**1. Escriba un programa que le un archivo, determine si existe o no**

echo "Ingrese nombre de archivo : "

read file

if [ -f $file ]

then

echo "El archivo " $file "existe"

else

echo "El archivo " $file "no existe"

fi

**2. Lea un nombre el cual representa un archivo e imprima. Si el archivo es un archivo o un directorio de lo contrario que no exista debe imprimir un letrero archivo no existe.**

echo "Ingrese Nombre : "

read $data

if [ -d $data ]; then

echo "Es un directorio"

$cond=true

fi

if [ -f $data ]; then

echo "Es un archivo"

$cond=true

fi

if [ $cond = false ]; then

echo "No existe"

fi

**3. Le un nombre archivo y un nombre. Si el archivo existe debe de copiarlo en el**

**directorio leído, si el directorio no existe debe de crearlo y luego copiar.**

echo "Ingrese nombre de archivo : "

read file

if [ -f $file ]; then

echo "Ingrese nombre de directorio : "

read dir

if [ -d $dir ]; then

cp $file $dir

else

mkdir $dir

cp $file $dir

fi

else

echo "El archivo " $file "no existe"

fi

**4- Escriba un programa que calcule el área de un triángulo rectángulo, dada la altura y la**

**base.**

echo "Ingrese base : "

read base

echo "Ingrese altura : "

read altura

area=`expr $(($base \* $altura)) / 2`

echo "Area : $area"

**5- Lea un archivo si existe el archivo debe leer un segundo archivo y renombrarlo.**

echo "Ingrese archivo : "

read file

if [ -f $file ]; then

echo "Ingrese nuevo nombre : "

read new\_name

mv $file $new\_name

fi

**6. Escriba un programa que al pasarle por argumento un parámetro. Determine si es un**

**archivo, un directorio o si el parámetro no existe.**

if [ -f $1 ]; then

echo "El archivo existe"

else

echo "El archivo no existe"

fi

**7. Haga un programa que al pasarle por parámetro un archivo o directorio, devuelva el**

**tamaño en MB.**

echo "Ingrese nombre de archivo : "

read $file

du -sh $file

**8. Escriba un programa que al pasarle por argumento un parámetro. Determine si es un**

**archivo, un directorio o si el parámetro no existe. *Nota: si no se pasa ningún parámetro debe enviar un mensaje diciéndole cuales son los***

***parámetros.***

if [ $# -eq 0 ]

then

echo "No ha pasado parametros"

else

cond=false

if [ -d $1 ]; then

echo "El parametro es un directorio"

$cond=true

fi

if [ -f $1 ]; then

echo "El parametro es un archivo"

$cond=true

fi

if [ $cond = false ]; then

echo "El parametro no existe"

fi

fi

**9.Realice un programa que pase como parámetro un comando y presente la ayuda de ese**

**comando.**

man $1

**12. Realice un programa que elimine un archivo o directorio pasado como parámetro, y le**

**pregunte si está seguro de eliminarlo.**

echo "Estas Seguro que desea eliminar? (Si->1 No -> 0)"

read cond

if [ cond -eq 1 ]; then

if [ -d $1 ]; then

rm -rf $1

fi

if [ -f $1 ]; then

rm $1

fi

else

echo "NO SE ELIMINO EL ARCHIVO"

fi

**13. Lea un número e imprímalo desde la unidad hasta el mismo**

echo "Ingrese numero : "

read num

for ((cont = 1; cont <= num; cont++)); do

echo $cont

done

**14. Lea un número e imprima los números desde el mismo hasta la unidad.**

echo "Ingrese numero : "

read num

for ((i = num; i >= 1; i--)); do

echo $i

done

**15. Realice un programa que lea 10 números e imprima el mayor de ellos.**

declare mayor

for ((i=1; i <= 10; i++)); do

echo "Ingrese #" $i " : "

read numero

if [ $i -eq 1 ]; then

mayor=$i

else

if [ $numero -gt $mayor ]; then

mayor=$i

fi

fi

done

echo "Mayor : " $mayor

**16. Lea 10 números e imprima el menor.**

declare menor

for ((i=1; i <= 10; i++)); do

echo "Ingrese #" $i " : "

read numero

if [ $i -eq 1 ]; then

menor=$i

else

if [ $numero -lt $menor ]; then

menor=$i

fi

fi

done

echo "menor : " $menor

**17. Lea 10 números e imprima la suma y el promedio de ellos.**

declare menor

declare mayor

suma=0

for((i=1; i <= 10; i++)); do

echo "Ingrese #" $i " : "

read num

if [ $i -eq 1 ]; then

menor=$i

mayor=$i

else

if [ $num -lt $menor ]; then

menor=$i

fi

if [ $num -gt $mayor ]; then

mayor=$i

fi

fi

suma=`expr $suma + $num`

done

promedio=`expr $suma / 10`

echo "Suma : " $suma

echo "Promedio : " $promedio

echo "Mayor : " $mayor

echo "Menor : " $menor

**18. Lea 10 números e imprima el mayor, menor, la suma y el promedio.**

suma=0

for((i=1; i <= 10; i++)); do

echo "Ingrese #" $i " : "

read num

suma=`expr $suma + $num`

done

promedio=`expr $suma / 10`

echo "Suma : " $suma

echo "Promedio : " $promedio

**19. Lea un número e imprima si el número es par o impar.**

echo "Ingrese numero : "

read numero

if [ `expr $numero % 2` -eq 0 ]; then

echo "es par"

else

echo "es impar"

fi

**20. Lea dos números e imprima los números pares que hay entre ellos.**

echo "Numero 1 : "

read num\_one

echo "Numero 2 : "

read num\_two

for((i=num\_one; i <= num\_two; i++)); do

if [ `expr $i % 2` -eq 0 ]; then

echo $i

fi

done

**21. Imprima los números del 1 al 500 y diga el total de la suma de los números**

**pares e impares.**

suma\_par=0

suma\_impar=0

for ((i=1; i <= 500; i++)); do

if [ `expr $i % 2` -eq 0 ]; then

suma\_par=`expr $suma\_par + $i `

else

suma\_impar=`expr $suma\_impar + $i `

fi

done

echo "Suma Pares : " $suma\_par

echo "Suma Impares : " $suma\_impar

**22. Lea un número e imprima el factorial del mismo.**

echo "Ingrese un numero : "

read num

fact=1

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

fact=`expr $fact + $fact \\* $i`

#fact=$(($fact + $fact \* i))

done

echo $fact

**23. Lea dos números e imprima el factorial de la suma.**

echo "Ingrese un numero : "

read num\_one

echo "Ingrese otro numero : "

read num\_two

suma=`expr $num\_one + $num\_two`

fact=1

for((i = 1; i <= suma; i++)); do

fact=`expr $fact + $fact \\* $i`

#fact=$(($fact + $fact \* i))

done

echo $fact

**24. Lea un número y genere la tabla de multiplicar del mismo hasta el multiplicador**

**de 12.**

echo "Ingrese numero : "

read num

for((i = 1; i <= 12; i++)); do

echo $num " \* " $i " = " $(($num \* $i))

done

**25. Genere la tabla de multiplicar de los números 1 hasta el 13.**

for((i=1; i <=13; i++)); do

echo "##### Table de Multiplicar del #" $i " #####"

for((j=1; j <= 12; j++)); do

echo $i " \* " $j " = " $(($i \* $j))

done

done

**26. Lea dos números e imprima la multiplicación de ellos utilizando la suma**

**sucesiva.**

echo "Numero 1 : "

read num\_one

echo "Numero 2 : "

read num\_two

mult=0

while [ $num\_two -gt 0 ]; do

mult=`expr $mult + $num\_one`

num\_two=`expr $num\_two - 1`

done

echo "Multiplicacion : " $mult

**27. Lea dos números e imprima la división de ellos utilizando la resta sucesiva.**

echo "Numero 1 : "

read num\_one

echo "Numero 2 : "

read num\_two

div=0

while [ $num\_one -ge $num\_two ]; do

num\_one=`expr $num\_one - $num\_two`

div=`expr $div + 1`

done

echo "Division : " $div

**28. Lea un numero y un nombre e imprima el nombre tanta veces sea el numero.**

echo "Numero : "

read num

echo "Nombre : "

read nombre

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

echo $nombre

done

**29. Lea un nombre y un numero si el numero es par imprima el nombre tanta veces**

**sea la mitad del numero, de lo contrario de ser impar imprima el nombre tanta**

**veces sea el número.**

echo "Numero : "

read num

echo "Nombre : "

read nombre

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

for((i=1; i <= num / 2; i++)); do

echo $nombre

done

else

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

echo $nombre

done

fi

**30. Lea un número e imprima los divisores de eses números.**

echo "Numero : "

read num

for((i = 1; i <= 10; i++)); do

if [ $(($i % 2)) -eq 0 ]; then

echo $i

fi

done

**31. Lea un número e imprima si el número es primo o no.**

divs=0

echo "num : "

isprimo=true

read num

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

if [ `expr $num % $i` -eq 0 ]; then

divs=`expr $divs + 1`

fi

done

if [ $divs -gt 2 ]; then

echo "No Es primo"

else

echo "es primo"

fi

**32- Imprima cuantos números primos hay desde el 1 al 1000**

for ((h=1; h <= 1000; h++)); do

divs=0

for ((i=1; i <= h; i++)); do

if [ `expr $h % $i` -eq 0 ]; then

divs=`expr $divs + 1`

fi

done

if [ ! $divs -gt 2 ]; then

echo $h

fi

done

**33- Hacer un programa que lea un número y genere la serie Fibonacci hasta el numero**

**leído: Ejemplo se lee el 34 debe imprimir: 1, 2, 3, 5, 8, 13, 21, 34**

echo "Numero : "

read num

x=0

y=1

aux=0

while [ $aux -lt $num ]; do

aux=$((x+y))

echo $aux

x=y

y=aux

done

**34- Leer un número el cual representa la cantidad de digito diga cual es el número**

**máximo que puede tener esa cantidad de digito e imprima los números del 1**

**hasta el número máximo.**

echo "Numero : "

read num

maxnum=""

concat="9"

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

maxnum\_string=$maxnum $concat

done

echo $maxnum