


EJERCICIOS FOR IN Y FOR

Ejercicio 1:

Genera un script que pasados dos argumentos numéricos, que será el número de filas y el número de columnas respectivamente, rellene con un "*" cada elemento de la tabla creada. Fíjate en los ejemplos.



```
salvu@MiniSally:~/Escritorio$ ./Script3_3.sh 4 4
****
****
****
****

salvu@MiniSally:~/Escritorio$ ./Script3_3.sh 6 10
*****
*****
*****
*****
*****
*****
```

```
#!/bin/bash

for i in `seq 1 $1`
do
    for j in `seq 1 $2`
    do
        echo -n "*"
    done
    echo
done
```

-n para que no haga el salto de línea y dibuje los * seguidos

Ejercicio 2:

Crea un programa que, usando un bucle for, añada el permiso de ejecución al grupo del usuario (nada más al grupo del usuario, no al usuario propietario ni al resto), de los archivos y directorios de la carpeta que le pases como argumento.

Pasar como argumento /home/marina/Documentos

```
#!/bin/bash
if [ $# -ne 1 ]
then
    echo "ERROR. PASA 1 ARGUMENTO"
elif ! [ -d $1 ]
then
    echo "NO ES UN DIRECTORIO"
else
    cd $1
    for recurso in `ls`
    do
        chmod g+x $recurso
    done
fi
```

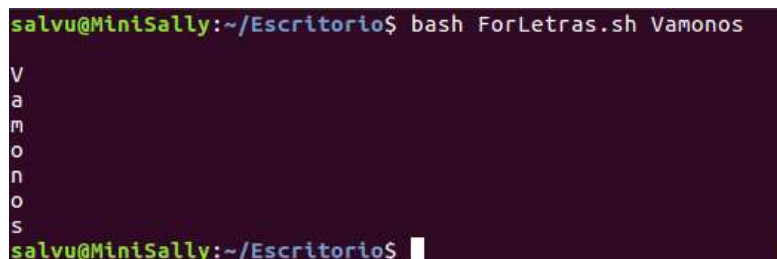
Ejercicio 3:

Teniendo en cuenta que el comando “expr” nos puede devolver la longitud de una palabra (length) y que su función substr puede seleccionar una subcadena de una cadena de texto:

Utiliza un bucle FOR que pasada una palabra como argumento devuelva una letra por línea.

*Recuerda usar el manual para ver cómo usar las opciones length y substr de expr.

También puedes hacer uso de la opción “-c” del comando cut para seleccionar las subcadenas (letras). **Cut -c 2-3, cut -c 2,3 → coge solo los caracteres indicados.



```
salvu@MiniSally:~/Escritorio$ bash ForLetras.sh Vamonos
V
a
m
o
n
o
s
salvu@MiniSally:~/Escritorio$
```

```
#!/bin/bash
longitud=`expr length $1`
for num_letra in `seq 1 $longitud`
do
    echo $1 | cut -c $num_letra
done
```

Ejercicio 4:

Crea una carpeta llamada "EliminaTXT", cuya ruta absoluta se pasará como argumento. El script debe, mediante un bucle "for", eliminar uno a uno cada fichero .txt (únicamente los ficheros .txt, nada más). Una vez terminado debe mostrar un ls -l para comprobar el correcto funcionamiento y corroborar que no quedan ficheros .txt en el directorio. Se dará por sentado que los ficheros no han sido nombrados con espacios en blanco.

Se pasa como argumento la ruta absoluta de la carpeta eliminaTXT (/home/marina/eliminaTXT)

```
#!/bin/bash
IFS=$'\n'
for i in `ls $1/*.txt`
do
    echo "HAS BORRADO $i"
done

echo `ls $1`
```


Falta añadir después del do rm -rf \$i para borrar de verdad el archivo.

Ejercicio 5:

Dado un fichero llamado expira.log que contenga en cada fila un nombre de usuario, se debe realizar un script que espere como argumento un número, dicho número será el número de días tras los cuales expirará la contraseña (comando chage).

El script debe recorrer todos los usuarios del fichero mediante un bucle for ((...)) y debe configurarlos para que su contraseña expire después de los días pasados como argumento.

Ejemplo de expira.log



```
prueba
prueba1
carlos
miguel
jose
```

Ejemplo de uso del script:

```
Modificado fecha de expiración de prueba a 1 días
Modificado fecha de expiración de prueba1 a 2 días
Modificado fecha de expiración de carlos a 3 días
Modificado fecha de expiración de miguel a 4 días
Modificado fecha de expiración de jose a 5 días
```

```
#!/bin/bash
#PARA VER EL CAMBIO DEL CHAGE SE PUEDE VER EN /ETC/SHADOW
num_lineas=`wc -l expira.log | cut -d' ' -f1`
for ((i=1;i<=$num_lineas;i++))
do
    usuario=`head -n$i expira.log | tail -n1`
    echo "chage -M $usuario, modificada fecha de expiración `echo $* | cut -d ' ' -f $i`"
done
```

Ejercicio 6.

Escribe un script que solicite un número impar como argumento. En caso de que se introduzca un número par debe devolver un error.

Acto seguido el script devolverá un árbol de asteriscos, teniendo una base igual al número pasado como argumento.

Ejemplos:

[illegible]

```

#!/bin/bash
if [ `expr $1 % 2` -eq 0 ]
then
    echo "ERROR. INTRODUCER UN NÚMERO IMPAR"
else
    col=$1
    let "fil=($1 / 2) + 1"

    let "tope=$fil + 1"

    for (( i=$fil;i>0;i-- ))
    do
        for (( j=1;j<=$col;j++ ))
        do
            #      echo "J $j  I $i  TOPE $tope"
            if [[ $j -ge $i && $j -lt $tope ]]
            then
                echo -n "*"
            else
                echo -n " "
            fi
        done
        #read
        echo
        let "tope = $tope + 1"
        #echo $tope
    done
fi

```