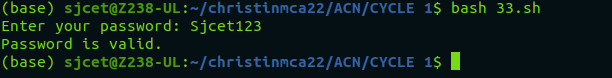
if ! [[ "$password" =~ [a-z]+ ]] || ! [[ "$password" =~ [A-Z]+ ]]; then

echo "Password must include both small and capital case letters."

exit 1

fi

echo "Password is valid."

****

**34. Write a shell script to print the count of files and subdirectories in the specified directory.**

**Code:**

echo "Enter directory path: "

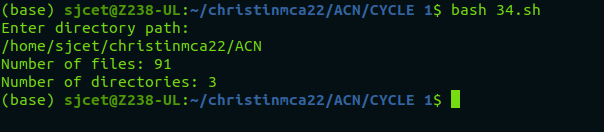
read directory

num\_files=$(find $directory -type f | wc -l)

num\_directories=$(find $directory -type d | wc -l)

echo "Number of files: $num\_files"

echo "Number of directories: $num\_directories"

****

**35. Write a shell script to reverse the list of strings and reverse each string further in the list.**

**Code:**

my\_list=("string1" "string2" "string3" "string4")

my\_list=($(echo "${my\_list[@]}" | tr ' ' '\n' | tac | tr '\n' ' '))

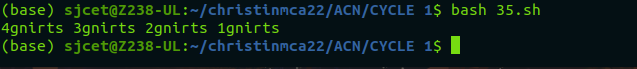
for i in "${!my\_list[@]}"

do

my\_list[$i]=`echo ${my\_list[$i]} | rev`

done

echo "${my\_list[@]}"



**LAMP INSTALLATION**

To install LAMP on your computer follow these steps.

## **Step 1: Update your system**

sudo apt-get update

## **Step 2: Install Mysql**

sudo apt-get install mysql-server mysql-client libmysqlclient-dev

## **Step 3: Install Apache server**

sudo [apt](https://www.youtube.com/results?search_query=%23apt)-get install apache2 apache2-doc apache2-npm-prefork apache2-utils libexpat1 ssl-cert

## **Step 4: Install PHP (php7.0 latest version of PHP)**

sudo apt-get install libapache2-mod-php7.0 php7.0 php7.0-common php7.0-curl php7.0-dev php7.0-gd php-pear php-imagick php7.0-mcrypt php7.0-mysql php7.0-ps php7.0-xsl

## 

## **Step 5: Install Phpmyadmin(for database)**

sudo apt-get install phpmyadmin