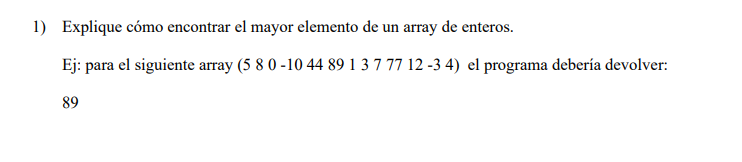
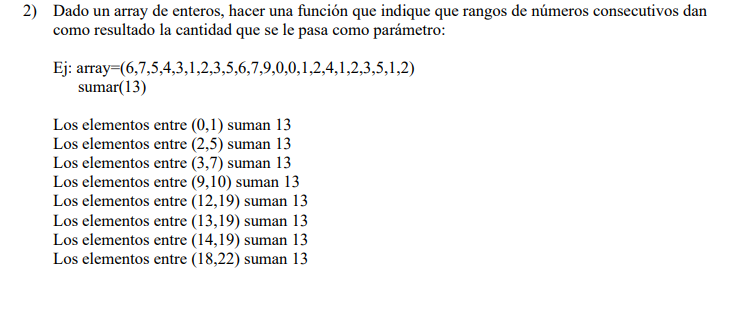
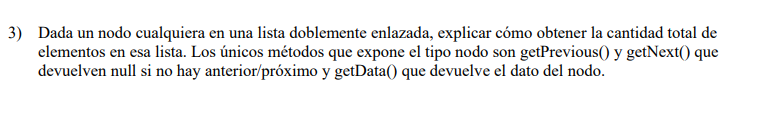


**Consideraciones sobre la resolucion en codigo de los problemas de programacion**

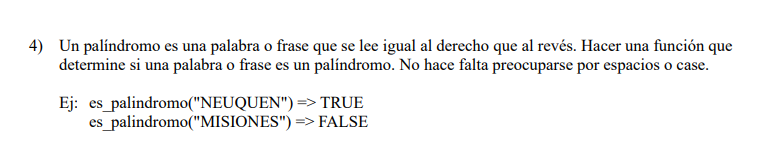


Para encontrar el mayor, al estar desordenado el array, simplemente lo recorro y voy guardando el elemento mayor.

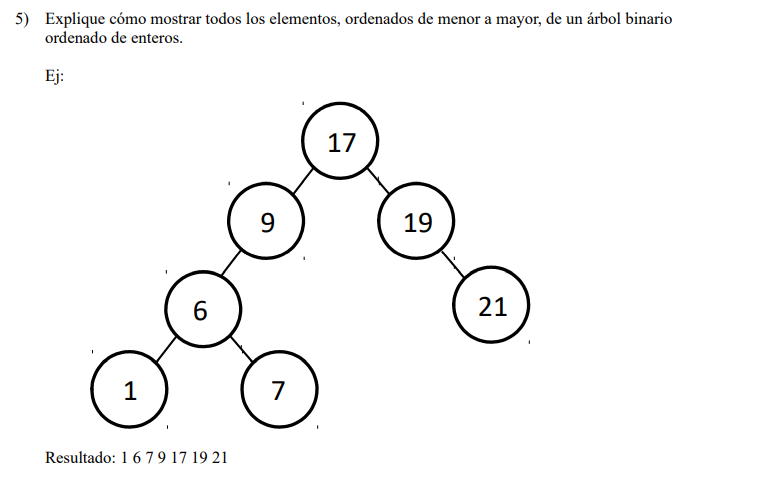
Para este problema itero el array, mientras guardo una suma parcial que inicilmente toma el valor del elemento ‘i’ del array. Comparo si la suma parcial es menor al parametro, si es asi avanzo, aumento la suma parcial con el siguiente, y un offset que me permite moverme en el rango de dicha suma de elementos. Cuando llego a una suma igual al parametro, agrego el rango entre el elemento ‘i’ del array y el offset a un nuevo array de intervalos. Si la suma supera al parametro, avanzo en el array de numeros sin crear el rango.



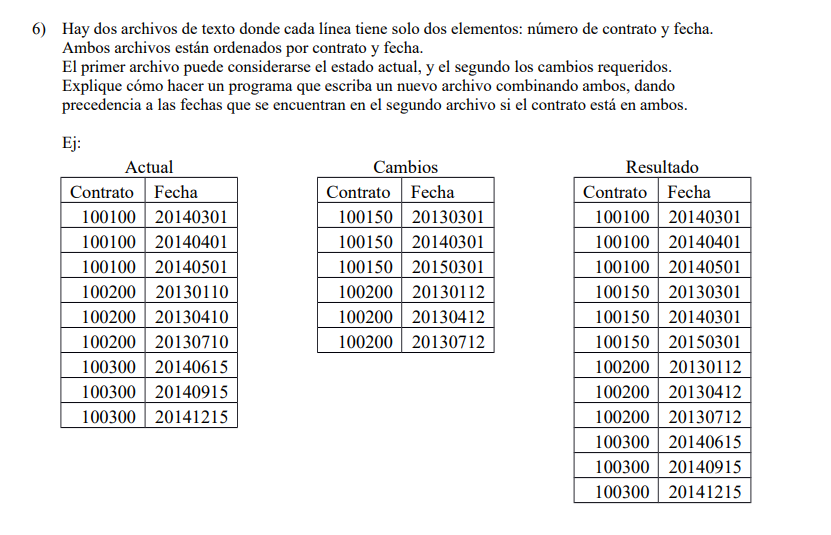
Para este problema creo una estructura de Lista, con elementos de tipo Nodo. Para obtener el tama;o de una lista, creo un metodo getTamanioLista recorriendo la misma con un Nodo auxiliar e ir sumando un contador con cada elemnto no-NULL que encuentro



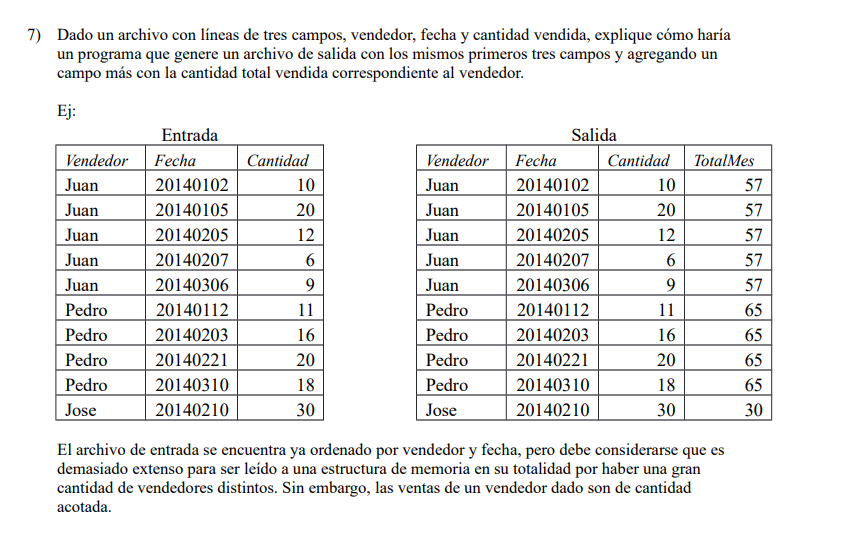
Para este problema armo una pila y una cola con los caracteres de la palabra, comparo el carácter del tope de la pila con la cabeza la cola, mientras sean iguales voy removiendo los elementos de cada colección. Si ambas coleccionas quedan vacias, la palabra es un palindromo.



Utilizaria el recprrido Inorden del arbol, es decir primero recorro los sub-arboles de la izquierda, luego los nodos padres, y por ultimo los sub-arboles de la derecha

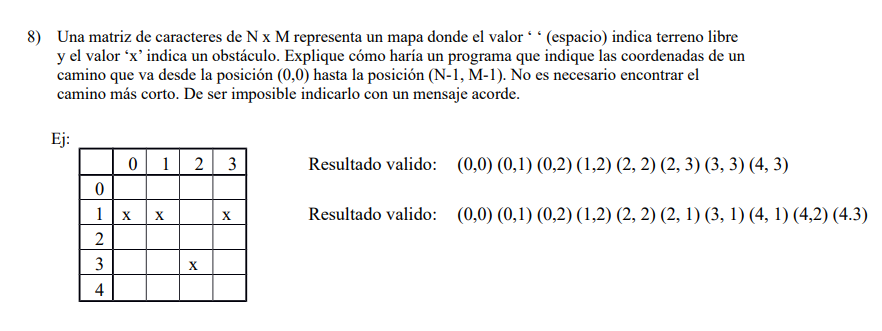


En este problema aplico el algoritmo de apareo de archivos, siempre priorizando el archivo de Cambios en caso que en ambos archivos coincida un numero de contrato determinado. Dicho algoritmo consiste en ir comparando, mientras no nos encontremos en Fin de Archivo, en ambos el numero de contrato. Avanzo en el archivo en donde dicho numero sea menor, y voy agregando el registro en el tercer archivo de apareo. Si el numero de contrato coinciden, priorizo el segundo archivo.



En este problema aplico un corte de control, para traer los totales de venta de cada vendedor y cuantos registros hay de cada uno en el archivo de Entrada. Dichas cantidades las almaceno en una lista de tipo

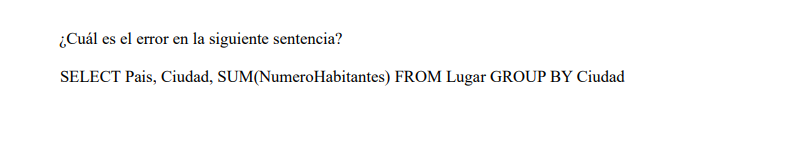
‘EntradaCantidadVentas’. Luego aplico un segundo corte de control sobre el mismo archivo, haciendo el corte con la cantidad de registros por cada vendedor, que ya almacene en la lista creada anteriormente, y voy creando cada registro sobre el nuevo archivo con la cantidad total vendida por cada vendedor

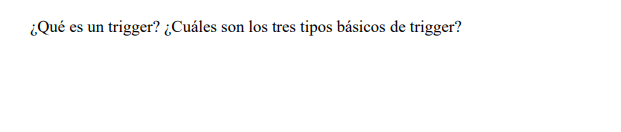


Este problema se puede resolver con una matriz que representa el mapa, y voy recorriendolo y agregado coordenadas de acuerdo a si puedo a vanzar por el mismo. En caso de no poder avanzar aplico backtracking para desahacer una coordenada (o coordenadas). Para esto creo un array de movimientos posibles dentro del tablero. Voy iterando desde la posicion (0,0) y voy verificando qué movimiento tengo disponible, si la siguiente disponible no está visitada, la puedo realizar porque está dentro de los límites y no contenga una ‘X’ seteo la posicion actual como visitada, avanzo en esa Coordenada disponible y agrego dicha cooredenada a un array de camino que contiene las coordenadas del mismo. A su vez en dicha coordenada que avanzo, llamo iteritivamente al método en cuestión y realizo los mismos pasos. En caso de encontrarse en una coordenada donde ya no tengo movimientos para realizar, aplico Backtracking sacando el seteo de visitada a la posicion Actual, y quitando del array de camino la coordenada que habia avanzado de modo que pueda retroceder.

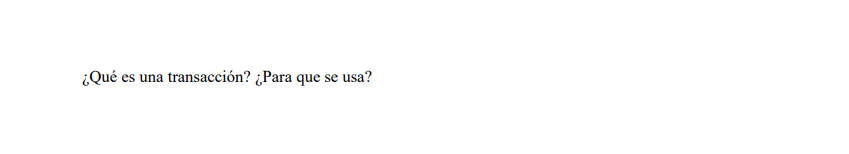


Una primary Key, es un campo clave que identifica unívocamente a un registro de una tabla. Una forign key, o clave foránea, es una clave que identifica a un campo de otra tabla

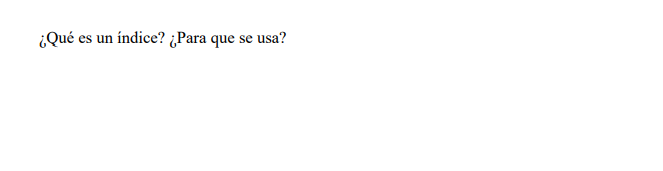




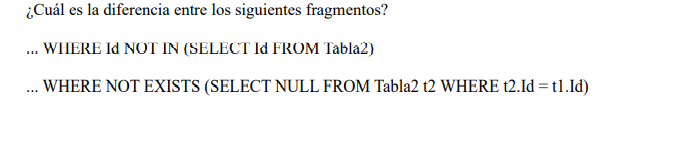
Un trigger es un objeto en SQL que se dispara cuando ocurre algún evento en una tabla. Los 3 tipos de triggers más comunes son de tipo INSERT, DELETE,UPDATE.



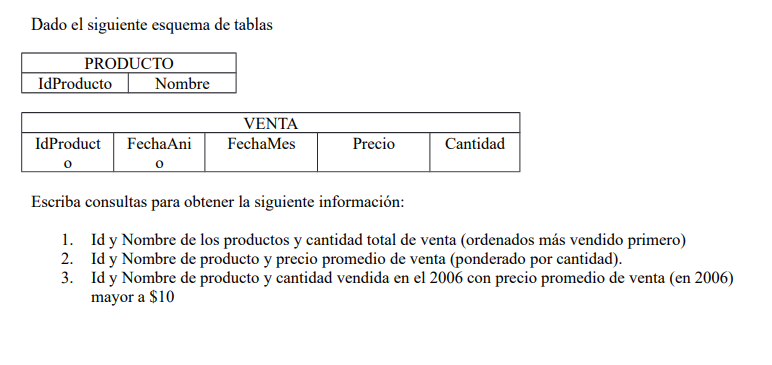
Una transacción es una unidad de trabajo que se ejecuta de forma indivisible, es decir son un conjunto de sentencias SQL que se ejecutan todas juntas. Cumplen con las propiedades ACID: Atomicidad, es decir que se ejecuta todo, o se rechaza la operación; Consistencia, luego de ejecutada la transacción el estado final de la base concuerda con el esquema de datos definido; Aislamiento, as transacciones se ejecutan de forma independiente una de otra; Durabilidad, los cambios producidos por las transacciones quedan persistentes en el tiempo a pesar de algún posible error en la base.



Los índices me permiten el acceso a un registro de forma más rápida, al crear índices por campos las búsquedas por dicho campo se realizan como un acceso directo y no recorriendo la tabla de forma secuencial



Si bien tienen el mismo propósito de buscar valores Id de la primera tabla que no existan en la segunda, la primera consulta no computa los valores NULL, mientras que la segunda sí.



1)

select p.IdProducto as Id,

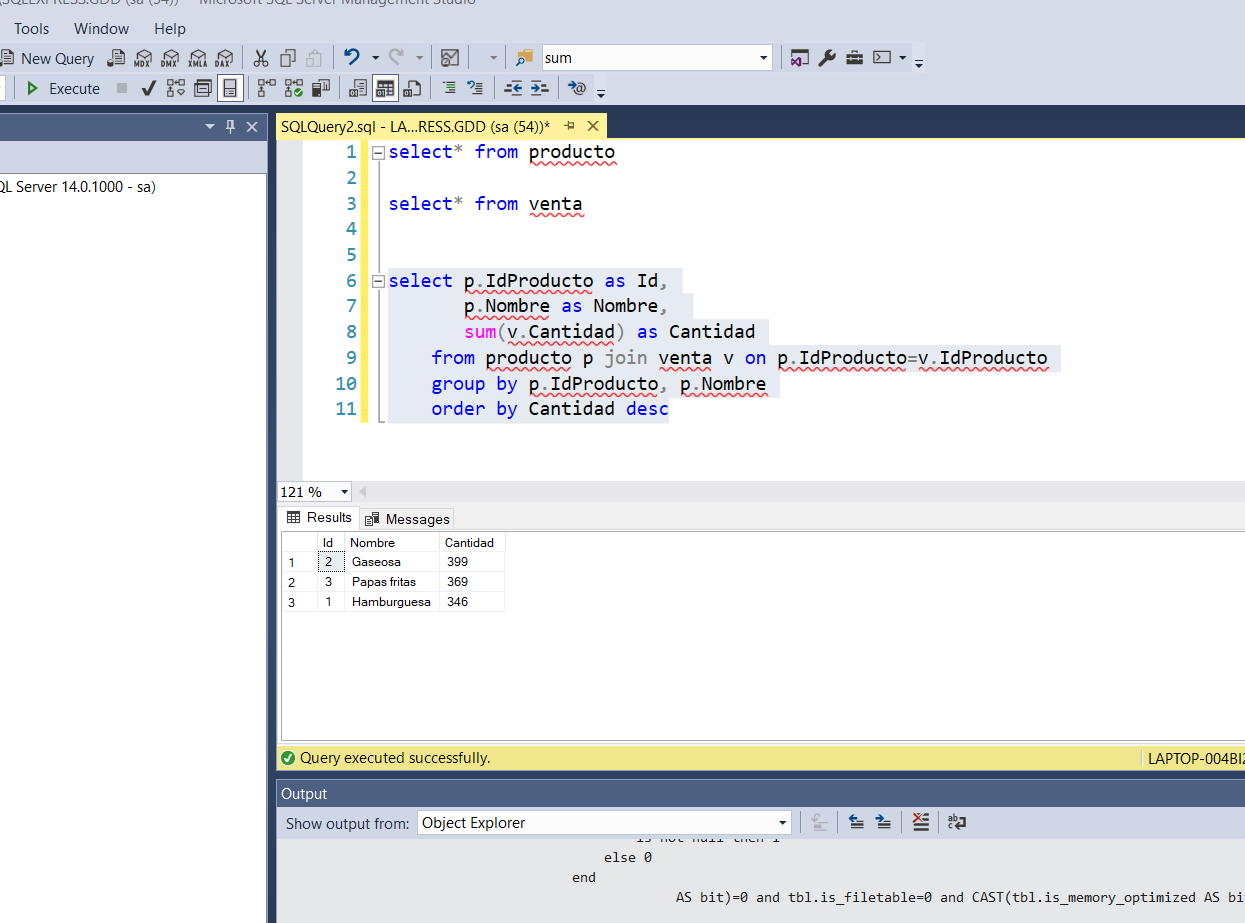
p.Nombre as Nombre,

sum(v.Cantidad) as Cantidad

from producto p join venta v on p.IdProducto=v.IdProducto

group by p.IdProducto, p.Nombre

order by Cantidad desc



2)

select p.IdProducto as Id,

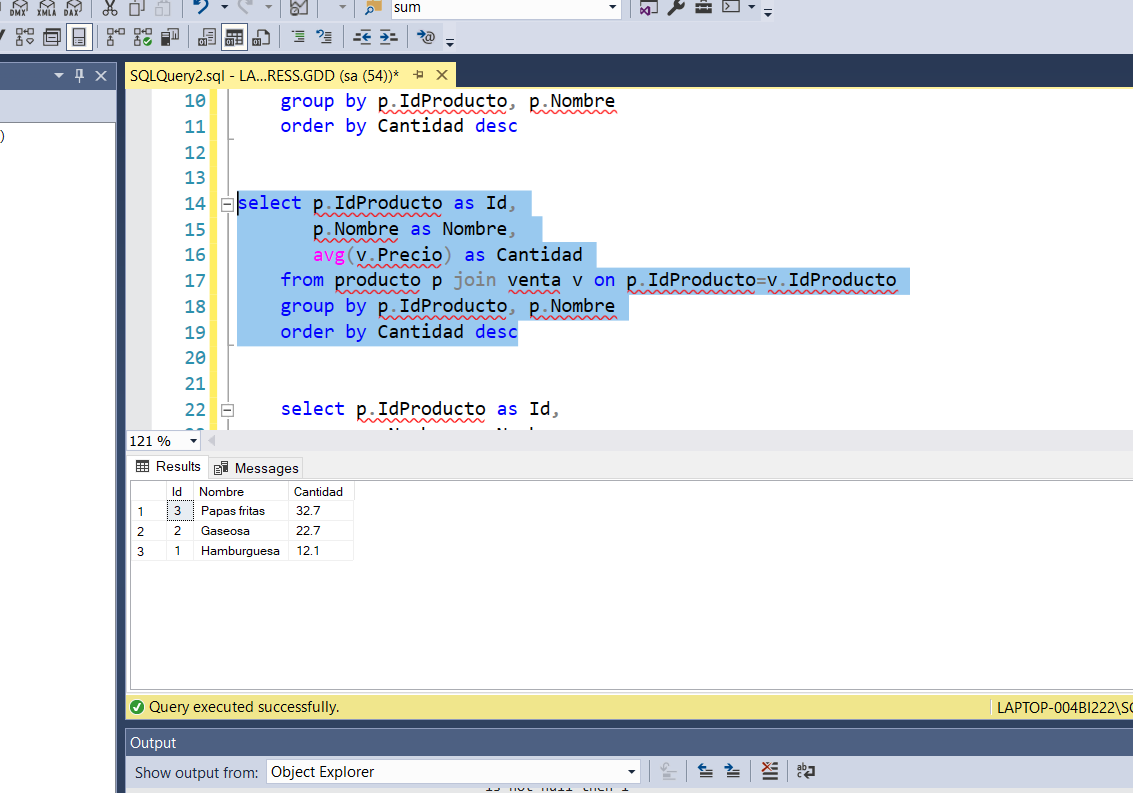
p.Nombre as Nombre,

avg(v.Precio) as Cantidad

from producto p join venta v on p.IdProducto=v.IdProducto

group by p.IdProducto, p.Nombre

order by Cantidad desc



3)

select p.IdProducto as Id,

p.Nombre as Nombre,

sum(v.Cantidad) as Cantidad,

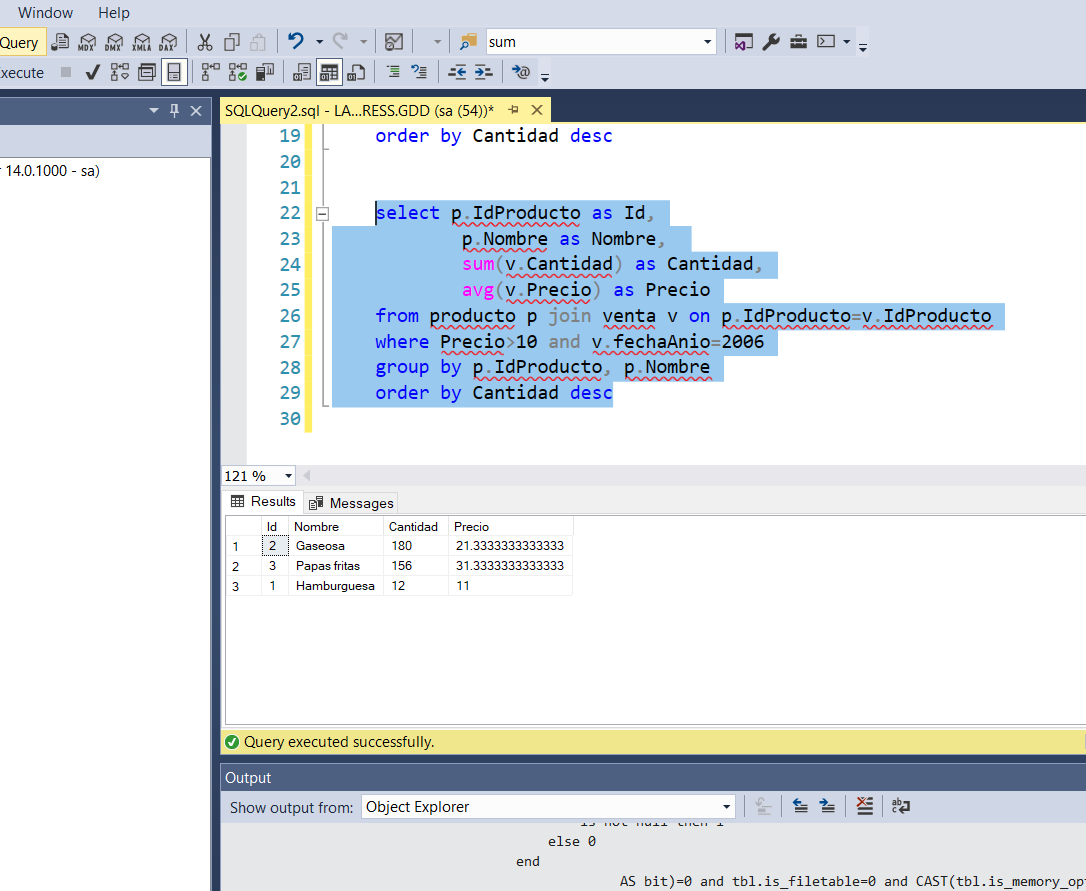
avg(v.Precio) as Precio

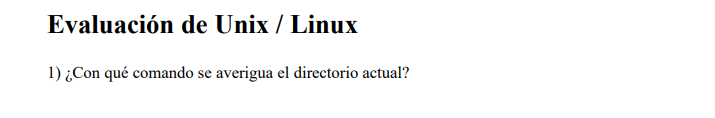
from producto p join venta v on p.IdProducto=v.IdProducto

where Precio>10 and v.fechaAnio=2006

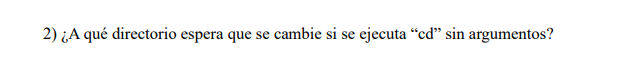
group by p.IdProducto, p.Nombre

order by Cantidad desc

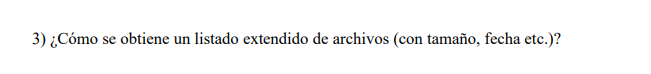




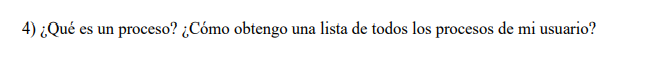
PWD



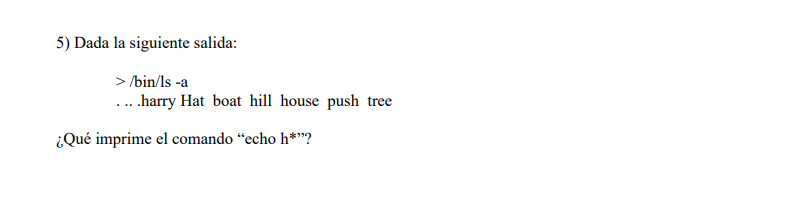
A root

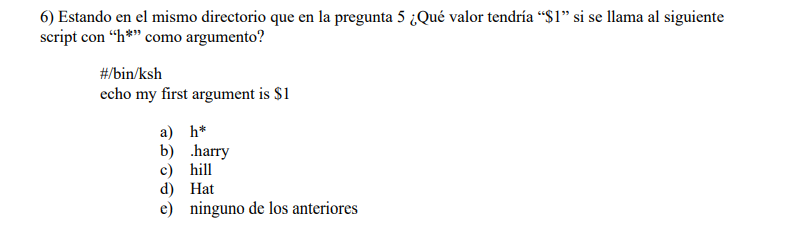


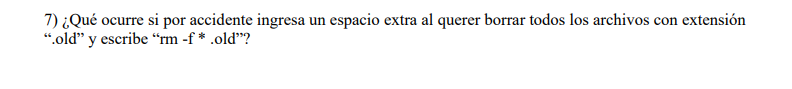
ls -xlcls



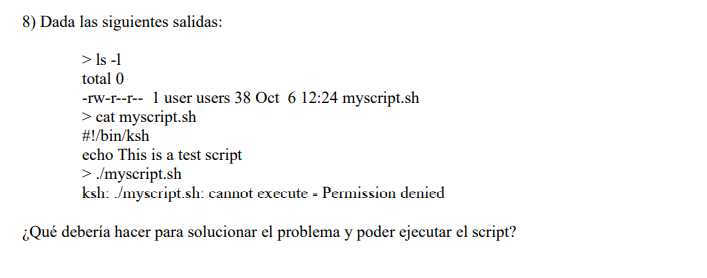
Un proceso es un programa en ejecución, es decir es la entidad a la que el Sistema Operativo asigna recursos como ID, un estado, PC o Program Counter que es la dirección de la próxima instrucción a ejecutar, archivos, memoria, dispositivos de E/S asignados al mismo. Esta información se guarda en una estructura llamada PCB, o Process Control Block.



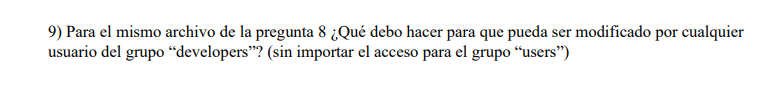




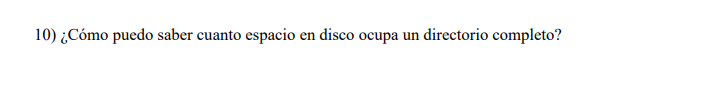
Se borran todos los archivos del directorio actual



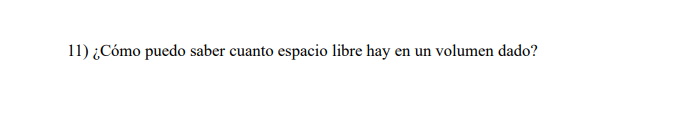
Se deben asignar permisos con el comando ‘chmod’, ya sea para cualquier usuario: ‘chmod 0755 script.sh’ o para el usuario ‘chmod 0700 script.sh’



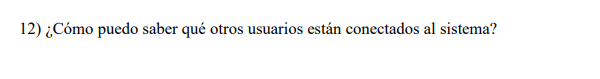
Mediante el comando ‘chmod g+w scipt.sh’



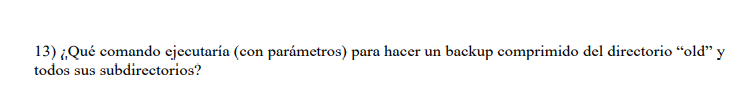
Comando ‘du -h’



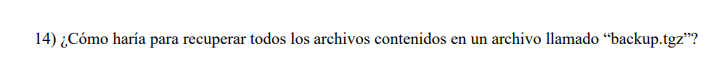
Comando ‘df’



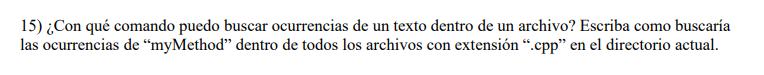
Comando ‘who’



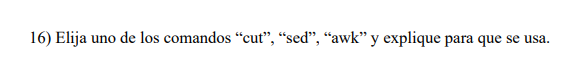
Mediante el comando ‘tar -czvf backup.tar.gz /old’



Mediante el comando ‘tar -xzvf backup.tgz’



Mediante el comando grep: ‘grep myMethod \*.cpp’



El comando cut se utiliza para cortar el contenido de un archivo, indicándole el rango de bytes a cortar, o definiendo un delimitador y en base a ese delimitador un rango. Por ejemplo

Defino este archivo:

localhost:~/hola# cat > ejemplo.cpp

hola.chau.hastaLuego

hello.bye.goodMorning

Si quiero mostrar los caracteres 1 y 2:

localhost:~/hola# cut -c 1-2 ejemplo.cpp

ho

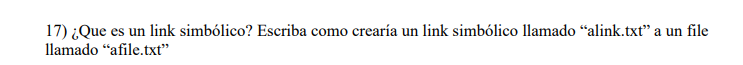
he

Si quiero obtener rangos definidos mediante un delimitador por ejemplo el ‘.’

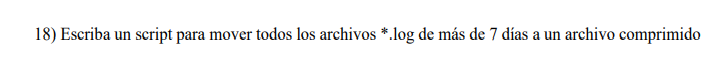
localhost:~/hola# cut -f 1-2 -d"." ejemplo.cpp

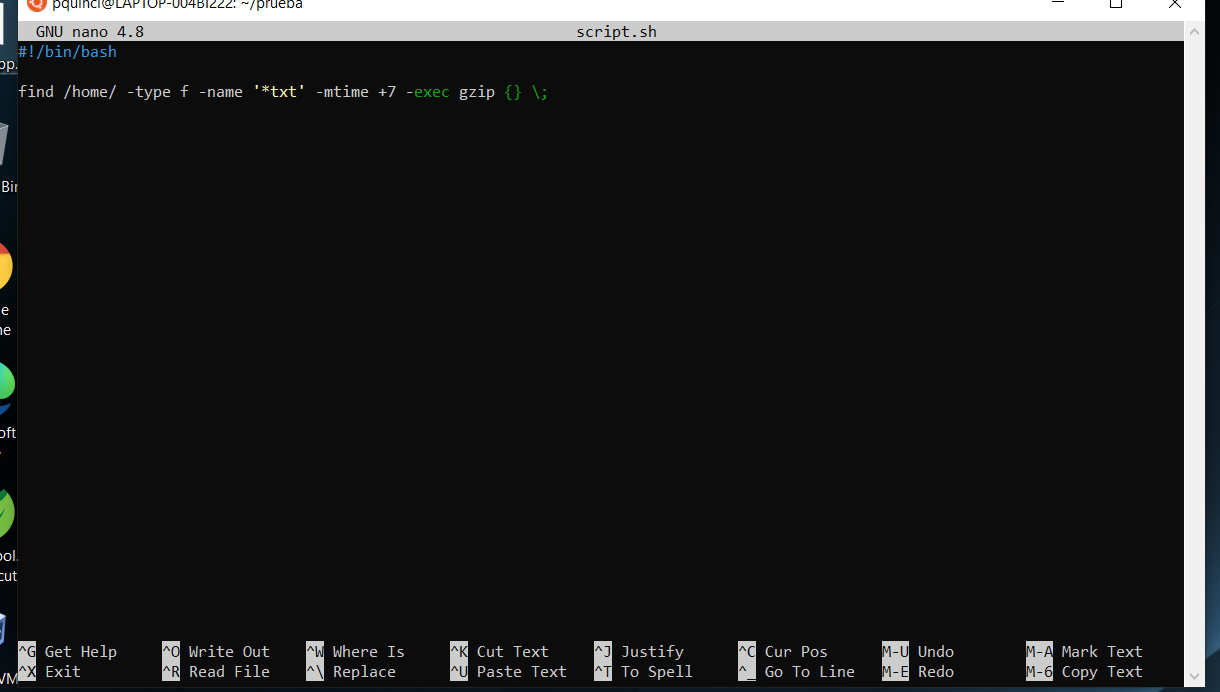
hola.chau

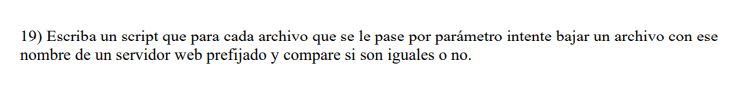
hello.bye

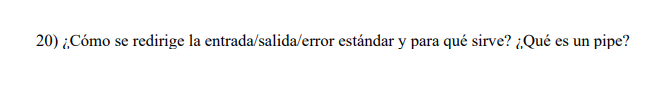


Es un tipo de archivo que apunta a otro, similar a los accesos directos. Para crear un symlink a dicho archivo debería usar el comando ln: ‘ln alink.txt afile.txt’





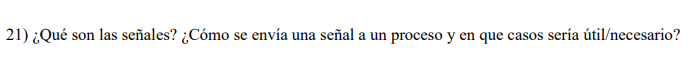




Mediante el comando ‘>’. Por ejemplo, quiero listar un directorio y enviarlo a un archivo:

‘ls -la > salida.txt’. Si quiero redirigir un error: ‘ls -la /root > errores.txt’ este ejemplo aplica para un ususario que no sea root y no tenga permisos para acceder a dicho directorio.

Un pipe comunica procesos, mediante lo cual la salida de un proceso corresponde a la entrada del siguiente. Por ejemplo, para listar archivos que contengan una cierta palabra: ‘ls | grep hola’ En este caso la salida del grep será la entrada del comando ‘ls’.



Una señal es un aviso que se le da a algún proceso sobre un evento que sucedió, las señales síncronas son por ejemplo aquellas recibidas por accesos ilegales a memoria, división por cero, etc. Y las señales asíncronas son las que se reciben de forma externa al proceso, por ejemplo cuando un usuario presiona Ctrl+C .