

**NOMBRE:**

1. Crea un programa que reciba por parámetro dos números y muestre los números que estén entre los dos. Podrá recibir los números de menor a mayor o viceversa, es decir, si pasamos como parámetros 1 3 mostrará del 1 al 3 y si pasamos 3 1 mostrará del 3 al 1. (1,5 puntos)

```
#!/bin/bash
if [ $1 -lt $2 ]
then
    for ((i=$1;i<=$2;i++))
    do
        echo $i
    done
else
    for ((i=$1;i>=$2;i--))
    do
        echo $i
    done
fi
```

2. Realiza un script que compruebe si el formato de un dominio de internet es correcto. Los dominios tendrán el siguiente formato. Pueden contener letras, números y guiones los nombres de dominio y subdominios, siempre que empiecen por letra (todo minúscula). La longitud también será mínimo de 1 carácter. La extensión final del dominio (.com,.net etc...), sólo podrá contener letras y una extensión entre 2 y 4 caracteres. Se podrán anidar todos los subdominios que se quieran -> (este.es.un.dominio.muymuy.largo.com) (1 punto)

```
#!/bin/bash
var="([a-z])[a-z0-9_-]+\.[a-z]{2,4}"
if [[ $1 =~ $var ]]
then
    echo "El dominio $1 es correcto"
else
    echo "El dominio $1 es incorrecto"
fi
```

3. Crea un script que muestre los nombres de los procesos que lleven más de 3 segundos de tiempo de ejecución en la CPU. (2 puntos).

```
#!/bin/bash
i=0 #para eliminar la línea de cabecera, así para la 1ª línea del for no ejecutamos el if

for linea in `ps -ef | tr -s ' ' '#' | cut -d '#' -f7,8`
do
    if [ $i -gt 0 ]
    then
        tiempo=`echo $linea | cut -d '#' -f1`
        nom=`echo $linea | cut -d '#' -f2`
        h=`echo $tiempo | cut -d ':' -f1`
        m=`echo $tiempo | cut -d ':' -f2`
        s=`echo $tiempo | cut -d ':' -f3`

        if [ $h -ge 1 ] || [ $m -ge 1 ] || [ $s -ge 3 ]
        then
            echo $nom
        fi
    fi
    i=1
done
```

4. Crea un programa que lea del fichero /root/matar\_usuarios, una lista de usuarios, y para cada uno de ellos, mate a todos sus procesos. (1,5 puntos)

```
#!/bin/bash
for usu in `cat /root/matar_usuarios`
do
    for p in `ps -e u | tr -s ' ' ' ' | grep -w $usu | cut -d ' ' -f2`
    do
        #kill -9 $p
        echo "se ha matado el proceso $p"
    done
done
```

**5. Crea un programa capaz de gestionar el fichero del ejercicio anterior. Las opciones que debe contener son: (4 puntos)**

- Listar los usuarios.
- Añadir un usuario. Se comprobará si el usuario ya existe en el fichero o no, para añadirlo. Se mostrarán mensajes de error en el caso de que no se haya introducido usuario o el usuario ya exista y también mensaje cuando el usuario se haya añadido correctamente. El fichero debe permanecer ordenado alfabéticamente.
- Eliminar un usuario. Se comprobará si el usuario existe en el fichero y se mostrará mensaje de error si no existe.
- Salir. Solo se podrá salir pulsando esta opción.

```
while [ 1 -eq 1 ]
do
    echo 1 – listar
    echo 2 – añadir
    echo 3 – borrar
    echo 4 - salir

    read op

    if [ $op -eq 1 ]
    then
        cat /root/matar_usuarios
    fi

    if [ $op -eq 2 ]
    then
        read -p "¿Qué usuario quieres añadir?" usu
        usufich=`cat /root/matar_usuarios | grep -w $usu`

        if [[ ($usu != $usufich) && (-n $usu) ]]
        then
            echo $usu >> /root/matar_usuarios
            sort -k1 /root/matar_usuarios
            echo "usuario añadido correctamente al fichero matar_usuarios"

        elif [[ $usufich = $usu ]]
        then
            echo "el usuario ya existe en el fichero matar_usuarios"

        elif [[ -z $usu ]]
        then
            echo "usuario no válido"
```

```
        then
            echo "no has introducido ningún usuario"
    fi
    if [ $op -eq 3 ]
    then
        read -p "¿Qué usuario quieres borrar?" borrar
        borrarfich=`cat /root/matar_usuarios | grep -w $borrar`
        if [[ -z $borrarfich ]]
        then
            echo "el usuario no está en el fichero"
        else
            for linea in `cat /root/matar_usuarios`
            do
                if [ $linea != $borrar ]
                then
                    echo $linea >> aux
                fi
            done
            mv aux /root/matar_usuarios
        fi
    fi
    if [ $op -eq 4 ]
    then
        exit
    fi
done
```