**PRÁCTICA 2 - PROGRAMACIÓN SHELL-SCRIPTS**

Desarrollar dentro de un directorio, de nombre **Shell-Script**, cada uno de los siguientes  
programas. Cada programa se llamará ejercicio1.sh, ejercicio2.sh, y así sucesivamente.

1. Programa que reciba tres palabras como parámetros y las visualice al revés.

#!/bin/bash

echo $3 $2 $1;

2. Programa que visualice mensaje de error si no se le pasan parámetros.

#!/bin/bash

if [ $# -eq 0 ]

then

echo “Error, no se han pasado parámetros”;

fi

3. Programa que visualice un mensaje de error si no se le pasan parámetros, y que los  
visualice si se le pasan.

#!/bin/bash

if [ $# -eq 0 ]

then

echo “Error, no se han pasado parámetros”;

else

echo $\*;

fi

4. Programa que reciba dos números y diga cuál es el mayor.

#!/bin/bash

if [ $1 -gt $2 ]

then

echo “El mayor es $1”;

else

echo “El mayor es $2;

fi

5. Programa que acepte dos números como parámetro y diga si el primero es mayor,  
menor o si son iguales.

#!/bin/bash

if [ $1 -gt $2 ]

then

echo “El mayor es $1”;

else

echo “El mayor es $2;

fi

6. Programa que reciba dos números como parámetro y devuelva la suma si el primero  
es mayor que el segundo y la resta en caso contrario.

#!/bin/bash

if [ $1 -gt $2 ]

then

echo $(($1 + $2));

else

echo $(($2 - $1));

fi

7. Programa llamado existe al que se le pasa como parámetro un fichero y dice si existe,  
no existe o es un directorio.

8. Realiza un shell-script que muestre todos los números pares, desde el 0 hasta el 100.

9. Realiza un shell-script, que calcule la potencia de un número. El shell pedirá la base  
y el exponente, devolviendo la potencia.

10. Realiza un shell-script que realice una división de dos números que se pasen por  
parámetro por el método de las restas.

11. Implementar un shell-script que copie todos los ficheros que se le pasen por  
parámetro, al directorio Seguridad. Si directorio no existe, se deberá crear.

12. Implementa un shell-scipt que indique si los ficheros pasados por parámetro existen  
y si son ficheros o directorios.

13. Realiza un shell-script que muestre el nombre de cada uno de los ficheros pasados  
por parámetro y visualice su contenido por pantalla.

14. Implementa un shell-script, que permita intercambiar el nombre de 2 ficheros  
pasados por parámetro.

15. Realiza un shell-script que limpie la pantalla, muestre todos los nombres de ficheros  
que hay en el directorio actual (mostrando sus atributos), la fecha actual y vuestro  
nombre.

16. Realiza un shell-script que permita recibir, al menos, dos parámetros. El primero  
serán opciones, y el segundo un fichero. Si el primer parámetro es -e, el shell  
ejecutará el programa que se pasa como segundo parámetro, si el primer parámetro  
es -m, el shell mostrará el contenido del fichero pasado como segundo parámetro.

17. Realiza un shell-script, que pida dos números por teclado. El programa pintará tantos  
\*, como indique el número mayor de los dos introducidos.

18. Implementa un shell-script que compruebe si los ficheros pasados por parámetro  
existen. Si existen, se muestra el nombre del fichero, si no existen, se muestra un  
mensaje de error.