EJERCICIOS REPASO PARA EVAL. EXTRAORDINARIA

1. **Realiza un script llamado gastos.sh que te va a pedir una serie de gastos que has podido tener en un viaje. Primero preguntará el concepto, que puede ser una palabra entre "gasolina", "comida", "otros" o "salir" (con el cual el programa finaliza), no dejes meter ninguna otra palabra. Después de cada concepto te preguntará la cantidad gastada (introduce un número entero siempre)**
    - **El programa te preguntará de forma indefinida por gastos (concepto → cantidad) hasta que escribas "salir" como concepto. Y se pueden introducir tantos gastos del mismo, o diferente concepto como se quiera y en el orden que se quiera.**
    - **Al final del programa (al seleccionar "salir") te mostrará los 3 tipos de gastos de forma ordenada y el total de cada gasto.**
    - **Ejemplo de salida al final del programa**

      Gastos gasolina:
      60€
      70€
      Total 130€

      Gastos comida:
      35€
      25€
      20€
      Total: 80€

```
#!/bin/bash
#script para control de gastos, con menu de opciones

contador_gasolina=0
contador_comida=0
contador_otros=0

while [ 1 -eq 1 ]
do
        echo "Elige un concepto de gasto: gasolina/comida/otros/salir"
        read concepto

        case $concepto in
                gasolina)
                        read -p "introduce el importe de gasolina: " gasolina
                        gasolina_array[$contador_gasolina]=$gasolina
                        let contador_gasolina++
                        #echo "${gasolina_array[*]}"
                ;;
                comida)
                        read -p "introduce el importe de comida: " comida
                        comida_array[$contador_comida]=$comida
                        let contador_comida++
                        #echo "${comida_array[*]}"
```

```
                ;;
                otros)
                        read -p "introduce el importe de otros: " otros
                        otros_array[$contador_otros]=$otros
                        let contador_otros++
                        #cho "${otros_array[*]}"
                ;;

                salir)
                        #calculo de los gastos de gasolina e impresión por pantalla
                        echo "Gastos gasolina:"
                        for (( i = 0; i < ${#gasolina_array[*]}; i++ ))
                        do
                                echo "${gasolina_array[$i]}€"
                                let "total_gasolina = $total_gasolina +
${gasolina_array[$i]}"
                        done
                        echo -e "Total $total_gasolina€\n"

                        #calculo de los gastos de comida e impresión por pantalla
                        echo "Gastos comida:"
                        for (( i = 0; i < ${#comida_array[*]}; i++ ))
                        do
                                echo "${comida_array[$i]}€"
                                let "total_comida = $total_comida +
${comida_array[$i]}"
                        done
                        echo -e "Total $total_comida€\n"

                        #calculo de los gastos otros e impresión por pantalla
                        echo "Gastos otros:"
                        for (( i = 0; i < ${#otros_array[*]}; i++ ))
                        do
                                echo "${otros_array[$i]}€"
                                let "total_otros = $total_otros + ${otros_array[$i]}"
                        done
                        echo -e "Total $total_otros€\n"


                        exit
                ;;
                *)
                        echo "no has escrito una opción correcta, por favor vuelve a
intentarlo"
                ;;
        esac
done
```

2. **Crea un script que pregunte el nombre de un usuario y calcule el número de ficheros con permisos de ejecución existentes en su carpeta de trabajo. El programa debe entrar también en los subdirectorios existentes.**

```
#!/bin/bash

read -p "Dime el nombre de un usuario: " usuario

for linea in `cat /etc/passwd | tr -s ' ' '@'`
do
        usu=`echo $linea | cut -d ':' -f1`
        if [ $usu = $usuario ]
        then
                dir=`echo $linea | cut -d ':' -f6`
        fi
done
find $dir -user $usuario -perm /u+x | wc -l
```

3. **Crea un programa que reciba el nombre de un usuario y muestre el número de procesos que tiene en memoria y el número de ficheros que tiene en su carpeta de trabajo.**

```
#!/bin/bash
np=`ps -e u | grep -c ^$1`
nf=`find /home/$1 -user $1 2>/dev/null | wc -l`
echo El usuario $1 tiene $np procesos y $nf ficheros
```

4. **Crea un programa que reciba un número y calcule su factorial. Ejemplo: El factorial de 5 es 5*4*3*2*1**

```
#!/bin/bash
echo "escribe un número"
read num
i=$num
res=1
while [ $i -gt 1 ]
do
    res=$[$res*$i]
    i=$[$i-1]
done
echo "el factorial de $num es $res"
```

```
**************************
```

```
#!/bin/bash
read -p "escribe un número    " num
res=1
for ((i=1;i<=$num;i++))
do
    let res=$res*$i
done
echo "el resultado es $res"
```

5. **Crea un programa que funcione como una ayuda sobre el funcionamiento de los comandos del sistema. El programa mantendrá un fichero donde se relacionará comandos con una descripción.**
   Se deben implementar las siguientes funciones:
   - **Insertar comando y descripción. Se comprobará antes si el comando ya existe en la lista y se mostrará el correspondiente mensaje de error. La lista estará ordenada alfabéticamente.**
   - **Listar los comandos existentes (solo los comandos, sin la descripción).**
   - **Buscar la descripción de un comando. Ej. ls - lista el contenido de un directorio. Si el comando no está se mostrará mensaje de error.**
   - **Borrar comando.**
   - **Salir. Solo se podrá salir del programa cuando se pulse esta opción.**

```
#!/bin/bash
while [ 1 -eq 1 ]
do
        echo
        echo "Elige una opción"
        echo 1 - añadir
        echo 2 - listar
        echo 3 - buscar
        echo 4 - borrar
        echo 5 - salir
        echo
        read op

        case $op in
        1)
                echo "Introduzca comando"
                read com
                nom_op1=`cat diccionario | grep -w $com | cut -d '#' -f1`
                if [[ $nom_op1 = $com ]]
                then
                        echo "El comando ya está en la lista, elija otro"
                else
                        echo "Introduzca descripción"
                        read desc
                        echo $com#$desc >> diccionario
                        sort diccionario > diccionario1
                        mv diccionario1 diccionario
                        echo "Comando añadido correctamente"
                fi
        ;;

        2)
                cat diccionario | cut -d '#' -f1
    ;;

        3)
                echo "Introduzca el comando"
                read com
```

```
                                #for linea in `cat diccionario | tr -s ' ' '|'`
                                #do
                                        #nom=`echo $linea | cut -d '#' -f1`
                                        #if [ $nom = $com ]
                                        #then
                                                #desc=`echo $linea | cut -d '#' -
f2 | tr -s '|' ' '`
                                                #echo $desc
                                        #fi
                                #done
                        descripcion=`cat diccionario | grep -w $com`
                        if [[ -z $descripcion ]]
                        then
                                echo "El comando no está en la lista"
                        else
                                echo "$descripcion"
                        fi
                ;;

                4)
                        read -p "Introduzca el comando a borrar     "  com1
                        borrar=`cat diccionario | grep -w $com1`

                        if [[ -z $borrar ]]
                        then
                                echo "El comando no está en la lista, escriba otro"
                        else
                                touch aux
                                for linea in `cat diccionario | tr -s ' ' '@'`
                                do
                                        nom=`echo $linea | cut -d '#' -f1`
                                        if [ $nom != $com1 ]
                                        then
                                                echo $linea | tr -s '@' ' ' >> aux
                                        fi
                                done
                                mv aux diccionario
                                echo "comando borrado correctamente"
                        #else
                                #cat diccionario | egrep -v $com1 > diccionario1
                                #mv diccionario1 diccionario
                                #echo "comando borrado correctamente"
                        fi
                ;;

                5)
                        exit
                ;;
esac
done
```

6. **Realiza un script llamado decimales.sh que a partir de un archivo llamado decimales.txt que contiene una serie de números decimales, te diga:**

   - **El número mayor**
   - **El número menor**
   - **La media de todos los números (con 2 decimales de precisión)**

```
#!/bin/bash
array_dec=( `cat decimales.txt` )
#echo "${array_dec[*]}"
max=0
min=${array_dec[0]}
suma=0
#echo $min
#echo ${#array_dec[*]}
for (( i=0;i<${#array_dec[*]};i++ ))
do
        if (( $(echo "scale=2; ${array_dec[$i]} > $max" | bc -l) ))
        then
                max=${array_dec[$i]}
        fi
        if (( $(echo "scale=2; ${array_dec[$i]} < $min" | bc -l) ))
        then
                min=${array_dec[$i]}
        fi
        suma=$(echo "scale=2; $suma+${array_dec[$i]}" | bc -l)
done
        media=$(echo "scale=2; $suma/${#array_dec[*]}" | bc -l)

echo "El número mayor es: $max"
echo "El número menor es: $min"
echo "la media de los números decimales es: $media"
```