Los grupos del 1 al 13 deben llevar hechos al laboratorio el día 16 de marzo los ejercicios del 1 al 12.

Propuesta de trabajo personal al finalizar la sesión 1:

1. Queremos utilizar nombres que nos resulten más familiares para algunos mandatos, como:
   - dir en lugar de ls –l
   - escribe <argumentos> en lugar de echo <argumentos>

Realiza dos guiones que hagan estos cambios automáticamente.

2. Escribir el siguiente programa shell:
   - Pedir al usuario su nombre y almacenarlo en una variable.
   - Añadirlo a un fichero llamado **usuarios** que cuelgue del HOME.
   - Almacenar en otra variable el contenido del fichero **usuarios**.
   - Mostrar esta última variable por pantalla.

3. Escribir el siguiente programa shell:
   - Pedir al usuario su nombre y su primer apellido y almacenarlos cada uno en una variable distinta.
   - Añadir el nombre y el apellido a un fichero llamado **cadenas,** cada uno en una línea.
   - Almacenar en una variable los nombres de ficheros de tu HOME que contengan en su nombre la cadena **su**.
   - Mostrar esta última variable por pantalla.

4. Escribir el siguiente programa shell:

Tenemos almacenada la ruta de un fichero en una variable llamada FICHERO.
   - Mostrar por pantalla el contenido del fichero.
   - Guardar en una variable el número de líneas que tiene el fichero.
   - Mostrar por pantalla el valor de la variable y el del décimo argumento que se haya introducido en el programa.

5. Realizar el siguiente programa shell:
   - Mover un fichero que se pasa como argumento al directorio almacenado en una variable llamada DESTINO
   - Guardar en una variable la ruta del directorio actual
   - Mostrar por pantalla esa nueva variable y el número de argumentos que ha recibido el programa

6. Escribir el siguiente programa shell:
   - Copiar un fichero que se pasa como argumento al directorio que está almacenado en la variable RUTA.
   - Guardar en una nueva variable el número de usuarios conectados al sistema.
   - Mostrar por pantalla el contenido de la variable que has creado y el de la variable HOME.

7. La orden *pwd* nos indica el directorio de trabajo actual. Entonces,

- ¿qué hace el siguiente guión?
  mandato=pwd
  echo “El contenido de la variable mandato es $mandato”
- ¿y este otro guión?
  mandato=`pwd`
  echo “El contenido de la variable mandato es $mandato”
- ¿cómo explicas la diferencia?

8. ¿Hay alguna diferencia entre las salidas que producen por pantalla estos tres guiones?

```
mandato=`date`
echo “la fecha de hoy es $mandato”

echo “la fecha de hoy es `date`”

echo “la fecha de hoy es date”
```

9. Escribe un guión del shell que almacene en una variable USUARIOS el nombre de usuario de todos los usuarios que están conectados en este momento.

- Modifica el guión para que muestre el contenido por pantalla.
- Ahora utiliza argumentos en el guión y guarda en la variable USUARIOS sólo aquellos nombres de usuario que contengan la cadena de caracteres que se pase como primer argumento.

10. Repite el guión anterior, pero ahora en vez de tomar como argumento la cadena de caracteres, debe leerla por teclado.

11. Realiza un guión que guarde en una variable MIS_VARIABLES todas las variables de entorno que comiencen por MI, si es que hay alguna. Después, haz que las muestre por pantalla.

- haz una versión nueva que en vez de utilizar la constante “MI” acepte cualquier cadena de caracteres que se pase como primer argumento,
- haz otra versión adicional que lea por teclado la cadena.

12. Escribe un guión que reciba N argumentos y que muestre por pantalla los que ocupan las posiciones 2, 3, ..., N. Utiliza para ello las órdenes *echo* y *shift*.

Estudiar el apartado 5.2 de tus apuntes y realizar los siguientes ejercicios:

1. Crear un programa shell que muestre el contenido del directorio actual si hay más de 2 ficheros o directorios.

2. Modificar el programa anterior para sea el usuario el que diga el número mínimo de ficheros o directorios que debe haber para que se muestren.

3. Modificar el programa 1 para que el número mínimo de ficheros o directorios se introduzca como argumento.

4. Crear un programa shell que compruebe si hay más de 3 ficheros (sólo ficheros) en el directorio actual y los muestre.

5. Hacer lo mismo que en el ejercicio anterior pero con directorios (sólo directorios).

6. Crear un programa shell al que se le introduzcan 2 argumentos, el primero una opción: -f para listar ficheros y –d para listar directorios, y el segundo el número mínimo de ficheros o directorios que debe haber para hacer el listado.

7. Crear un programa shell que compruebe que el fichero **prog1** existe y no está vacío.

8. Crear un programa shell al que se le pasa un argumento. Comprobar si es un fichero ordinario, sacar un mensaje por pantalla, y si no lo es, comprobar si es directorio y sacar un mensaje por pantalla.

9. Modificar el programa **mostrar** para que abandone la ejecución si el primer argumento es distinto de Pedro.

10. Modificar el programa **demo_shift** para que dé un mensaje cuando no se introduzca ningún argumento y abandone la ejecución.