

## Enunciado trabajo

Crear un script con comandos de Linux que admita dos o más argumentos; de forma que si no es así debe visualizar el mensaje *Error de sintaxis*. Una vez comprobado esto, debe presentar el siguiente menú por pantalla y programar, mediante funciones, las opciones según se detallan a continuación. Una vez terminada cada opción debe volver a mostrarse el menú y así sucesivamente hasta que escoja la opción "FIN":

Menú            [O] Ordenados  
                  [D] Directorio  
                  [L] Listado  
                  [F] FIN

*Opciones:*

*O u o:* Considerando los argumentos del script como números enteros, comprueba si están o no ordenados de forma ascendente, e informa mediante un mensaje de ello. Si algún argumento rompe el orden ascendente no se debe continuar comprobando el resto de argumentos.

*D o d:* Solicita al usuario el nombre de un directorio existente y comprueba si cada argumento del script es o no un fichero que podemos ejecutar. En el caso de que sea un fichero ejecutable se debe crear un enlace de ese fichero en el directorio cuyo nombre se ha obtenido del usuario. En cualquier otro caso se visualizará el mensaje "*No es un fichero o no puedo ejecutarlo*". **No se puede usar la orden *if*.**

*L o l:* Muestra mediante distintos mensajes el número de directorios y ficheros no vacíos existentes entre los argumentos recibidos por el script. Si algún argumento no es un directorio o no es un fichero no vacío se visualizará el mensaje "*No es un fichero no vacío o directorio*".

*F o f:* Sale del menú. Al salir del menú se debe mostrar el mensaje "*Fin del programa*" y finalizar el script.

*Nota:* Los argumentos con los que se invoque al script deben ser acordes con la opción del menú a elegir.

## Enunciado trabajo

Crear un script con comandos de Linux que admita dos o más argumentos; de forma que si no es así debe visualizar el mensaje *Error de sintaxis*. Una vez comprobado esto, debe presentar el siguiente menú por pantalla y programar, mediante funciones, las opciones según se detallan a continuación. Una vez terminada cada opción debe volver a mostrarse el menú y así sucesivamente hasta que escoja la opción “FIN”:

Menú            [R] Resto  
                  [C] Contar  
                  [D] Doble  
                  [F] FIN

Opciones:

*R o r*: Considerando los argumentos del script como números enteros, visualiza por pantalla el resto de dividir cada argumento con el siguiente argumento, es decir, el resto entre el primer y segundo argumento, resto entre el segundo y tercero, y así sucesivamente. Se debe tener en cuenta que no se puede dividir por cero.

*C o c*: Solicitar al usuario que introduzca el nombre de un nuevo fichero y comprobar que no existe. Si se cumple esto, mediante **un solo comando**, almacenar en dicho fichero el número de líneas, palabras y bytes de todos los ficheros dados como argumentos del script (no es necesario comprobar si son o no ficheros). En el caso de que se trate de un fichero existente visualizar el mensaje “*Ese fichero ya existe*”. **No se puede usar la orden if**.

*D o d*: Presenta por pantalla el doble de cada uno de los números que se encuentran entre el menor y el mayor de los dos primeros argumentos del script tratados como números enteros (sin incluirlos).

*F o f*: Sale del menú. Al salir del menú se debe mostrar el mensaje “*Fin del programa*” y finalizar el script.

Nota: Los argumentos con los que se invoque al script deben ser acordes con la opción del menú a elegir.

## Enunciado trabajo

Crear un script con comandos de Linux que admita tres o más argumentos; de forma que si no es así debe visualizar el mensaje “*Error de sintaxis*”. Una vez comprobado esto, debe presentar el siguiente menú por pantalla y programar, mediante funciones, las opciones según se detallan a continuación. Una vez terminada cada opción debe volver a mostrarse el menú y así sucesivamente hasta que escoja la opción “FIN”:

Menú            [P] *Primeras*  
                  [M] *Mitad*  
                  [E] *Procesos en Ejecución*  
                  [F] *FIN*

Opciones:

*P o p*: Pide un nombre de fichero por teclado, comprueba si dicho fichero existe y si tiene permiso de lectura. Si es así, debes mostrar por pantalla las 5 primeras líneas del fichero. Si no se cumple alguna de las condiciones iniciales (exista el fichero y permiso de lectura sobre él), debes presentar el siguiente mensaje por pantalla: “*El fichero no existe o no tiene permiso de lectura*”. **No se puede utilizar la instrucción if.**

*M o m*: Si el número de argumentos del script es impar visualiza el que está en medio, pero si el número de argumentos es par visualiza los dos argumentos que están en medio.

*E o e*: Muestra mediante mensajes si cada uno de los argumentos recibidos por el script corresponden o no al PID de procesos que en ese momento se están ejecutando en el sistema.

*F o f*: Sale del menú. Al salir del menú se debe mostrar el siguiente mensaje: “Fin del programa” y se acabará el script.

Nota: Los argumentos con los que se invoque al script deben ser acordes con la opción del menú a elegir.

## Enunciado trabajo

Crear un script con comandos de Linux que admita dos o más argumentos; de forma que si no es así debe visualizar el mensaje “*Error de sintaxis*”. Una vez comprobado esto, debe presentar el siguiente menú por pantalla y programar, mediante funciones, las opciones según se detallan a continuación. Una vez terminada cada opción debe volver a mostrarse el menú y así sucesivamente hasta que escoja la opción “FIN”:

Menú            [F] *Fichero*  
                  [C] *Calcular el cubo*  
                  [V] *Visualizar ficheros*  
                  [F] *FIN*

Opciones:

*V o v:* Comprueba si cada argumento del script es un fichero o no. En el caso de que sea un fichero debes añadirle el contenido del propio script que estás realizando. En el caso de que ese argumento no corresponda a un fichero debes visualizar el mensaje “*No es un fichero*”. **No se puede usar la orden if.**

*C o c:* Calcula y muestra por pantalla el cubo de los números dados como argumentos al script que no sean negativos. Si el argumento es negativo debes mostrar el mensaje “*Argumento negativo*”.

*V o v:* Mientras que el usuario no introduzca la cadena “Terminar” debes solicitarle el nombre de un fichero. Si realmente corresponde a un fichero existente, debes visualizar sus dos últimas líneas. En el caso de que no corresponda con un fichero debes mostrar el mensaje “*No es un fichero*”. No se puede usar la orden if.

*F o f:* Sale del menú. Al salir del menú se debe mostrar el siguiente mensaje: “Fin del programa” y se acabará el script.

Nota: Los argumentos con los que se invoque al script deben ser acordes con la opción del menú a elegir.

## Enunciado trabajo

Crear un script con comandos de Linux que admita tres o más argumentos; de forma que si no es así debe visualizar el mensaje *Error de sintaxis*. Una vez comprobado esto, debe presentar el siguiente menú por pantalla y programar, mediante funciones, las opciones según se detallan a continuación. Una vez terminada cada opción debe volver a mostrarse el menú y así sucesivamente hasta que escoja la opción “FIN”:

Menú            [S] Sucesión  
                  [C] Contar  
                  [P] Potencia  
                  [F] FIN

Opciones:

*S o s:* Considerando los argumentos del script como números enteros, informa mediante un mensaje si los argumentos cumplen o no la siguiente sucesión: cada argumento, excepto el primero, se obtiene sumando al argumento anterior el valor del primer argumento. Si algún argumento rompe esta sucesión no se debe continuar comprobando el resto de argumentos.

Ejemplos:

4	1	5	9	13	17	# si cumple la sucesión
3	2	5	8	11		# si cumple la sucesión
3	7	10	12	15		# no cumple la sucesión
2	8	10	12	15		# no cumple la sucesión

*C o c:* Solicitar al usuario que introduzca el nombre de un nuevo directorio y comprobar que existe. Si se cumple esto, se debe suponer que todos los elementos contenidos en dicho directorio son ficheros, es decir, no es necesario comprobar si son o no ficheros. Almacenar en un fichero llamado *recuento* el número de ficheros que cuelgan de ese directorio. En el caso de que se trate de un directorio inexistente visualizar el mensaje “*Ese directorio no existe*”. **No se puede usar la orden if.**

*P o p:* Solicitar al usuario que introduzca un número entero positivo mayor que 0 (no es necesario comprobar que se cumple esta condición) y presentar por pantalla el resultado de elevar cada argumento del script al número introducido. Los argumentos del script deben ser tratados como números.

*F o f:* Sale del menú. Al salir del menú se debe mostrar el mensaje “*Fin del programa*” y finalizar el script.

Nota: Los argumentos con los que se invoque al script deben ser acordes con la opción del menú a elegir.

## Enunciado trabajo

Crear un script con comandos de Linux que admita tres o más argumentos; de forma que si no es así debe visualizar el mensaje *Error de sintaxis*. Una vez comprobado esto, debe presentar el siguiente menú por pantalla y programar, mediante funciones, las opciones según se detallan a continuación. Una vez terminada cada opción debe volver a mostrarse el menú y así sucesivamente hasta que escoja la opción “FIN”:

Menú            [P] Predecir  
                  [I] Inodo  
                  [T] Triangular  
                  [F] FIN

Opciones:

*P o p:* Solicitar al usuario el número mínimo de veces que cree que se va a repetir el nombre del script entre los argumentos que éste recibe. Al ir comprobando el número de ocurrencias, se debe tener en cuenta que, si se excede el número mínimo propuesto por el usuario, no se debe continuar comprobando el resto de argumentos. Finalmente, se debe informar mediante un mensaje si el usuario ha acertado con ese número mínimo de ocurrencias o si se ha quedado corto. **No se puede usar la orden if.**

*I o i:* Solicitar al usuario que introduzca el inodo de un fichero y el nombre de un posible directorio. Comprobar que dicho directorio existe, en cuyo caso se debe mostrar el número de ficheros contenidos en ese directorio con ese mismo inodo. En el caso de que se trate de un directorio inexistente visualizar el mensaje “*Ese directorio no existe*”.

*T o t:* Tratando los argumentos del script como números enteros positivos (no es necesario comprobarlo), determina e informa, mediante un mensaje, si cumplen o no la *sucesión triangular*: 1 3 6 10 15 21 28 36 45 ... . Ésta sucesión se puede expresar mediante la regla  $x_n = n(n+1)/2$ , donde  $n$  corresponde a la posición del término en la sucesión, es decir para el primer término  $n$  será 1 ( $x_1=1(1+1)/2=1$ ), para el segundo término  $n$  valdrá 2 ( $x_2=2(2+1)/2=3$ ), y así sucesivamente.

*F o f:* Sale del menú. Al salir del menú se debe mostrar el mensaje “*Fin del programa*” y finalizar el script.

Nota: Los argumentos con los que se invoque al script deben ser acordes con la opción del menú a elegir.

## Enunciado trabajo

Crear un script con comandos de Linux que admita tres o más argumentos; de forma que si no es así debe visualizar el mensaje *Error de sintaxis*. Una vez comprobado esto, debe presentar el siguiente menú por pantalla y programar, mediante funciones, las opciones según se detallan a continuación. Una vez terminada cada opción debe volver a mostrarse el menú y así sucesivamente hasta que escoja la opción “Fin de programa”:

### *Menú*

[C] Contar valores positivos y negativos  
[A] Área y perímetro de forma cuadrada  
[R] Reverso de una cadena  
[F] Fin del programa

### *Opciones:*

**C o c:** Considerando los argumentos del script como números enteros, elabora una función que permita contar cuales son negativos y positivos. Una vez contados, se mostrará por pantalla un resumen con los resultados finales. Por ejemplo: “Positivos: 5 y Negativos 0”. **No se podrá hacer uso de las órdenes for o if.**

**A o a:** Solicitar al usuario que introduzca cuatro valores numéricos por teclado que servirán como lados de una forma cuadrada o rectangular. Se deberá comprobar que los valores introducidos se corresponden con una forma cuadrada, y caso afirmativo, se calcularán tanto el área como el perímetro de dicha forma. En caso de que la forma no sea cuadrada se deberá mostrar el mensaje “La forma no es cuadrada”.

**R o r:** Considerando el primer argumento que se le pasa al script como una cadena de caracteres, se elaborará una función que permita obtener el reverso/espejo de dicha cadena. Una vez calculado, se mostrará por pantalla dicho resultado. Por ejemplo: “HOLA en espejo es: ALOH”. **Uso exclusivo de for para la gestión de bucles.**

**F o f:** Sale del menú. Al salir del menú se debe mostrar el mensaje “Fin del programa” y finalizará el script.

**Nota:** Los argumentos con los que se invoque al script deben ser acordes con la opción del menú a elegir.

## Enunciado trabajo

Crear un script con comandos de Linux que admita tres o más argumentos; de forma que si no es así debe visualizar el mensaje *Error de sintaxis*. Una vez comprobado esto, debe presentar el siguiente menú por pantalla y programar, mediante funciones, las opciones según se detallan a continuación. Una vez terminada cada opción debe volver a mostrarse el menú y así sucesivamente hasta que escoja la opción “Fin de programa”:

*Menú*

[P] *Palíndromo*  
[D] *Divisores*  
[R] *Resumen de argumentos de entrada*  
[F] *Fin del programa*

*Opciones:*

*P o p:* Considerando el primer argumento del script como una cadena de caracteres, construir una función en bash que compruebe si dicho argumento tiene las propiedades de un palíndromo. Es decir, si se puede leer igual de izquierda a derecha que de derecha a izquierda; por ejemplo: “ala” . Una vez comprobado, se indicará mediante un mensaje si la palabra es un palíndromo o no. **No se podrá hacer uso de la orden for.**

*D o d:* Por medio de una función en bash, solicitar al usuario que introduzca un valor numérico por teclado. Una vez introducido, se calcularán los divisores primitivos (factores) de ese número, es decir, si es divisible por el conjunto numérico del 1 al 9. Una vez hechos los cálculos, se mostrará un resumen indicando por cuales es divisible. **Se debe utilizar la orden for.**

*R o r:* Suponiendo un número infinito de argumentos de entrada, y que a su vez todos son ficheros, construir una función en bash que permita: Contar cuántos de esos ficheros existen, y por otro lado, cuántos de los existentes tienen permisos de escritura. Al final, se mostrará un mensaje resumen indicando el total de ficheros recibidos, cuántos existen y cuántos tienen permisos de escritura. **No se podrá utilizar la orden if y se debe hacer uso de shift**

*F o f:* Sale del menú. Al salir del menú se debe mostrar el mensaje “Fin del programa” y finalizará el script.

*Nota:* Los argumentos con los que se invoque al script deben ser acordes con la opción del menú a elegir.



## Enunciado trabajo

Crear un script con comandos de Linux que admita dos o más argumentos; de forma que si no es así debe visualizar el mensaje *Error de sintaxis*. Una vez comprobado esto, debe presentar el siguiente menú por pantalla y programar, mediante funciones, las opciones según se detallan a continuación. Una vez terminada cada opción debe volver a mostrarse el menú y así sucesivamente hasta que escoja la opción “FIN”:

Menú            [O] Ordenados  
                  [I] Inodo  
                  [V] Visualizar  
                  [F] FIN

*Opciones:*

*O u o:* Considerando los argumentos del script como números enteros, comprueba si están o no ordenados de forma descendente, e informa mediante un mensaje de ello. Si algún argumento rompe el orden descendente no se debe continuar comprobando el resto de argumentos.

*I o i:* Solicitar al usuario que introduzca el inodo de un fichero y el nombre de un posible directorio. Comprobar que dicho directorio existe, en cuyo caso se debe mostrar el número de ficheros contenidos en ese directorio que no tienen ese mismo inodo. En el caso de que se trate de un directorio inexistente visualizar el mensaje “*Ese directorio no existe*”.

*V o v:* Mientras que el usuario no introduzca la cadena “Terminar” debes solicitarle el nombre de un fichero. Si realmente corresponde a un fichero existente, debes visualizar sus dos primeras líneas. En el caso de que no corresponda con un fichero debes mostrar el mensaje “*No es un fichero*”. **No se puede usar la orden if.**

*F o f:* Sale del menú. Al salir del menú se debe mostrar el mensaje “*Fin del programa*” y finalizar el script.

*Nota:* Los argumentos con los que se invoque al script deben ser acordes con la opción del menú a elegir.

## Enunciado trabajo

Crear un script con comandos de Linux que admita dos o más argumentos; de forma que si no es así debe visualizar el mensaje “*Error de sintaxis*”. Una vez comprobado esto, debe presentar el siguiente menú por pantalla y programar, mediante funciones, las opciones según se detallan a continuación. Una vez terminada cada opción debe volver a mostrarse el menú y así sucesivamente hasta que escoja la opción “FIN”:

Menú            [V] Vacío  
                  [C] Calcular el cubo  
                  [R] Recuento  
                  [F] FIN

Opciones:

*V o v:* Comprueba si cada argumento del script es un fichero o no. En el caso de que sea un fichero debes añadirle las últimas tres líneas del propio script que se está ejecutando. En el caso de que ese argumento no corresponda a un fichero debes visualizar el mensaje “*No es un fichero*”. **No se puede usar la orden if.**

*C o c:* Calcula y muestra por pantalla el cubo de los números dados como argumentos al script que sean pares. Si el argumento es impar debes mostrar el mensaje “*Argumento impar*”.

*R o r:* Solicita que el usuario introduzca el nombre de dos ficheros y comprueba que ambos tienen permiso de lectura. Sólo cuando se cumpla esto y mediante un solo comando debes almacenar en el fichero *recuento* el número de líneas, palabras y bytes de ambos ficheros, en cualquier otro caso se visualizará el mensaje “*Alguno o los dos ficheros no tienen permiso de lectura*”. **No se puede usar la orden if.**

*F o f:* Sale del menú. Al salir del menú se debe mostrar el siguiente mensaje: “Fin del programa” y se acabará el script.

Nota: Los argumentos con los que se invoque al script deben ser acordes con la opción del menú a elegir.

## Enunciado trabajo

Crear un script con comandos de Linux que admita dos o más argumentos; de forma que si no es así debe visualizar el mensaje “*Error de sintaxis*”. Una vez comprobado esto, debe presentar el siguiente menú por pantalla y programar, mediante funciones, las opciones según se detallan a continuación. Una vez terminada cada opción debe volver a mostrarse el menú y así sucesivamente hasta que escoja la opción “FIN”:

Menú            [R] *Recuento*  
                  [M] *Mayor/Menor*  
                  [F] *FIN*

Opciones:

*R o r*: Solicita que el usuario introduzca el nombre de dos ficheros y comprueba que ambos tienen permiso de lectura. Sólo cuando se cumpla esto y mediante un solo comando debes almacenar en el fichero *recuento* el número de líneas, palabras y bytes de ambos ficheros, en cualquier otro caso se visualizará el mensaje “*Alguno o los dos ficheros no tienen permiso de lectura*”. No se puede usar la orden *if*.

*M o m*: Visualiza por pantalla el mayor y el menor de los argumentos dados al script.

*F o f*: Sale del menú. Al salir del menú se debe mostrar el siguiente mensaje: “Fin del programa” y se acabará el script.

Nota: Los argumentos con los que se invoque al script deben ser acordes con la opción del menú a elegir.