FOD - Repaso

Las respuestas correctas suman un punto, las incorrectas restan 0,5. Sin responder es neutro. Cada ejercicio tiene un y solo un inciso correcto.

1) Un archivo serie

a. Tiene acceso secuencial y orden de búsqueda lineal

b. Tiene acceso secuencial indizado y orden de búsqueda logarítmico

C. Tiene acceso secuencial y orden de búsqueda constante

d. Tiene acceso secuencia indizado y orden de búsqueda lineal

e. Todas las anteriores

f. Algunas de las anteriores

g. Ninguna de las anteriores

2) Un proceso de actualización de un archivo con n archivos

a. Se puede aplicar con todos los archivos ordenados por algún criterio

b. Se puede aplicar con todos los archivos desordenados

C. Se puede aplicar con algunos de los archivos ordenados por algún criterio y el resto desordenados

d. Se puede aplicar con cada uno de los n archivos ordenados por diferentes criterios.

e. Todas las anteriores son válidas

f. Algunas de las anteriores son válidas

g. Ninguna de las anteriores son válidas

3) Si se realiza la baja de un registro en un archivo secuencial indizado

a. Es igual de eficiente hacer una baja lógica que una baja física

b. Es más eficiente hacer una baja lógica que una baja física

c. Es menos eficiente hacer una baja lógica que una baja física

d. Ninguna de las anteriores

4) Un archivo con registros de longitud variable

a. Ocupa más espacio que el mismo archivo con registros de longitud fija

b. Ocupa el mismo espacio que el mismo archivo con registros de longitud fija

c. Ocupa menos espacio que el mismo archivo con registros de longitud fija

d. No tengo datos suficientes para contestar.

5) Para poder realizar un algoritmo de corte de control sobre un archivo

a. Es suficiente que el archivo esté ordenado por al menos un criterio

b. Es necesario que el archivo esté ordenado por al menos un criterio

c. Es suficiente que el archivo esté ordenado por al menos dos criterios

d. Es necesario que el archivo esté ordenado por al menos dos criterios

e. Todas las anteriores con válidas

f. Algunas de las anteriores son válidas

g. Ninguna de las respuestas anteriores

6) Una clave secundaria

a. Identifica unívocamente un elemento del archivo

b. Identifica unívocamente varios elementos de un archivo

c. Está constituida por al menos un atributo del archivo donde está definida

d. Todas las anteriores con válidas

e. Algunas de las anteriores son válidas

f. Ninguna de las respuestas anteriores vale

7) Un Índice

a. Se implementa con un vector

b. Se implementa con un árbol binario

c. Se implementa con un árbol b

d. Se implementa con un árbol b+

e. Todas las anteriores con válidas

f. Algunas de las anteriores son válidas

g. Ninguna de las respuestas anteriores vale

8) Un árbol binario paginado

a. Puede desbalancearse

b. Puede estar balanceado

c. Puede no estar balanceado

d. Todas las anteriores con válidas

f. Algunas de las anteriores son válidas

g. Ninguna de las respuesta anteriores vale

9) Dado un índice de un archivo. Para un momento determinado:

a. Si se implementó con árbol b tiene mayor altura que si se implementó con un árbol B+

b. Si se implementó con un árbol b tiene igual altura que si se implementó con un árbol b+

c. Si se implementó con un árbol b tiene menor altura que si se implementó con un árbol b+

d. No tengo datos suficientes para responder

10) Un árbol multicamino

a. Es un árbol binario

b. Es un árbol AVL

c. Es un árbol B+

d. Todas las anteriores con válidas

f. Algunas de las anteriores son válidas

g. Ninguna de las respuestas anteriores vale

11) El acceso directo a un elemento de datos es

a. Más eficiente que el acceso secuencial

b. Más eficiente que el acceso secuencial indizado

c. Más eficiente que utilizar un árbol B+

d. Todas las anteriores son válidas

f. Algunas de las anteriores son válidas

g. Ninguna de las anteriores son válidas

12) Una función de hash en teoría debe ser

a. Uniforme y aleatoria

b. Uniforme pero no aleatoria

c. Aleatoria pero no uniforme

d. No debe ser ni uniforme ni aleatoria

13) La densidad de empaquetamiento

a. Mide la relación entre espacio disponible y espacio utilizado

b. Mide la relación entre la cantidad de registros que tiene un archivo y la capacidad de almacenamiento disponible para el mismo

c. Es el cociente entre el número de registros del archivo y el espacio disponible

d. Todas las anteriores son válidas

e. Algunas de las anteriores son válidas

f. Ninguna de las anteriores es válida

g. Se realiza una operación de escritura y una de lectura

14) Cuando se produce una inserción en hash estático, sobre el archivo de datos

a. Se realiza al menos una operación de lectura y al menos una de escritura

C. Se realiza una operación de lectura y al menos una de escritura

d. Se realiza una operación de escritura y al menos una de lectura

15) En hashing extensible, cuando la Densidad de empaquetamiento tiende a 1:

a. Se debe pasar de hash estático a dinámico

b. Se debe reformular el tamaño del nodo

C. Se deben aumentar la cantidad de registros por nodo.

d. Algunas de las anteriores

e. Ninguna de las Anteriores.

FOD REPASO

1) Un archivo directo
a. Tiene acceso directo y orden de búsqueda lineal
b. Tiene acceso directo y orden de búsqueda logarítmico
c. Tiene acceso directo y orden de búsqueda exponencial
d. Tiene acceso secuencial indizado y orden de búsqueda lineal
e. Todas las anteriores
f. Algunas de las anteriores
g. Ninguna de las anteriores

2) Un proceso de merge de n archivos, para que sea lo más eficiente posible en términos de performance/rendimiento
a. Se debe aplicar con todos los archivos ordenados por algún criterio
b. Se debe aplicar con todos los archivos desordenados
c. Se debe aplicar con algunos de los archivos ordenados por algún criterio y el resto desordenados
d. Se debe aplicar con cada uno de los n archivos ordenados por diferentes criterios
e. Todas las anteriores
f. Algunas de las anteriores
g. Ninguna de las anteriores

3) Si se realiza la baja de un registro en un archivo directo
a. Es igual de eficiente hacer una baja lógica que una baja física
b. Es más eficiente hacer una baja lógica que una baja física considerando el espacio ocupado
c. Es menos eficiente hacer una baja lógica que una baja física considerando el rendimiento/performance
d. Ninguna de las anteriores

4) En un archivo directo con registros de longitud variable

a. Puede ocupar más espacio que el mismo archivo con registros de longitud fija
b. Puede ocupar el mismo espacio que el mismo archivo con registros de longitud fija
c. Ocupa menos espacio que el mismo archivo con registros de longitud fija

d. Ninguna de las anteriores

5) Para poder realizar un algoritmo de bajas sobre un archivo

a. Es suficiente que el archivo este ordenado por al menos un criterio

b. Es necesario que el archivo este ordenado por al menos un criterio

c. Es suficiente que el archivo este ordenado por al menos dos criterios

d. Es necesario que el archivo este ordenado por al menos dos criterios

E. Todas las anteriores son validas

f. Algunas de las anteriores son validas

g. Ninguna de las respuestas anteriores

6) Una clave

a. Identifica unívocamente un elemento del archivo

b. Identifica varios elementos de un archivo

c. Esta constituida por al menos dos atributos del archivo donde está definida

d. Todas las anteriores

e. Algunas de las anteriores

f. Ninguna de las anteriores

7) Un índice secundario implementado con un árbol binario

a. Puede desbalancearse

b. Puede estar balanceado

c. Puede balancearse

d. Se desbalancea fácilmente

e. Todas las anteriores

f. Algunas de las anteriores

g. Ninguna de las anteriores

8) Un árbol

a. Puede desbalancearse

B. Puede estar balanceado

c. Esta balanceado en altura

d. Puede no estar balanceado

e. Todas las anteriores

f. Algunas de las anteriores

g. Ninguna de las anteriores

9) Dado un índice de un archivo con registros de longitud variable

a. Sus registros pueden ser de longitud variable

b. Sus registros deben ser de longitud variables

c. Sus registros pueden ser de longitud fija

d. Sus registros son de longitud fija

10) Un árbol

a. Puede ser árbol binario y AVL

b. Puede ser B* y B+

c. Puede ser B y B+

d. Todas las anteriores son válidas

e. Algunas de las anteriores son válidas

f. Ninguna de las respuestas anteriores vale

11) Una función Hash uniforme y aleatoria

a. Tiene en promedio 1, accesos a registros

b. Tiene en promedio entre 1 y 2, accesos a registros

c. Tiene en promedio menor a 1, accesos de a registros

d. Ninguna de las anteriores

12) La densidad de empaquetamiento

a. Se modifica con cada alta o baja, usando hashing extensible

b. Se modifica con cada alta, usando hashing extensible

c. Se modifica con cada baja, usando hashing extensible

d. No se modifica con altas o bajas, usando hashing extensible

e. Todas las anteriores

f. algunas de las anteriores

g. ninguna de las anteriores

13) Cuando se produce una inserción en hash estático sobre el archivo de datos

a. Se realiza una operación de escritura y una de lectura

b. Se realiza al menos una operación de lectura y al menos una de escritura

c. Se realiza una operación de lectura y al menos una de escritura

d. Se realiza una operación de escritura y al menos una de lectura

14) Si hubo colisión con hashing dinámico

a. Ocurre overflow

b. No ocurre overflow

c. Se trata con un algoritmo específico de tratamiento de colisiones

d. Algunas de las anteriores

e. Ninguna de las anteriores

FOD - Examen de trabajos prácticos - Segunda Fecha - 4/07/23 Tema 1

1. Archivos Secuenciales

Suponga que tiene un archivo con información de los partidos de los últimos años de los equipos de primera división del fútbol Argentino. Dicho archivo contiene: código de equipo, nombre de equipo, año, código de torneo, código de equipo rival, goles a favor, goles en contra, puntos obtenidos (0, 1 o 3 dependiendo de si perdió, ganó o empató el partido). El archivo está ordenado por los siguientes criterios: año, código de torneo y código de equipo.

Se le solicita definir las estructuras de datos necesarias y escribir el módulo que reciba el archivo y genere un informe por pantalla con el siguiente formato de ejemplo:

Informe resumen por equipo del fútbol Argentino

Año 1

cod torneo 1

cod_equipo 1 nombre equipo 1

cantidad total de goles a favor equipo 1

cantidad total de goles en contra equipo 1

diferencia de gol (resta de goles a favor - goles en contra) equipo 1

cantidad de partidos ganados equipo 1

cantidad de partidos perdidos equipo 1

cantidad de partidos empatados equipo 1

cantidad total de puntos en el torneo equipo 1

---

cod_equipo n nombre equipo n

idem anterior para equipo n

El equipo "nombre equipo fue campeón del torneo codigo de torneo 1 del año 1

cod_torneo n

Idem anterior para cada equipo en el torneo n

El equipo "nombre equipo fue campeón del torneo código de torneo n del año 1

Año n

Idem anterior para cada torneo del año n

Nota: se asume que por torneo hay un único equipo campeón con mayor puntaje. 2- Árboles

2. Arboles

Dado el siguiente árbol B+ de orden 4 y con política de resolución de underflows a derecha, realice las siguientes operaciones indicando lecturas y escrituras en el orden de ocurrencia. Además, debe describir detalladamente lo que sucede en cada operación. +58, -403, +260, -550

4:0 (240) 1 (403) 2 (500) 3

0: (3)(45) (60) 1 1: (240) (255)(360) 2 2: (409)(420) 3 3: (550)-1

FOD -Examen de trabajos prácticos- Tercera Fecha - 8/08/23 - Tema 1

1. Archivos Secuenciales

Una empresa dedicada a la venta de golosinas posee un archivo que contiene Información sobre los productos que tiene a la venta. De cada producto se registran los siguientes datos: código de producto, nombre comercial, precio de venta, stock actual y stock mínimo.
La empresa cuenta con 20 sucursales. Diariamente, se recibe un archivo detalle de cada una de las 20 sucursales de la empresa que indica las ventas diarias efectuadas por cada sucursal. De cada venta se registra código de producto y cantidad vendida. Se debe realizar un procedimiento que actualice el stock en el archivo maestro con la información disponible en los archivos detalles y que además informe en un archivo de texto aquellos productos cuyo monto total vendido en el día supere los $10.000. En el archivo de texto a exportar, por cada producto incluido, se deben informar todos sus datos. Los datos de un producto se deben organizar en el archivo de texto para facilitar el uso eventual del mismo como un archivo de carga.
El objetivo del ejercicio es escribir el procedimiento solicitado, junto con las estructuras de datos y módulos usados en el mismo.

Notas:

- Todos los archivos se encuentran ordenados por código de producto.
- En un archivo detalles pueden haber 0, 1 o N registros de un producto determinado.
- Cada archivo detalle solo contiene productos que seguro existen en el archivo maestro.
- Los archivos se deben recorrer una sola vez. En el mismo recorrido, se debe realizar la actualización del archivo maestro con los archivos detalles, así como la generación del archivo de texto solicitado.

2 - Árboles

Dado el siguiente árbol B de orden 4 y con política de resolución de underflows derecha o izquierda, realice las siguientes operaciones indicando lecturas y escrituras en el orden de ocurrencia. Además, debe describir detalladamente lo que sucede en cada operación. +410, -200, -500, -100

2:0 (100) 4 (300) 1 (600) 3

0: (10) 4: (200) 1: (400) (450) (500) 3: (700)

FOD -Examen de trabajos prácticos- Segunda fecha 24/06/2024

Escribir claramente en la primera hoja del examen: legajo, apellido y nombre, turno (MM, MT) y temas que rinde (por su número). En cada hoja indicar: legajo/ apellido y nombre y número de hoja/total.

1. Archivos Secuenciales

Se desea automatizar el manejo de información referente a los casos positivos de dengue para la Provincia de Buenos Aires. Para esto se cuenta con un archivo maestro que contiene la siguiente información: código de municipio, nombre municipio y cantidad de casos positivos. Dicho archivo está ordenado por código de municipio.

Todos los meses se reciben 30 archivos que contienen la siguiente información: código de municipio y cantidad de casos positivos. El orden de cada archivo detalle es por código de municipio. En cada archivo puede venir información de cualquier municipio, un municipio puede aparecer cero una o más de una vez en cada archivo.

Realice el sistema completo que permita la actualización de la información del archivo maestro a partir de los archivos detalle recalculando la cantidad de casos positivos e informando por pantalla aquellos municipios (código y nombre) donde la cantidad total de casos positivos es mayor a 15.

Nota: cada archivo debe recorrerse una única vez.
Nota 1: Los nombres de los archivos deben pedirse por teclado. Se puede suponer que los nombres ingresados corresponden a archivos existentes.
Nota 2: El informe debe incluir cualquier municipio que cumpla la condición, independientemente de si se actualiza o no.

2. Árboles

Considere un árbol B de orden 5 que representa un índice primario a un archivo de datos que contiene información de los clientes de un supermercado. Dicho índice proporciona acceso indizado por el número de cliente. A continuación se presenta el estado del árbol B en un momento determinado:

8: 2 (150) 7

2: 0 (120) 3 7: 4 (210)6(300)1(600)9

0: (30) 3: (130) 4: (160)(170) 6: (220)(230)(240)(250) 1: (400)(500) 9:(700)

Resuelva los siguientes incisos:

a) Definir en Pascal los tipos necesarios para definir la estructura de un árbol B (índice) con las características indicadas.
b) Realizar el alta de la clave 280 sobre el árbol B dado. Graficar el árbol B resultante. Indicar los nodos leídos y escritos. Explicar las decisiones tomadas para resolver la operación.
c) Realizar la baja de la clave 120 sobre el árbol resultante del punto b, considerando política derecha para la resolución de underflows. Graficar el árbol B resultante. Indicar los nodos leídos y escritos. Explicar las decisiones tomadas para resolver la operación.

FOD – Teórico 11-07-2024 – Nombre y Apellido:.....

Las respuestas correctas suman un punto, las incorrectas restan 0,5. Sin responder es neutro. Cada ejercicio tiene un y solo un inciso correcto. Tema 1

1. Un archivo secuencial
   a. Puede tener acceso Directo y orden de búsqueda lineal
   b. Puede tener acceso Directo y orden de búsqueda logarítmico
   c. Puede tener acceso Directo y orden de búsqueda exponencial
   d. Tiene acceso secuencial indizado y orden de búsqueda lineal
   e. Todas las anteriores
   f. Algunas de las anteriores
   g. Ninguna de las anteriores
2. Un proceso de actualización maestro-detalle de archivos
   a. Se debe aplicar con todos los archivos ordenados por algún criterio
   b. Se puede aplicar con todos los archivos desordenados
   c. Se puede aplicar con algunos de los archivos ordenados por algún criterio y el resto desordenados
   d. Se puede aplicar con cada uno de los archivos ordenados por diferentes criterios.
   e. Todas las anteriores
   f. Algunas de las anteriores
   g. Ninguna de las anteriores
3. La búsqueda binaria es aplicable a
   a. Archivos ordenados con registros de longitud fija
   b. Archivos ordenados con registros de longitud variable
   c. Archivos desordenados con registros de longitud fija
   d. Archivos desordenados con registros de longitud variable
   e. Hay más de una opción correcta
   f. Ninguna opción es correcta
4. Un archivo de datos:
   a. Siempre es conveniente que sea de registros de longitud fija
   b. Siempre es conveniente que sea de registros de longitud variable
   c. Con registros de longitud fija, se puede recuperar un registro en un solo acceso a disco.
   d. Algunas de las opciones son correctas
   e. Ninguna de las opciones son correctas.
5. Para poder realizar un algoritmo de bajas sobre un archivo
   a. El archivo puede estar ordenado por al menos un criterio
   b. Es necesario que el archivo esté ordenado por al menos un criterio
   c. El archivo puede estar ordenado por al menos dos criterios
   d. Es necesario que el archivo esté ordenado por al menos dos criterios
   e. Todas las anteriores con válidas
   f. Algunas de las anteriores son válidas
   g. Ninguna de las respuestas anteriores
6. Una clave secundaria
   a. Identifica unívocamente un elemento del archivo
   b. Identifica varios elementos de un archivo
   c. Está constituida por al menos dos atributos del archivo donde está definida
   d. Todas las anteriores
   e. Algunas de las anteriores
   f. Ninguna de las anteriores
7. Un árbol
   a. Puede no estar completamente balanceado
   b. Puede estar balanceado
   c. Puede no estar balanceado
   d. Se desbalancea fácilmente
   e. Todas las anteriores
   f. Algunas de las anteriores
   g. Ninguna de las anteriores

8. Sea un arbol B de orden 200, el nodo X tiene 99 elementos, si se borra un elemento de dicho nodo, que es un nodo hoja o terminal:
   a. Solo se borra el elemento
   b. Se produce un underflow y se debe redistribuir con el nodo x-1 o x+1
   c. Se produce un underflow y se debe concatenar con el nodo x-1 o x+1
   d. Todas las anteriores
   e. Algunas de las anteriores
   f. Ninguna de las anteriores
9. Un arbol B+ de orden M
   a. Es un arbol multicamino
   b. Es un arbol balanceado
   c. Es un arbol en que cada nodo (salvo la raíz) tiende a llenarse en al menos 2/3
   d. Todas las anteriores
   e. Algunas de las anteriores

10. Un árbol balanceado que crece de abajo hacia arriba
   a. Puede ser árbol binario y AVL
   b. Puede ser B* y B+
   c. Puede ser B y B+
   d. Todas las anteriores
   e. Algunas de las anteriores
   f. Ninguna de las respuestas anteriores

11. La doble dispersión
   a. Aplica siempre 2 funciones de dispersión
   b. Se usa en casos de que la densidad de empaquetamiento es mayor a 1
   c. Requiere que la densidad de empaquetamiento sea entre 0,5 y 1
   d. Se puede usar en caso de saturación
   e. Todas las anteriores
   f. Algunas de las anteriores
   g. Ninguna de las anteriores

12. Una función de hashing aceptable
   a. Tiene un promedio 1, de acceso a registros
   b. Tiene un promedio entre 1 y 3, de acceso a registros
   c. Tiene un promedio menor a 1, de acceso a registros
   d. Ninguna de las anteriores
13. La densidad de empaquetamiento extensible
   a. Se modifica con cada alta o baja, utilizando hashing extensible
   b. Se modifica con cada alta, utilizando hashing extensible
   c. Se modifica con cada baja, utilizando hashing extensible
   d. No se modifica con altas o bajas, utilizando hashing extensible
   e. Todas las anteriores
   f. Algunas de las anteriores
   g. Ninguna de las anteriores
14 Si se usa una política de hash
   a. Es posible utilizarla sobre un archivo de registros de longitud variable
   b. Es posible utilizarla sobre un archvio de registros de longitud fija con campos de long. Variable.
   c. La función de Hash es el parámetro fundamental del método
   d. Ninguna de las opciones es válida
15. Una colisión en hash estático
   a. Es inevitable que pueda ocurrir
   b. Genera una utilización desproporcionada de los métodos de tratamiento de desbordes
   c. Es el peor parámetro de hashing
   d. Ninguna de las anteriores

FOD – Teórico 12-07-2024 – Nombre y Apellido:..... [redacted] ...

Las respuestas correctas suman un punto, las incorrectas restan 0,5. Sin responder es neutro. Cada ejercicio tiene un y solo un inciso correcto. TEMA 1

1. Cuando se realiza un alta en un arbol b:
   a. Se debe realizar en un nodo interno del arbol
   b. Se puede realizar en un nodo interno del arbol
   c. Puede llegar a necesitar una división del nodo raíz
   d. Es necesario llegar a un nodo hoja
   e. Algunas de las anteriores es valida
   f. Ninguna de las anteirores es valida.

2. Sabiendo que la eficiencia de búsqueda de una clave candidata implementada por un arbol b, es de orden logaritmico entonces la eficiencia de búsqueda de una clave secundaria implementada por un arbol b es:
   a. Orden lineal
   b. Orden logarítmico
   c. Orden Constante
   d. Es Uno.

3. Si se realiza la baja de un registro en un archivo
   a. Es igual de eficiente hacer una baja lógica que una baja física
   b. Es más eficiente hacer una baja lógica que una baja física
   c. Es menos eficiente hacer una baja lógica que una baja física
   d. Ninguna de las anteriores

4. Un archivo directo con registros de longitud fija
   a. Puede ocupar más espacio que el mismo archivo con registros de longitud variable
   b. Puede ocupar el mismo espacio que el mismo archivo con registros de longitud variable
   c. Ocupa menos espacio que el mismo archivo con registros de longitud variable
   d. Ninguna de las anteriores

5. El acceso promedio para recuperar un dato en un archivo ordenado
   a. Tiene orden lineal
   b. Tiene orden logarítmico
   c. Tiene orden constante
   d. Las opciones a y b son verdaderas
   e. Hay dos opciones pero no la a y la b
   f. Ninguna es correcta.

6. Los archivos con registros de long variable
   a. Ocupan menos espacio que los archivos con registro de longitud fija
   b. Ocupan mas espacio que los archivos con registros de longitud fija
   c. Ocupan el mismo espacio que los archivos con registro de longitud fija
   d. Ninguna de las anterioes es valida.

7. A partir del NRR de un archivo
   a. Se puede acceder a un registro de un archivo fragmentado en solo un acceso
   b. Se puede acceder a un registro de un archivo no fragmentado en un solo acceso
   c. Se puede acceder a un registro de un archivo en un solo acceso
   d. Se puede lograr acceso directo a un archivo
   e. Todas las anteriores
   f. Algunas de las anteriores
   g. Ninguna de las anteriores
   h. Ninguna de las anteriores

8. Un árbol B+ con prefijos simples
   a. Puede desbalancearse
   b. Puede estar balanceado
   c. Está balanceado
   d. Puede no estar balanceado
   e. Todas las anteriores
   f. Algunas de las anteriores
   g. Ninguna de las anteriores
9. Dado un índice de un archivo
   a. La cantidad de acceso para encontrar una clave es entre 1 y 2
   b. La cantidad de acceso es mayor a 2
   c. La cantidad de acceso para encontrar una clave es menor a 4
   d. La cantidad de acceso para encontrar una clave es mayor a 4
   e. Sus registros son de longitud fija
10. Si se implementa un algoritmo que permite generar un arbol b+. Ese algoritmo
   a. Necesita tener acceso a dos nodos en cada momento
   b. Necesita tener acceso a al menos dos nodo en cada momento
   c. Las estructuras internas del arbol pueden implementarse como un arbol B*.
   d. Todas las anteriores
   e. Algunas de las anteriores
   f. Ninguna de las anteriores
11. Si se quiere dispersar un archivo de 40000 elementos
   a. Se requiere un archivo de 40000 cubetas
   b. Se requiere un archivo de al menos 400000 cubetas
   c. Se requiere un archivo de a lo sumo 40000 cubetas
   d. Todas las anteriores
   e. Algunas de las anteriores
   f. Ninguna de las anteriores
12. La densidad de empaquetamiento
   a. Es el principal parámetro de hashing
   b. Es un parámetro de eficiencia para hashing estatico
   c. Evita la existencia de colisiones y por ende de overflow
   d. Ninguna es correcta.

13. El método de tratamiento de desbordes
   a. Afecta la densidad de empaquetamiento
   b. Puede afectar la densidad de empaquetamiento
   c. Afecta la densidad de empaquetamiento solo con algunas variantes de tratamiento de desborde.
   d. Es necesario cuando se aplica hash estático
   e. Todas las anteriores
   f. Algunas de las anteriores
   g. Ninguna de las anteriores

14. Cuando se produce una baja en hash estático, sobre el archivo de datos
   a. Se realiza una operación de escritura y una de lectura
   b. Se realiza al menos una operación de lectura y al menos una de escritura
   c. Se realiza una operación de lectura y al menos una de escritura
   d. Se realiza una operación de escritura y al menos una de lectura

15. Si hubo saturación con hashing
   a. Ocurrió colisión al menos 2 veces
   b. No ocurrió colisión
   c. Se trata con saturación progresiva
   d. Algunas de las anteriores
   e. Ninguna de las Anteriores.

FOD -Examen de trabajos prácticos- Primera fecha 04/06/2024

Escribir claramente en la primera hoja del examen: legajo, apellido y nombre, turno (MM, MT) y temas que rinde (por su número). En cada hoja indicar: legajo/ apellido y nombre y número de hoja/total.

1. Archivos Secuenciales

Considere que se tiene un archivo que contiene información de los préstamos otorgados por una empresa financiera que cuenta con varias sucursales. Por cada préstamo se tiene la siguiente información: número de sucursal donde se otorgó, DNI del empleado que lo otorgó, número de préstamo, fecha de otorgación y monto otorgado. Cada préstamo otorgado por la empresa se considera como una venta. Además, se sabe que **el archivo está ordenado por los siguientes criterios: código de sucursal, DNI del empleado y fecha de otorgación (en ese orden)**.

Se le solicita **definir las estructuras de datos necesarias y escribir el módulo que reciba el archivo de datos y genere un informe en un archivo de texto** con el siguiente formato:

Informe de ventas de la empresa

Sucursal <número sucursal>

Empleado: DNI <DNI empleado>

Año	*Cantidad de ventas*	*Monto de ventas*
……	……	……
……	……	……
Totales	*<Total ventas empleado>*	*<Monto total empleado>*

Empleado: DNI <Dni empleado>

…

Cantidad total de ventas sucursal: ……

Monto total vendido por sucursal: ……

Sucursal <numero sucursal>

….

….

Cantidad de ventas de la empresa: ……

Monto total vendido por la empresa: ……

Notas:

- **El archivo de datos se debe recorrer solo una vez.**
- Para determinar el año de otorgación de un préstamo, puede asumir que existe una función **extraerAño(fecha)** que, a partir de una fecha dada, devuelve el año de la misma.
- En la generación del archivo de texto solo interesa que aparezca la información requerida, NO es necesario que se incluyan los espacios en blanco o tabulaciones que se incluyen en el informe dado como ejemplo.

2. Archivos Directos

Realice el proceso de dispersión mediante el método de hashing extensible, sabiendo que cada registro tiene capacidad para dos claves. El número natural indica el orden de llegada de las mismas. **Deberá explicar los pasos que realiza en cada operación y dibujar los estados sucesivos correspondientes.**

1	JavaScript	10100111	2	C#	10101010
3	HTML	00111110	4	NoSQL	01101111
5	Python	0110101	6	Rust	11110000
7	SQL	01011101	8	Perl	01011011
9	Java	00110100	10	PHP	11100011

FOD -Examen de trabajos prácticos- Primera Fecha - 06/06/23 tema 2

Escribir claramente en la primera hoja del examen: legajo, apellido y nombre, turno (MM, MT) y temas que rinde (por su número). En cada hoja indicar: legajo/ apellido y nombre y número de hoja/total.

1. Archivos Secuenciales

Suponga que tiene un archivo con información referente a los empleados que trabajan en una multinacional. De cada empleado se conoce el dni (único), nombre, apellido, edad, domicilio y fecha de nacimiento.

Se solicita hacer el mantenimiento de este archivo **utilizando la técnica de reutilización de espacio llamada lista invertida**.

Declare las estructuras de datos necesarias e implemente los siguientes módulos:

Agregar empleado: solicita al usuario que ingrese los datos del empleado y lo agrega al archivo sólo si el dni ingresado no existe. Suponga que existe una función llamada existeEmpleado que recibe un dni y un archivo y devuelve verdadero si el dni existe en el archivo o falso en caso contrario. **La función existeEmpleado no debe implementarla**. Si el empleado ya existe, debe informarlo en pantalla.

Quitar empleado: solicita al usuario que ingrese un dni y lo elimina del archivo solo si este dni existe. Debe utilizar la función existeEmpleado. En caso de que el empleado no exista debe informarse en pantalla.

Nota: Los módulos que debe implementar deberán guardar en memoria secundaria todo cambio que se produzca en el archivo.

2 - Árboles

Dado un árbol B de orden 4 y con política derecha para la resolución de underflow, para cada operación dada debe:

a. Dibujar el árbol resultante.
b. Explicar las decisiones tomadas.
c. Indicar las lecturas y escrituras en el orden de ocurrencia.

Las operaciones a realizar son: +260, -310, -490, -60

Árbol inicial:

2: 0 (100) 1 (310) 4 (600) 3

0: (60) 1: (160)(210)(240) 4: (420)(490) 3: (800)

3 - Hashing

Dado el archivo dispersado a continuación, grafique los estados sucesivos para las siguientes operaciones: +45, +89, -70, -34. Indique las lecturas y escrituras en cada

operación, y calcule la densidad de empaquetamiento después de la última operación.

Técnica de resolución de colisiones: **Saturación progresiva.**

f(x) = x MOD 11

Dirección	Clave	Clave
0	55	
1	23	34
2	46	
3		
4	70	
5	60	
6	50	
7	84	
8		
9	42	
10	21	65

FOD -Examen de trabajos prácticos- Primera Fecha - 05/06/2018

Escribir claramente en la primera hoja del examen: legajo, apellido y nombre, turno (LM, MM, MT) y temas que rinde (por su número). En cada hoja indicar: número de hoja/total.

1. Archivos Secuenciales

La empresa de software 'X' posee un servidor web donde se encuentra alojado el sitio de la organización. En dicho servidor, se almacenan en un archivo todos los accesos que se realizan al sitio.
La información que se almacena en el archivo es la siguiente: año, mes, dia, idUsuario y tiempo de acceso al sitio de la organización. El archivo se encuentra ordenado por los siguientes criterios: año, mes, dia e idUsuario.
Se debe realizar un procedimiento que genere un informe en pantalla, para ello se indicará el año calendario sobre el cual debe realizar el informe. El mismo debe respetar el formato mostrado a continuación:

```
Año : —
    Mes:-- 1
        día:-- 1
                idUsuario 1   Tiempo Total de acceso en el dia 1 mes 1
                -------
                idusuario N   Tiempo total de acceso en el dia 1 mes 1
        Tiempo total acceso dia 1 mes 1
        ----------
        día N
                idUsuario 1   Tiempo Total de acceso en el dia N mes 1
                -------
                idusuario N   Tiempo total de acceso en el dia N mes 1
        Tiempo total acceso dia N mes 1
    Total tiempo de acceso mes 1
    ----
    Mes 12
        día 1
                idUsuario 1   Tiempo Total de acceso en el dia 1 mes 12
                -------
                idusuario N   Tiempo total de acceso en el dia 1 mes 12
        Tiempo total acceso dia 1 mes 12
        ----------
        día N
                idUsuario 1   Tiempo Total de acceso en el dia N mes 12
                -------
                idusuario N   Tiempo total de acceso en el dia N mes 12
        Tiempo total acceso dia N mes 12
    Total tiempo de acceso mes 12
Total tiempo de acceso año
```

Se deberá tener en cuenta las siguientes aclaraciones:

- El año sobre el cual realizará el informe de accesos debe leerse desde teclado.
- El año puede no existir en el archivo, en tal caso, debe informarse en pantalla "año no encontrado".
- Debe definir las estructuras de datos necesarias.
- El recorrido del archivo debe realizarse una única vez procesando sólo la información necesaria.

2. Árboles en Archivos

Dado el siguiente árbol B+ de orden 4 y con política de resolución de underflows a derecha. Realice las operaciones detalladas a continuación. Para ello deberá dibujar los estados sucesivos del árbol a partir del resultado de cada operación e indicar nodos leídos y escritos en el orden de ocurrencia. Además, debe describir lo que sucede en cada operación:

Operaciones: +100, -540 , - 67 , -422, -445, -556.

2: 0 (241) 1 (422) 3 (540) 4

0: (3)(35)(67) 1 1: (241)(300)(329) 3 3: (422)(445) 4 4: (556)(600) -1

3. Archivos Directos

Dado el archivo dispersado a continuación, grafique los estados sucesivos para las siguientes operaciones: +12, +45, +89, +59, +26 , -45
Técnica de resolución de colisiones: **Saturación progresiva encadenada**.

NOTA: Indicar Lecturas y Escrituras

f(x) = x MOD 11

Dirección	Enlace	Clave
0	-1	
1	-1	78
2	-1	
3	-1	47
4	-1	
5	-1	27
6	-1	
7	-1	51
8	-1	
9	-1	53
10	-1	

FRANJAMORADA FACULTAD DE INFORMÁTICA · UNIVERSIDAD NACIONAL DE LA PLATA

FOD -Examen de trabajos prácticos- Primera Fecha - 04/06/2019

Escribir claramente en la primera hoja del examen: legajo, apellido y nombre, turno (LM, MM, MT) y temas que rinde (por su número). En cada hoja indicar: número de hoja/total.

1. Archivos Secuenciales

Una empresa que comercializa fármacos recibe de cada una de sus 30 sucursales un resumen mensual de las ventas y desea analizar la información para la toma de futuras decisiones.
El formato de los archivos que recibe la empresa es: cod_farmaco, nombre, fecha, cantidad_vendida, forma_pago (campo String indicando contado o tarjeta).
Los archivos de ventas están ordenados por: cod_farmaco y fecha.
Cada sucursal puede vender cero, uno o más veces determinado fármaco el mismo día, y la forma de pago podría variar en cada venta. Realizar los siguientes procedimientos:

a) Recibe los 30 archivos de ventas e informa por pantalla el fármaco con mayor cantidad_vendida.

NO b) Recibe los 30 archivos de ventas e informa por pantalla la fecha en la que se produjeron más ventas al contado mostrando fecha y cantidad de pagos contado.

c) Recibe los 30 archivos de ventas y guarda en un archivo de texto un resumen de ventas por fecha y fármaco con el siguiente formato: cod_farmaco, nombre, fecha, cantidad_total_vendida. (el archivo de texto deberá estar organizado de manera tal que al tener que utilizarlo pueda recorrer el archivo realizando la menor cantidad de lecturas posibles). **Nota**: en el archivo de texto por fecha, cada fármaco aparecerá a lo sumo una vez. Además de escribir cada procedimiento deberá declarar las estructuras de datos utilizadas.

2. Árboles en Archivos

Dado el árbol **B+** que se detalla más abajo, con orden 6, es decir, capacidad de 5 claves como máximo. **Muestre los estados sucesivos** al realizar la siguiente secuencia de operaciones: +159, -5 y -190, además **indicar nodos leídos y escritos en el orden de ocurrencia. Política de resolución underflow derecha**

Nodo 2: 5, i, 0(10)1(60)3(115)4(145)5(179)6
Nodo 0: 2, h, (1)(5) -> 1
Nodo 1: 2, h, (34)(44) -> 3
Nodo 3: 2, h, (60)(113) -> 4
Nodo 4: 4, h, (120)(125)(131)(139) -> 5
Nodo 5: 5, h, (145)(153)(158)(160)(177) -> 6
Nodo 6: 2, h, (179)(190) -> -1

3. Archivos Directos

Realice el proceso de dispersión mediante el método de hashing extensible, sabiendo que cada registro tiene capacidad para dos claves. El número natural indica el orden de llegada de las mismas. **Deberá explicar los pasos que realiza en cada operación y dibujar los estados sucesivos correspondiente.**

1	Alfa Romeo	10100111	2	Peugeot	10101010
3	Audi	00111110	4	Nissan	01101111
5	BMW	01101011	6	Toyota	11110000
7	Fiat	01011101	8	Suzuki	01011011
9	Ford	00110100	10	Renault	11100011

franja MORADA INFORMATICA UNLP

NIVELACIÓN PROGRAMACIÓN

Sean los archivos:

"**ESCUELAS**": secuencial, ordenado por código de provincia, distrito escolar y número de escuela ascendente.
Contiene un registro por cada escuela de un distrito de una provincia.

El diseño de registro es:
Código de provincia: 1 carácter
Distrito escolar: entero, 2 dígitos
Código de escuela: entero, 2 dígitos
Cantidad de varones: entero, 3 dígitos
Cantidad de mujeres: entero, 3 dígitos
Cantidad de niños que concurren al comedor: entero, 3 dígitos

"**PROVINCIAS**": secuencial, ordenado por código de provincia ascendente.
Contiene un registro por cada provincia del país.
El diseño del registro es:
Código de provincia: 1 carácter
Nombre de provincia: 20 caracteres
Se pide: Obtener a partir de los mismos el listado cuyo diseño se da a continuación.

****** LISTADO DE ESCUELAS ******
Por cada PROVINCIA:
Encabezado de la PROVINCIA con el Nombre de la provincia

Datos a nivel provincia:
DISTRITO
ESCUELA
TOTAL ALUMNOS
% COMEDOR (1)

Al finalizar cada provincia:
TOTAL DE ESCUELAS (2)
TOTAL ALUMNOS (3)

Al finalizar del listado, datos a nivel país en nueva página:
Total de escuelas país
Total alumnado país
Escuela/s con más alumnado del país (5)

-Aclaraciones:-
(1)-Porcentaje de niños que concurren al comedor.
(2)-Cantidad de escuelas de la provincia.
(3)-Cantidad de alumnos de la provincia.
(4)-Código de distrito y código de escuela con más alumnos de esa provincia.
(5)-Nombre de provincia, código de distrito y código de escuela con más alumnos del país.
(5)-En caso que el máximo alumnado se registre en más de una provincia-distrito-escuela, informar todos los casos encontrados

1. Archivos Secuenciales **Parcial Tercera Fecha 08/08/2023**

Una empresa dedicada a la venta de golosinas posee un archivo que contiene información sobre los productos que tiene a la venta. De cada producto se registran los siguientes datos: código de producto, nombre comercial, precio de venta, stock actual y stock mínimo.

La empresa cuenta con 20 sucursales. Diariamente, se recibe un archivo detalle de cada una de las 20 sucursales de la empresa que indica las ventas diarias efectuadas por cada sucursal. De cada venta se registra código de producto y cantidad vendida. Se debe realizar un procedimiento que actualice el stock en el archivo maestro con la información disponible en los archivos detalles y que además informe en un archivo de texto aquellos productos cuyo monto total vendido en el día supere los $10.000. En el archivo de texto a exportar, por cada producto incluido, se deben informar todos sus datos. Los datos de un producto se deben organizar en el archivo de texto para facilitar el uso eventual del mismo como un archivo de carga.

El objetivo del ejercicio es escribir el procedimiento solicitado, junto con las estructuras de datos y módulos usados en el mismo.

Notas:

- Todos los archivos se encuentran ordenados por código de producto.
- En un archivo detalles pueden haber 0, 1 o N registros de un producto determinado.
- Cada archivo detalle solo contiene productos que seguro existen en el archivo maestro.
- Los archivos se recorren una sola vez. En el mismo recorrido, se debe realizar la actualización del archivo maestro con los archivos detalles, así como la generación del archivo de texto solicitado

2 - Arboles

Dado el siguiente árbol B de orden 4 y con política de resolución de underflows derecha o izquierda, realice las siguientes operaciones indicando lecturas y escrituras en el orden de ocurrencia. Además, debe describir detalladamente lo que sucede en cada operación. +410, -200, -500, -100

2: 0 (100) 4 (300) 1 (600) 3

0: (10) 4: (200) 1: (400) (450) (500) 3: (700)

3 - Hashing

Dado el archivo dispersado a continuacion, grafique los estados sucesivos para las siguientes operaciones: +47, +63, +23, -23, -12. NOTA: indicar lecturas y Escrituras necesarias para cada operacion.

Tecnicas de resolucion de colisiones: **Dispersion Doble**

f1 = x MOD 11 **f2 = x MOD 7+1**

Dirección	Clave
0	
1	12
2	
3	36
4	37
5	
6	72
7	
8	41
9	
10	

1. Archivos Secuenciales **Examen de trabajos prácticos - 04/06/2024**

Considere que se tiene un archivo que contiene información de los préstamos otorgados por una empresa financiera que cuenta con varias sucursales. Por cada préstamo se tiene la siguiente información: número de sucursal donde se otorgó, DNI del empleado que lo otorgó, número de préstamo, fecha de otorgación y monto otorgado. Cada préstamo otorgado por la empresa se considera como una venta. Además, se sabe que el archivo está ordenado por los siguientes criterios: código de sucursal, DNI del empleado y fecha de otorgación (en ese orden).

Se le solicita definir las estructuras de datos necesarias y escribir el módulo que reciba el archivo de datos y genere un informe en un archivo de texto con el siguiente formato.

```
Informe de ventas de la empresa
   Sucursal <número sucursal>
      Empleado: DNI <DNI empleado>
         Año Cantidad de ventas Monto de ventas
         ............. ..................... ....................
      Totales <Total ventas empleado> <Monto total empleado>
      Empleado: DNI <DNI empleado>
      ……..
   Cantidad total de ventas sucursal: ……..
   Monto total vendido por sucursal:  ……..
   Sucursal <número sucursal>
   ...……..
Cantidad de ventas de la empresa:  .............
Monto total vendido por la empresa: .............
```

Notas:
*** El archivo de datos se debe recorrer solo una vez.**
*** Para determinar el año de otorgación de un préstamo, puede asumir que existe una función extraerAño(fecha) que, a partir de una fecha dada, devuelve el año de la misma.**
*** En la generación del archivo de texto solo interesa que aparezca la información requerida, NO es necesario que se incluyan los espacios en blanco o tabulaciones que se incluyen en el informe de modo como ejemplo.**

2. Archivos Directos

Realice el proceso de dispersión mediante el método de hashing extensible, sabiendo que cada registro tiene capacidad para dos claves. El número natural indica el orden de llegada de las mismas. Deberá explicar los pasos que realiza en cada operación y dibujar los estados sucesivos correspondientes.

1: JavaScript	10100111	2: C#	10101010
3: HTML	00111110	4: NoSQL	01101111
5: Python	01101101	6: Rust	11110000
7: SQL	01011001	8: Perl	01010111
9: Java	00110100	10: PHP	11100011

FOD - Examen de trabajos prácticos - Segunda fecha 24/06/2024

1 - Archivos Secuenciales

Se desea automatizar el manejo de información referente a los casos positivos de dengue para la Provincia de Buenos Aires. Para esto se cuenta con un archivo maestro que contiene la siguiente información: código de municipio, nombre municipio y cantidad de casos positivos. Dicho archivo está ordenado por código de municipio.

Todos los meses se reciben 30 archivos que contienen la siguiente información: código de municipio y cantidad de casos positivos. El orden de cada archivo detalle es por código de municipio. En cada archivo puede venir información de cualquier municipio, y municipio puede aparecer cero o más de una vez en cada archivo.

Realice el sistema completo que permita la actualización de la información del archivo maestro a partir de los archivos detalle recalculando la cantidad de casos positivos e informando por pantalla aquellos municipios (código y nombre) donde la cantidad total de casos positivos es mayor a 15.

Nota: cada archivo debe recorrerse una única vez.
Nota 1: Los nombres de los archivos deben pedirse por teclado. Se puede suponer que los nombres ingresados corresponden a archivos existentes.
Nota 2: El informe debe incluir cualquier municipio que cumpla la condición, independientemente de si se actualiza o no.

2 - Árboles

Considere un árbol B de orden 5 que representa un índice primario a un archivo de datos que contiene información de los clientes de un supermercado. Dicho índice proporciona acceso indizado por el número de cliente. A continuación se presenta el estado del árbol B en un momento determinado:

8: 2(150)7

2: 0(120)3 7: 4(210)6(300)1(600)9

0: (30) 3: (130) 4: (160)(170) 6: (220)(230)(240)(250) 1: (400) (500) 9: (700)

Resuelva los siguientes incisos:

a) Definir en Pascal los tipos necesarios para definir la estructura de un árbol B (índice) con las características indicadas.

b) Realizar el alta de la clave 280 sobre el árbol B dado. Graficar el árbol B resultante. Indicar los nodos leídos y escritos. Explicar las decisiones tomadas para resolver la operación.

c) Realizar la baja de la clave 120 sobre el árbol B resultante del punto b, considerando política derecha para la resolución de underflows. Graficar el árbol B resultante. Indicar los nodos leídos y escritos. Explicar las decisiones tomadas para resolver la operación.

AComAC F.O.D Examen Tema 1

Archivos

Se cuenta con un archivo que almacena información sobre los tipos de dinosaurios que habitaron durante la era mesozoica, de cada tipo se almacena: código, tipo de dinosaurio, altura y peso promedio, descripción y zona geográfica. El archivo no está ordenado por ningún criterio. Realice un programa que elimine tipos de dinosaurios que estuvieron en el periodo jurásico de la era mesozoica. Para ello se recibe por teclado los códigos de los tipos a eliminar.

Las bajas se realizan apilando registros borrados y las altas reutilizando registros borrados. El registro 0 se usa como cabecera de la pila de registros borrados: el número 0 en el campo código implica que no hay registros borrados y -N indica que el próximo registro a reutilizar es el N, siendo éste un número relativo de registro válido.

Dada la estructura planteada en el ejercicio, implemente los siguientes módulos:

{Abre el archivo y agrega un tipo de dinosaurios, recibido como parámetro manteniendo la política descripta anteriormente}

a. procedure agregarDinosaurios (var a: tArchDinos ; registro: recordDinos);

b. Liste el contenido del archivo en un archivo de texto, omitiendo los tipos de dinosaurios eliminados. Modifique lo que considere necesario para obtener el listado.

Arboles

Dado un árbol B de orden 5 y con política izquierda, para cada operación dada:

a. Dibuje el árbol resultante

b. Explique las decisiones tomadas

c. Escriba las lecturas y escrituras

Operaciones: -20, +2400, -200, -1400, -1600

nodo 2: 4 i 0(200)3(600)1(900)5(1200)4
nodo 0: 1 h(20)
nodo 3: 2 h(400)(500)
nodo 1: 2 h(700)(800)
nodo 5: 2 h(1000)(1100)
nodo 4: 4 h(1400)(1600)(1800)(2000)

Hashing

Dado el archivo dispersado a continuación, grafique los estados sucesivos para las siguientes operaciones: +56, +24, -59, -45. **Justificar brevemente e indicar lecturas y escrituras en todas las operaciones, al finalizar calcular la densidad de empaquetamiento.**

NOTA:

Dirección	Clave	Clave
0	22	
1	34	45
2	13	
3		
4	59	
5	49	
6	28	
7		
8		
9	42	
10	98	43

FOD -Examen de trabajos prácticos- Primera fecha 04/06/2024

Escribir claramente en la primera hoja del examen: legajo, apellido y nombre, turno (MM, MT) y temas que rinde (por su número). En cada hoja indicar: legajo/ apellido y nombre y número de hoja/total.

1. Archivos Secuenciales

Considere que se tiene un archivo que contiene información de los préstamos otorgados por una empresa financiera que cuenta con varias sucursales. Por cada préstamo se tiene la siguiente información: número de sucursal donde se otorgó, DNI del empleado que lo otorgó, número de préstamo, fecha de otorgación y monto otorgado. Cada préstamo otorgado por la empresa se considera como una venta. Además, se sabe que **el archivo está ordenado por los siguientes criterios: código de sucursal, DNI del empleado y fecha de otorgación (en ese orden)**.

Se le solicita **definir las estructuras de datos necesarias y escribir el módulo que reciba el archivo de datos y genere un informe en un archivo de texto** con el siguiente formato:

Informe de ventas de la empresa

Sucursal <número sucursal>

Empleado: DNI <DNI empleado>

Año	*Cantidad de ventas*	*Monto de ventas*
……	……	……
……	……	……
Totales	*<Total ventas empleado>*	*<Monto total empleado>*

Empleado: DNI <Dni empleado>

...

Cantidad total de ventas sucursal: ……

Monto total vendido por sucursal: ……

Sucursal <numero sucursal>

....

....

Cantidad de ventas de la empresa: ……

Monto total vendido por la empresa: ……

Notas:

- **El archivo de datos se debe recorrer solo una vez.**
- Para determinar el año de otorgación de un préstamo, puede asumir que existe una función **extraerAño(fecha)** que, a partir de una fecha dada, devuelve el año de la misma.
- En la generación del archivo de texto solo interesa que aparezca la información requerida, NO es necesario que se incluyan los espacios en blanco o tabulaciones que se incluyen en el informe dado como ejemplo.

2. Archivos Directos

Realice el proceso de dispersión mediante el método de hashing extensible, sabiendo que cada registro tiene capacidad para dos claves. El número natural indica el orden de llegada de las mismas. **Deberá explicar los pasos que realiza en cada operación y dibujar los estados sucesivos correspondientes.**

1	JavaScript	10100111	2	C#	10101010
3	HTML	00111110	4	NoSQL	01101111
5	Python	0110101	6	Rust	11110000
7	SQL	01011101	8	Perl	01011011
9	Java	00110100	10	PHP	11100011

FOD -Examen de trabajos prácticos- Primera Fecha - 06/06/23 tema 2

Escribir claramente en la primera hoja del examen: legajo, apellido y nombre, turno (MM, MT) y temas que rinde (por su número). En cada hoja indicar: legajo/ apellido y nombre y número de hoja/total.

1. Archivos Secuenciales

Suponga que tiene un archivo con información referente a los empleados que trabajan en una multinacional. De cada empleado se conoce el dni (único), nombre, apellido, edad, domicilio y fecha de nacimiento.

Se solicita hacer el mantenimiento de este archivo **utilizando la técnica de reutilización de espacio llamada lista invertida**.

Declare las estructuras de datos necesarias e implemente los siguientes módulos:

Agregar empleado: solicita al usuario que ingrese los datos del empleado y lo agrega al archivo sólo si el dni ingresado no existe. Suponga que existe una función llamada existeEmpleado que recibe un dni y un archivo y devuelve verdadero si el dni existe en el archivo o falso en caso contrario. **La función existeEmpleado no debe implementarla**. Si el empleado ya existe, debe informarlo en pantalla.

Quitar empleado: solicita al usuario que ingrese un dni y lo elimina del archivo solo si este dni existe. Debe utilizar la función existeEmpleado. En caso de que el empleado no exista debe informarse en pantalla.

Nota: Los módulos que debe implementar deberán guardar en memoria secundaria todo cambio que se produzca en el archivo.

2 - Árboles

Dado un árbol B de orden 4 y con política derecha para la resolución de underflow, para cada operación dada debe:

a. Dibujar el árbol resultante.
b. Explicar las decisiones tomadas.
c. Indicar las lecturas y escrituras en el orden de ocurrencia.

Las operaciones a realizar son: +260, -310, -490, -60

Árbol inicial:

2: 0 (100) 1 (310) 4 (600) 3

0: (60) 1: (160)(210)(240) 4: (420)(490) 3: (800)

3 - Hashing

Dado el archivo dispersado a continuación, grafique los estados sucesivos para las siguientes operaciones: +45, +89, -70, -34. Indique las lecturas y escrituras en cada

operación, y calcule la densidad de empaquetamiento después de la última operación.

Técnica de resolución de colisiones: **Saturación progresiva.**

f(x) = x MOD 11

Dirección	Clave	Clave
0	55	
1	23	34
2	46	
3		
4	70	
5	60	
6	50	
7	84	
8		
9	42	
10	21	65

Archivos

Examen Práctica CBD - Primera Fecha - 23/05/2023

Se desea manejar un archivo con información de profesionales de una clínica. Se sabe que el archivo utiliza la técnica de lista invertida para aprovechamiento de espacio. Es decir las bajas se realizan apilando registros borrados y las altas reutilizando registros borrados. El registro en la posición 0 del archivo se usa como cabecera de la pila de registros borrados.

```
Type
profesional= Record
      DNI:integer;
      nombre: String;
      apellido:String;
      sueldo: real;
end;
tArchivo = File of profesional;
```

Nota: El valor 0 en el campo DNI significa que no existen registros borrados, y -N indica que el próximo registro a reutilizar es el N, siendo éste un número relativo de registro válido.

Se solicita implementar los siguientes módulos:

{Crear el Archivo Maestro con un archivo de texto que se recibe como parámetro. Asumir que en el programa principal sólo está hecho el assign de los archivos. Tenga en cuenta la restricción de lectura de los campos en los archivos de texto}

```
Procedure crear(var arch: tArchivo; var info:TEXT);
```

{Abre el archivo y agrega una persona. La persona se recibe como parámetro y debe utilizar la política descripta anteriormente para recuperación de espacio}

```
Procedure agregar (var arch: tArchivo; p: profesional);
```

{Abre el archivo y elimina la persona que tiene el DNI que se recibe como parámetro manteniendo la política descripta anteriormente. La persona puede no existir. Si existe se agrega al archivo de bajas. Tenga en cuenta la restricción de lectura de los campos en los archivos de texto}

```
Procedure eliminar (var arch: tArchivo; DNI: integer; var bajas:TEXT);
```

Árboles

Dado el siguiente árbol B + de orden 5. La raíz se debe mantener en la posición 0 del archivo. Política de Derecha.
Realizar las operaciones : +52, +51, -88, -90. **Indicar el costo de cada operación (Lecturas y Escrituras)**

Nodo 0: 4, i, 1 (23)2 (52) 3 (73) 4 (88) 5
Nodo 1: 4, h, (5) (9) (15) (20) --> 2
Nodo 2: 3, h, (23) (32) (41)(50) --> 3
Nodo 3: 2, h, (53) (62) --> 4
Nodo 4: 1, h, (78) -->5
Nodo 5: 1, h, (90) --> -1

Hashing

Realice el proceso de dispersión mediante el método de hashing extensible, sabiendo que cada registro tiene capacidad para dos claves. El número natural indica el orden de llegada de las claves. Deberá explicar los pasos que realiza en cada operación y dibujar los estados sucesivos correspondientes. **Justifique brevemente.**

1	Java	0100111	2	C	0101010
3	Ruby	1011110	4	Haskell	1001111
5	Kotlin	1010111	6	Python	1110000
7	PHP	1011101	8	SQL	0011011

FOD -Examen de trabajos prácticos- Segunda fecha 24/06/2024

Escribir claramente en la primera hoja del examen: legajo, apellido y nombre, turno (MM, MT) y temas que rinde (por su número). En cada hoja indicar: legajo/ apellido y nombre y número de hoja/total.

1. Archivos Secuenciales

Se desea automatizar el manejo de información referente a los casos positivos de dengue para la Provincia de Buenos Aires. Para esto se cuenta con un archivo maestro que contiene la siguiente información: código de municipio, nombre municipio y cantidad de casos positivos. Dicho archivo está ordenado por código de municipio.

Todos los meses se reciben 30 archivos que contienen la siguiente información: código de municipio y cantidad de casos positivos. El orden de cada archivo detalle es por código de municipio. En cada archivo puede venir información de cualquier municipio, un municipio puede aparecer cero una o más de una vez en cada archivo.

Realice el sistema completo que permita la actualización de la información del archivo maestro a partir de los archivos detalle recalculando la cantidad de casos positivos e informando por pantalla aquellos municipios (código y nombre) donde la cantidad total de casos positivos es mayor a 15.

Nota: cada archivo debe recorrerse una única vez.
Nota 1: Los nombres de los archivos deben pedirse por teclado. Se puede suponer que los nombres ingresados corresponden a archivos existentes.
Nota 2: El informe debe incluir cualquier municipio que cumpla la condición, independientemente de si se actualiza o no.

2. Árboles

Considere un árbol B de orden 5 que representa un índice primario a un archivo de datos que contiene información de los clientes de un supermercado. Dicho índice proporciona acceso indizado por el número de cliente. A continuación se presenta el estado del árbol B en un momento determinado:

8: 2 (150) 7

2: 0 (120) 3 | 7: 4 (210)6(300)1(600)9

0: (30) | 3: (130) | 4: (160)(170) | 6: (220)(230)(240)(250) | 1: (400)(500) | 9:(700)

Resuelva los siguientes incisos:

a) Definir en Pascal los tipos necesarios para definir la estructura de un árbol B (índice) con las características indicadas.

b) Realizar el alta de la clave 280 sobre el árbol B dado. Graficar el árbol B resultante. Indicar los nodos leídos y escritos. Explicar las decisiones tomadas para resolver la operación.

c) Realizar la baja de la clave 120 sobre el árbol resultante del punto b, considerando política derecha para la resolución de underflows. Graficar el árbol B resultante. Indicar los nodos leídos y escritos. Explicar las decisiones tomadas para resolver la operación.

FOD -Examen de trabajos prácticos- Segunda Fecha - 4/07/23 **tema 1**

Escribir claramente en la primera hoja del examen: legajo, apellido y nombre, turno (MM, MT) y temas que rinde (por su número). En cada hoja indicar: legajo/ apellido y nombre y número de hoja/total.

1. Archivos Secuenciales

Suponga que tiene un archivo con información de los partidos de los últimos años de los equipos de primera división del fútbol Argentino. Dicho archivo contiene: código de equipo, nombre de equipo, año, código de torneo, código de equipo rival, goles a favor, goles en contra, puntos obtenidos (0, 1 o 3 dependiendo de si perdió, ganó o empató el partido). El archivo está ordenado por los siguientes criterios: año, código de torneo y código de equipo.
Se le solicita definir las estructuras de datos necesarias y escribir el módulo que reciba el archivo y genere un informe por pantalla con el siguiente formato de ejemplo:

Informe resumen por equipo del fútbol Argentino

Año 1

cod_torneo 1

cod_equipo 1 nombre equipo 1

cantidad total de goles a favor equipo 1

cantidad total de goles en contra equipo 1

diferencia de gol (resta de goles a favor - goles en contra) equipo 1

cantidad de partidos ganados equipo 1

cantidad de partidos perdidos equipo 1

cantidad de partidos empatados equipo 1

cantidad total de puntos en el torneo equipo 1

---

cod_equipo n nombre equipo n

idem anterior para equipo n

El equipo "nombre equipo" fue campeón del torneo codigo de torneo 1 del año 1

---

cod_torneo n

Idem anterior para cada equipo en el torneo n

El equipo "nombre equipo" fue campeón del torneo código de torneo n del año 1

Año n

Idem anterior para cada torneo del año n

Nota: se asume que por torneo hay un único equipo campeón con mayor puntaje.

2 - Árboles

Dado el siguiente árbol B + de orden 4 y con política de resolución de underflows a derecha, realice las siguientes operaciones indicando lecturas y escrituras en el orden de ocurrencia. Además, debe describir detalladamente lo que sucede en cada operación. +58, -403, +260, -550

4: 0 (240) 1 (403) 2 (500) 3

0: (3)(45)(60) 1 1: (240)(255)(360) 2 2: (409)(420) 3 3: (550) -1

3 - Hashing

Dado el archivo dispersado a continuación, **grafique los estados sucesivos** para las siguientes operaciones: **+90, +46, +82, -90**. La técnica de resolución de colisiones a emplear es **saturación progresiva encadenada**. Además, **indique las lecturas y escrituras** necesarias en cada operación. Finalmente, después de todas las operaciones, **indicar la densidad de empaquetamiento**.

La función de dispersión a utilizar es **f(x) = x MOD 11**.

Dir	Enlace	Clave
0	-1	
1	-1	12
2	8	24
3	-1	
4	-1	59
5	-1	
6	-1	17
7	-1	73
8	-1	57
9	-1	
10	-1	

FOD -Examen de trabajos prácticos- Tercera fecha 30/07/2024

Escribir claramente en la primera hoja del examen: legajo, apellido y nombre, turno (MM, MT) y temas que rinde (por su número). En cada hoja indicar: legajo/ apellido y nombre y número de hoja/total.

1. Archivos Secuenciales

Se cuenta con un archivo con información de los diferentes empleados que trabajan en una empresa. De cada empleado se conoce: número, nombre, apellido, dni, fecha de nacimiento y género. El número de empleado no puede repetirse.

Este archivo debe ser mantenido realizando bajas lógicas y utilizando la técnica de reutilización de espacio libre llamada ***lista invertida con registro cabecera*** (si el campo número en el registro cabecera es cero no hay espacio para reutilizar).

Escriba la definición de las estructuras de datos necesarias y los siguientes procedimientos:

ExisteEmpleado: módulo que dado un número de empleado devuelve la posición (NRR) en el archivo donde se encuentra el empleado con el número especificado (en caso de que exista). En caso de que el empleado no exista devuelve el valor cero.

AltaEmpleado: módulo que lee por teclado los datos de un nuevo empleado y lo agrega al archivo, reutilizando espacio disponible en caso de que hubiera. En caso de que el empleado que se desea agregar ya exista en el archivo, se debe informar por pantalla que ya existe el empleado en cuestión (el control de unicidad se debe realizar utilizando el módulo ExisteEmpleado).

BajaEmpleado: módulo que da de baja lógicamente un empleado cuyo número se lee por teclado. Para marcar un empleado como borrado se debe utilizar el campo número para mantener actualizada la lista invertida. Para buscar el empleado a borrar y verificar que exista debe utilizar el módulo ExisteEmpleado. En caso de no existir se debe informar "Empleado no existente".

2. Archivos Directos

Realice el proceso de dispersión mediante el método de hashing extensible, sabiendo que cada registro tiene capacidad para dos claves. El número natural indica el orden de llegada de las mismas.

1	Bit	00111111	**2**	Byte	11110100
3	KiB	10100101	**4**	MiB	01010111
5	GiB	01101011	**6**	TiB	10101010

Para cada operación mostrar el estado del archivo graficando y contestar lo siguiente:

¿Se duplica la tabla de direcciones?. **SI - NO.** Justificar
¿Se produce colisión?. **SI - NO.** Justificar
¿Se produce overflow?. **SI - NO.** Justificar
¿Cuántos bits utiliza para la dispersión?

Escaneado con CamScanner

FOD -Examen de trabajos prácticos- Tercera Fecha - 8/08/23 **tema 1**

Escribir claramente en la primera hoja del examen: legajo, apellido y nombre, turno (MM, MT) y temas que rinde (por su número). En cada hoja indicar: legajo/ apellido y nombre y número de hoja/total.

1. Archivos Secuenciales

Una empresa dedicada a la venta de golosinas posee un archivo que contiene información sobre los productos que tiene a la venta. De cada producto se registran los siguientes datos: código de producto, nombre comercial, precio de venta, stock actual y stock mínimo.

La empresa cuenta con 20 sucursales. Diariamente, se recibe un archivo detalle de cada una de las 20 sucursales de la empresa que indica las ventas diarias efectuadas por cada sucursal. De cada venta se registra código de producto y cantidad vendida. Se debe realizar un procedimiento que actualice el stock en el archivo maestro con la información disponible en los archivos detalles y que además informe en un archivo de texto aquellos productos cuyo monto total vendido en el día supere los $10.000. En el archivo de texto a exportar, por cada producto incluido, se deben informar todos sus datos. Los datos de un producto se deben organizar en el archivo de texto para facilitar el uso eventual del mismo como un archivo de carga.

El objetivo del ejercicio es escribir el procedimiento solicitado, junto con las estructuras de datos y módulos usados en el mismo.

Notas:

- Todos los archivos se encuentran ordenados por código de producto.
- En un archivo detalles pueden haber 0, 1 o N registros de un producto determinado.
- Cada archivo detalle solo contiene productos que seguro existen en el archivo maestro.
- Los archivos se deben recorrer una sola vez. En el mismo recorrido, se debe realizar la actualización del archivo maestro con los archivos detalles, así como la generación del archivo de texto solicitado.

2 - Árboles

Dado el siguiente árbol B de orden 4 y con política de resolución de underflows derecha o izquierda, realice las siguientes operaciones indicando lecturas y escrituras en el orden de ocurrencia. Además, debe describir detalladamente lo que sucede en cada operación. +410, -200, -500, -100

2: 0 (100) 4 (300) 1 (600) 3

0: (10) 4: (200) 1: (400)(450)(500) 3: (700)

3 - Hashing

Dado el archivo dispersado a continuación, grafique los estados sucesivos para las siguientes operaciones: +47, +63, +23, -23, -12. NOTA: Indicar Lecturas y Escrituras necesarias para cada operación.

Escaneado con CamScanner

Técnica de resolución de colisiones: **Dispersión Doble**

AComAC F.O.D Examen Tema 1

Archivos

Se cuenta con un archivo que almacena información sobre los tipos de dinosaurios que habitaron durante la era mesozoica, de cada tipo se almacena: código, tipo de dinosaurio, altura y peso promedio, descripción y zona geográfica. El archivo no está ordenado por ningún criterio. Realice un programa que elimine tipos de dinosaurios que estuvieron en el periodo jurásico de la era mesozoica. Para ello se recibe por teclado los códigos de los tipos a eliminar.

Las bajas se realizan apilando registros borrados y las altas reutilizando registros borrados. El registro 0 se usa como cabecera de la pila de registros borrados: el número 0 en el campo código implica que no hay registros borrados y -N indica que el próximo registro a reutilizar es el N, siendo éste un número relativo de registro válido.

Dada la estructura planteada en el ejercicio, implemente los siguientes módulos:

{Abre el archivo y agrega un tipo de dinosaurios, recibido como parámetro manteniendo la política descripta anteriormente}

a. procedure agregarDinosaurios (var a: tArchDinos ; registro: recordDinos);

b. Liste el contenido del archivo en un archivo de texto, omitiendo los tipos de dinosaurios eliminados. Modifique lo que considere necesario para obtener el listado.

Arboles

Dado un árbol B de orden 5 y con política izquierda, para cada operación dada:

a. Dibuje el árbol resultante

b. Explique las decisiones tomadas

c. Escriba las lecturas y escrituras

Operaciones: -20, +2400, -200, -1400, -1600

nodo 2: 4 i 0(200)3(600)1(900)5(1200)4
nodo 0: 1 h(20)
nodo 3: 2 h(400)(500)
nodo 1: 2 h(700)(800)
nodo 5: 2 h(1000)(1100)
nodo 4: 4 h(1400)(1600)(1800)(2000)

Hashing

Dado el archivo dispersado a continuación, grafique los estados sucesivos para las siguientes operaciones: +56, +24, -59, -45. **Justificar brevemente e indicar lecturas y escrituras en todas las operaciones, al finalizar calcular la densidad de empaquetamiento.**

NOTA:

Dirección	Clave	Clave
0	22	
1	34	45
2	13	
3		
4	59	
5	49	
6	28	
7		
8		
9	42	
10	98	43

AComAC F.O.D Examen Tema 2

Archivos

Se cuenta con un archivo que almacena información sobre los tipos de dinosaurios que habitaron durante la era mesozoica, de cada tipo se almacena: código, tipo de dinosaurio, altura y peso promedio, descripción y zona geográfica. El archivo no está ordenado por ningún criterio. Realice un programa que elimine tipos de dinosaurios que estuvieron en el periodo jurásico de la era mesozoica. Para ello se recibe por teclado los códigos de los tipos a eliminar.

Las bajas se realizan apilando registros borrados y las altas reutilizando registros borrados. El registro 0 se usa como cabecera de la pila de registros borrados: el número 0 en el campo código implica que no hay registros borrados y -N indica que el próximo registro a reutilizar es el N, siendo éste un número relativo de registro válido.

Dada la estructura planteada en el ejercicio, implemente los siguientes módulos:

{Abre el archivo y elimina el tipo de dinosaurio recibido como parámetro manteniendo la política descripta anteriormente}

a. *procedure eliminarDinos (var a: tArchDinos ; tipoDinosaurio: String);*

b. Liste en un txt (archivo de texto) el contenido del archivo omitiendo los tipos de dinosaurios eliminados. Modifique lo que considere necesario para obtener el listado.

Nota: Las bajas deben finalizar al recibir el código 100000

Arboles

Dado un árbol B de orden 5 y con política izquierda, para cada operación dada:

a. Dibuje el árbol resultante

b. Explique las decisiones tomadas

c. Escriba las lecturas y escrituras

Operaciones: -30, +3600, -300, -2100, -2400

nodo 2: 4 i 0(300)3(900)1(1350)5(1800)4
nodo 0: 1 h(30)
nodo 3: 2 h(600)(750)
nodo 1: 2 h(1050)(1200)
nodo 5: 2 h(1500)(1650)
nodo 4: 4 h(2100)(2400)(2700)(3000)

Hashing

Dado el archivo dispersado a continuación, grafique los estados sucesivos para las siguientes operaciones: +56, +24, -59, -45. **Justificar brevemente e indicar lecturas y escrituras en todas las operaciones, al finalizar calcular la densidad de empaquetamiento**

NOTA:

Dirección	Clave	Clave
0	22	
1	34	45
2	13	
3		
4	59	
5	49	
6	28	
7		
8		
9	42	
10	98	43

Tema 1 - Dispersión

3. Para las claves siguientes, realice el proceso de dispersión mediante el método de hashing extensible, sabiendo que cada nodo tiene capacidad para dos registros. El número natural indica el orden de llegada de las claves. Se debe mostrar el estado del archivo para cada operación. Justifique brevemente

1	Dalia	00000001	2	Lirio	01100011
3	Magnolia	11110101	4	Fresia	01010110
5	Caléndula	00101000	6	Jacinto	00110100
7	Orquídeas	10110010	8	Margarita	01111110
9	Gladiolo	01111100	10	Rosa	00110000

Tema 3 - Dispersión

3. Para las claves siguientes, realice el proceso de dispersión mediante el método de hashing extensible, sabiendo que cada nodo tiene capacidad para dos registros. El número natural indica el orden de llegada de las claves. Se debe mostrar el estado del archivo para cada operación. Justifique brevemente

1	húmero	00001001	2	esfenoides	01110100
3	rótula	11100010	4	cóccix	01011111
5	vértebras	00110000	6	occipital	01101011
7	tibia	10100110	8	vomer	01101101
9	palatino	01110001	10	esternón	00110111

1)

3. Archivos Directos

Para las claves siguientes, realice el proceso de dispersión mediante el método de hashing extensible, sabiendo que cada nodo tiene capacidad para dos registros. El número natural indica el orden de llegada de las claves.

1. Austria 00000001	6. Dinamarca 00110100
2. Alemania 01100011	7. Finlandia 10110010
3. Suiza 11110101	8. Polonia 01111110
4. Holanda 01010110	9. Italia 01111100
5. Bélgica 00101000	10. Francia 00110000

Nota: Justifique brevemente cómo realiza cada operación.

2)

3. Archivos Directos

Realice el proceso de dispersión mediante el método de hashing extensible, sabiendo que cada registro tiene capacidad para dos claves. El número natural indica el orden de llegada de las mismas.

Se debe explicar cada operación detalladamente y graficar el estado completo del archivo por cada operación.

1	Alfa	00001001	2	Beta	01110100
3	Gamma	11100010	4	Delta	01011111
5	Epsilon	00110000	6	Rho	01101011
7	Pi	10100110	8	Tau	01101101
9	Psi	01110001	10	Omega	00110111

Hoja 1/4

6/06/23

SAMBIDO ADRIAN A. Leg: 4091/1

Tema 1 Resumen Archivos, secuenciales, arboles Turno TARDE

T1	A
T2	D
T3	-

```
Program Parcial;

const
  Valor Alto = 9999;

type
  Producto = Record
              Codigo : Integer;
              Nombre : String;
              Descripcion: String;
              Precio Compra: Real;
              Precio Venta : Real;
              Ubicacion : String;
  end;

Archivo = File of Producto;

Procedure Leer (var maestro : Archivo; var reg : Producto);
Begin
  If (Not EOF(maestro)) Then
      Read (maestro, reg);   { ES UTILIZADA EN EXISTE PRODUCTO }
  Else
      reg.Codigo := ValorAlto;
end;

Procedure LeerDatos (var reg : Producto);
Begin
  Writeln ('Ingrese Codigo'); readln (reg.Codigo);
  Writeln ('Ingrese Nombre'); Readln (reg.nombre);
  Writeln ('Ingrese Descripcion'); Readln (reg.descripcion);
  Writeln ('Ingrese Precio Compra'); readln (reg.Precio Compra);
  Writeln ('Ingrese Precio Venta'); readln (reg.Precio Venta);
  Writeln ('Ingrese ubicacion'); readln (reg.Ubicacion);
end;

Function Existe Producto (var maestro : Archivo; Codigo : Integer);
```

Hoja 2/4

SAMBIDO ADRIAN A. LEG 40911

TEMA 1 RENDI ARCHIVOS SECUENCIALES 6/06/23 TURNO TARDE

ARBOLES

```
PROCEDURE AgregarProducto (VAR maestro : ARCHIVO).
VAR
Aux, Reg : Producto;
Begin

    LeerDatos (Reg).
    IF (ExisteProducto(maestro, reg.codigo)) then
        WriteLn ('El Producto ingresado existe');

    ELSE  Begin
        Reset (maestro).
        Read (maestro, Aux);
        IF (Aux.Codigo <> 0) then        Begin
            Seek (maestro, (Aux.Codigo * -1));
            Read (maestro, Aux);
            Seek (maestro, FilePos(maestro) - 1);
            Write (maestro, Reg);
            Seek (maestro, 0);
            Write (maestro, Aux);
        end;
        ELSE
            Seek (maestro, FileSize);
            Write (maestro, Reg);
        Close (maestro);
end;
```

Es una función que recibe un archivo

HOJA 3/4

RENDI: ARCHIVOS SECUENCIALES Y ÁRBOLES

TEMA 1 | SAMBUCO ADRIAN A | LEG 4091/1
6/06/23 | TURNO: TARDE

```
PROCEDURE quitarProducto (VAR maestro : ARCHIVO);
VAR
POS, Cod : integer;
Reg : PRODUCTO;
Begin

  WriteLn('INGRESE EL CODIGO DE PRODUCTO A ELIMINAR');
  Readln(Cod);
  IF(EXISTEPRODUCTO(maestro, Cod)) then   Begin
    Reset(maestro);
    Read(maestro, reg);
    Pos := Reg.Codigo;
    While (Reg.Codigo <> Cod) do   Begin
      Read(maestro, reg);
    end

    SEEK(maestro, FILEPOS(maestro) - 1);
    Reg.Codigo := POS;
    POS := FILEPOS(maestro) * -1;
    Write(maestro, reg);
    SEEK(maestro, 0);
    Reg.Codigo := POS;
    WRITE(maestro, Reg);
    CLOSE(maestro);
  end;

  Else

    WriteLn('EL PRODUCTO QUE DESEA ELIMINAR NO EXISTE')
end;
```

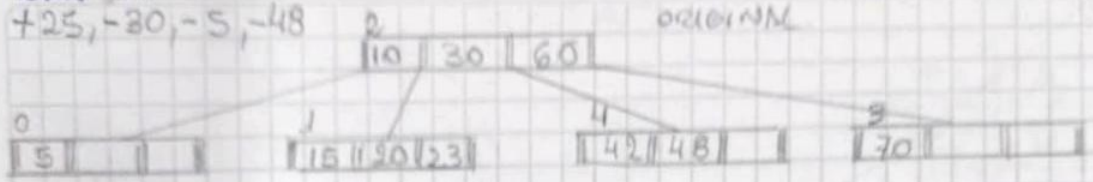
Hoja 4/4

Resol: Archivos secuenciales, árboles 6/06/23

SAMBIDO ADRIAN — Leg 4091/1

Tema 1 — Turno: tarde

+25, -30, -5, -48



ORIGINAL

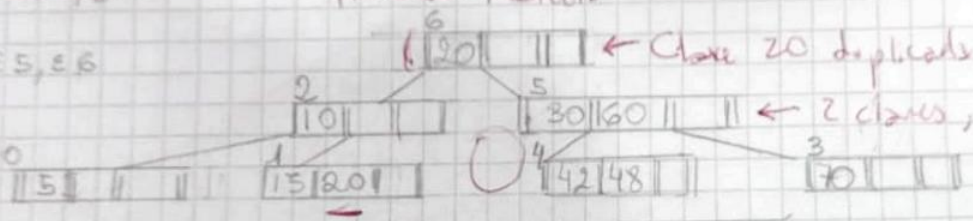
+25 va al nodo "1" y se produce overflow. Se separa

L2, L1 — Se produce overflow en el nodo "2"

15 20 — 23,25 — 10, 20 — 30 60

Falta L/E — Promoción incorrecta

E5, E6

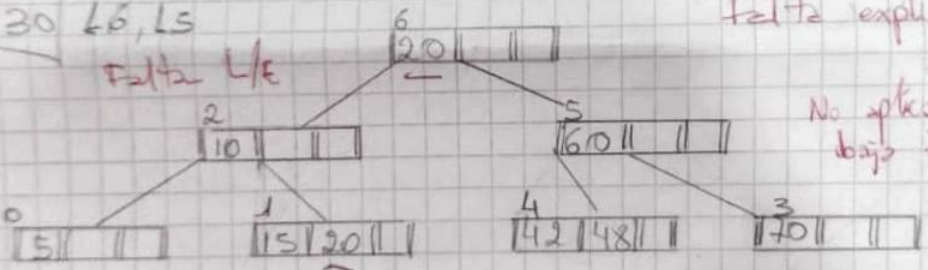


← Clave 20 duplicada

← 2 claves, 2 hijos: No es árbol B

-30 L6, L5 — Falta explicación

Falta L/E

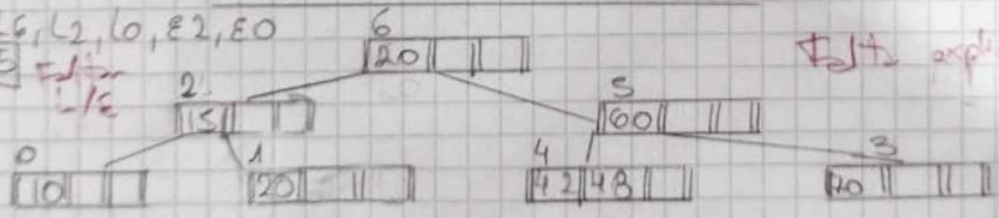


No aplica algoritmo de baja de nodo interno

Arrastra error

L6, L2, L0, E2, E0 — Falta explicación

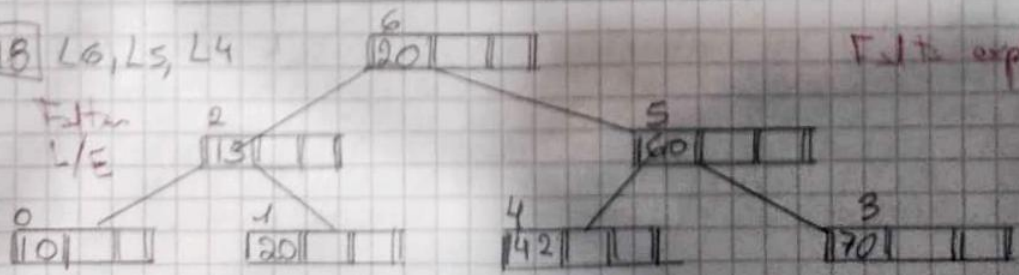
-5 Falta L/E



Arrastra error

-48 L6, L5, L4 — Falta explicación

Falta L/E



Arrastra error

FOD - Primera Fecha - 06/06/2017

Escribir claramente en la primera hoja del examen: legajo, apellido y nombre, turno (LM, MM, MT) y temas que rinde (por su número). En cada hoja indicar: número de hoja/total.

1. Archivos Secuenciales

Suponga que usted es administrador de un servidor de correo electrónico. Los logs del mismo (información guardada acerca de los movimientos que ocurren en el server) se encuentran en la ruta: /var/log/logsmall.dat, donde se guarda la siguiente información: número de usuario, nombre de usuario, nombre, apellido y cantidad de mails enviados.
Diariamente el servidor de correo genera un archivo con la siguiente información: número de usuario, cuenta destino y cuerpo del mensaje. Este archivo representa todos los correos enviados por los usuarios de un día determinado.
Ambos archivos están ordenados por número de usuario, se sabe que un usuario puede enviar cero, uno o mas mails por dia y los usuarios que aparecen en el archivo de envíos diarios seguro existen el logmail.dat.
a- Dado el archivo 6junio2017.dat el cual contiene los movimientos del 6 de junio de 2017, realice el procedimiento necesario para actualizar la información del log.
b- Dado un archivo detalle 6junio2017.dat del punto anterior, realice un procedimiento que genere un archivo de texto llamado listado.txt con la siguiente información:

nroUsuario	nombreUsuario	cantidadMensajesEnviados
…		
nroUsuario	nombreUsuario	cantidadMensajesEnviados
nroUsuario	nombreUsuario	cantidadMensajesEnviados

Debe realizar el programa principal con un menú que muestre las opciones necesarias para llevar a cabo ambos puntos. Debe incluir la declaración de todas las estructuras de datos y variables-necesarias.

Nota: Tener en cuenta que el listado.txt deberán aparecer **<u>todos</u>** los usuarios que existen en el sistema.

2. Árboles en Archivos

Dado el siguiente árbol B de orden 5, mostrar cómo quedaría el mismo luego de realizar las siguientes operaciones: +445, -490, -507, -608. Justificar detalladamente cada operación indicando lecturas y escrituras en orden de ocurrencia. Para la resolución de underflow debe utilizar la **política a izquierda**. Graficar cada operación por separado.

2: 0 (320) 1 (490) 4(555) 5(641) 3
0: (13)(153) 1: (325)(341)(431)(460) 4: (500)(507) 5: (608)(633) 3: (790)(923)

3. Archivos directos

Realice el proceso de dispersión mediante el método de hashing extensible, sabiendo que cada registro tiene capacidad para dos claves. El número natural indica el orden de llegada de las mismas.

1	Berlin	00000001	2	Praga	01100011
3	Estocolmo	11110101	4	Londres	01010110
5	Bucarest	00101000	6	Oslo	00110100
7	Copenhague	10110010	8	Roma	01111110
9	Riga	01111100	10	Varsovia	001100000

FOD -Examen de trabajos prácticos- Primera Fecha - 05/06/2018

Escribir claramente en la primera hoja del examen: legajo, apellido y nombre, turno (LM, MM, MT) y temas que rinde (por su número). En cada hoja indicar: número de hoja/total.

1. Archivos Secuenciales

La empresa de software 'X' posee un servidor web donde se encuentra alojado el sitio de la organización. En dicho servidor, se almacenan en un archivo todos los accesos que se realizan al sitio.
La información que se almacena en el archivo es la siguiente: año, mes, dia, idUsuario y tiempo de acceso al sitio de la organización. El archivo se encuentra ordenado por los siguientes criterios: año, mes, dia e idUsuario.
Se debe realizar un procedimiento que genere un informe en pantalla, para ello se indicará el año calendario sobre el cual debe realizar el informe. El mismo debe respetar el formato mostrado a continuación:

```
Año : —
    Mes:-- 1
        día:-- 1
                idUsuario 1   Tiempo Total de acceso en el dia 1 mes 1
                -------
                idusuario N   Tiempo total de acceso en el dia 1 mes 1
        Tiempo total acceso dia 1 mes 1
        ----------
        día N
                idUsuario 1   Tiempo Total de acceso en el dia N mes 1
                -------
                idusuario N   Tiempo total de acceso en el dia N mes 1
        Tiempo total acceso dia N mes 1
    Total tiempo de acceso mes 1
    ----
    Mes 12
        día 1
                idUsuario 1   Tiempo Total de acceso en el dia 1 mes 12
                -------
                idusuario N   Tiempo total de acceso en el dia 1 mes 12
        Tiempo total acceso dia 1 mes 12
        ----------
        día N
                idUsuario 1   Tiempo Total de acceso en el dia N mes 12
                -------
                idusuario N   Tiempo total de acceso en el dia N mes 12
        Tiempo total acceso dia N mes 12
    Total tiempo de acceso mes 12
Total tiempo de acceso año
```

Se deberá tener en cuenta las siguientes aclaraciones:

- El año sobre el cual realizará el informe de accesos debe leerse desde teclado.
- El año puede no existir en el archivo, en tal caso, debe informarse en pantalla "año no encontrado".
- Debe definir las estructuras de datos necesarias.
- El recorrido del archivo debe realizarse una única vez procesando sólo la información necesaria.

2. Árboles en Archivos

Dado el siguiente árbol B+ de orden 4 y con política de resolución de underflows a derecha. Realice las operaciones detalladas a continuación. Para ello deberá dibujar los estados sucesivos del árbol a partir del resultado de cada operación e indicar nodos leídos y escritos en el orden de ocurrencia. Además, debe describir lo que sucede en cada operación:

Operaciones: +100, -540 , - 67 , -422, -445, -556.

2: 0 (241) 1 (422) 3 (540) 4

0: (3)(35)(67) 1 1: (241)(300)(329) 3 3: (422)(445) 4 4: (556)(600) -1

3. Archivos Directos

Dado el archivo dispersado a continuación, grafique los estados sucesivos para las siguientes operaciones: +12, +45, +89, +59, +26 , -45
Técnica de resolución de colisiones: **Saturación progresiva encadenada**.

NOTA: Indicar Lecturas y Escrituras

f(x) = x MOD 11

Dirección	Enlace	Clave
0	-1	
1	-1	78
2	-1	
3	-1	47
4	-1	
5	-1	27
6	-1	
7	-1	51
8	-1	
9	-1	53
10	-1	

FRANJAMORADA FACULTAD DE INFORMÁTICA · UNIVERSIDAD NACIONAL DE LA PLATA

FOD -Examen de trabajos prácticos- Primera Fecha - 04/06/2019

Escribir claramente en la primera hoja del examen: legajo, apellido y nombre, turno (LM, MM, MT) y temas que rinde (por su número). En cada hoja indicar: número de hoja/total.

1. Archivos Secuenciales

Una empresa que comercializa fármacos recibe de cada una de sus 30 sucursales un resumen mensual de las ventas y desea analizar la información para la toma de futuras decisiones.
El formato de los archivos que recibe la empresa es: cod_farmaco, nombre, fecha, cantidad_vendida, forma_pago (campo String indicando contado o tarjeta).
Los archivos de ventas están ordenados por: cod_farmaco y fecha.
Cada sucursal puede vender cero, uno o más veces determinado fármaco el mismo día, y la forma de pago podría variar en cada venta. Realizar los siguientes procedimientos:

a) Recibe los 30 archivos de ventas e informa por pantalla el fármaco con mayor cantidad_vendida.

NO b) Recibe los 30 archivos de ventas e informa por pantalla la fecha en la que se produjeron más ventas al contado mostrando fecha y cantidad de pagos contado.

c) Recibe los 30 archivos de ventas y guarda en un archivo de texto un resumen de ventas por fecha y fármaco con el siguiente formato: cod_farmaco, nombre, fecha, cantidad_total_vendida. (el archivo de texto deberá estar organizado de manera tal que al tener que utilizarlo pueda recorrer el archivo realizando la menor cantidad de lecturas posibles). **Nota**: en el archivo de texto por fecha, cada fármaco aparecerá a lo sumo una vez. Además de escribir cada procedimiento deberá declarar las estructuras de datos utilizadas.

2. Árboles en Archivos

Dado el árbol **B+** que se detalla más abajo, con orden 6, es decir, capacidad de 5 claves como máximo. **Muestre los estados sucesivos** al realizar la siguiente secuencia de operaciones: +159, -5 y -190, además **indicar nodos leídos y escritos en el orden de ocurrencia. Política de resolución underflow derecha**

Nodo 2: 5, i, 0(10)1(60)3(115)4(145)5(179)6
Nodo 0: 2, h, (1)(5) -> 1
Nodo 1: 2, h, (34)(44) -> 3
Nodo 3: 2, h, (60)(113) -> 4
Nodo 4: 4, h, (120)(125)(131)(139) -> 5
Nodo 5: 5, h, (145)(153)(158)(160)(177) -> 6
Nodo 6: 2, h, (179)(190) -> -1

3. Archivos Directos

Realice el proceso de dispersión mediante el método de hashing extensible, sabiendo que cada registro tiene capacidad para dos claves. El número natural indica el orden de llegada de las mismas. **Deberá explicar los pasos que realiza en cada operación y dibujar los estados sucesivos correspondiente.**

1	Alfa Romeo	10100111	2	Peugeot	10101010
3	Audi	00111110	4	Nissan	01101111
5	BMW	01101011	6	Toyota	11110000
7	Fiat	01011101	8	Suzuki	01011011
9	Ford	00110100	10	Renault	11100011

franja MORADA INFORMATICA UNLP

Parcial primera fecha 7-6-22

1) Archivos secuenciales

Una red de librerías posee varios puntos de ventas y debe procesar un archivo con las ventas realizadas en sus locales. El archivo recibido tiene el siguiente formato: razón social librería, género literario, nombre del libro, precio y cantidad vendida. El archivo se encuentra ordenado por librería, luego por género y por último por nombre del libro. Escriba un programa (programa principal, estructuras y módulos) que dado el archivo descripto, realice un informe por pantalla con el siguiente formato:

Librería: A
- Género: A
  - Nombre de libro: X
  - Total vendido libro X _____
  - ……
  - Nombre de libro N
  - Total vendido libro N _____
- Monto vendido género A _____
- ……..
- Género: N
  - Nombre de libro: X
  - Total vendido libro X _____
  - ……
  - Nombre de libro: N
  - Total vendido libro N _____
- Monto vendido género N _____
- ……..

Monto vendido librería A _____

……

Librería N
- Género: A
  - Nombre de libro: X
  - Total vendido libro X _____
  - ……
  - Nombre de libro N
  - Total vendido libro N _____
- Monto vendido género A _____
- ……..
- Género: N
  - Nombre de libro: X
  - Total vendido libro X _____
  - ……
  - Nombre de libro: N
  - Total vendido libro N _____
- Monto vendido género N _____
- ……..

Monto vendido librería N _____

…..

Monto total librerías _____

2) Árboles en archivos

Dado un árbol B de orden 5 y con política izquierda para cada operación dada

a) Dibuje el árbol resultante
b) Explique las decisiones tomadas
c) Escriba las lecturas y escrituras

Operaciones: -1 +120 -10 -70 -80

nodo 2: 0(10)3(30)1(45)5(60)4

nodo 0: (1)

nodo 3: (20)(25)

nodo 1: (35)(40)
nodo 5: (50)(55)
nodo 4: (70)(80)(90)(100)

3) Archivos directos

Dado el archivo dispersado a continuación, grafique los estados sucesivos para las siguientes operaciones +12 +89 -59 -56
NOTA: indicar lecturas y escrituras en todas las operaciones
Térnica de resolución de colisiones: satuación progresiva f(x)= x MOD 11
Calcular densidad de empaquetamiento

Dirección	Clave	Clave
0	44	
1	34	56
2	57	
3		
4	59	
5	60	
6	28	
7	84	
8		
9	42	
10	54	65

FOD -Examen de trabajos prácticos- Primera Fecha - 06/06/23 **tema 1**

Escribir claramente en la primera hoja del examen: legajo, apellido y nombre, turno (MM, MT) y temas que rinde (por su número). En cada hoja indicar: legajo/ apellido y nombre y número de hoja/total.

1. Archivos Secuenciales

Suponga que tiene un archivo con información referente a los productos que se comercializan en un supermercado. De cada producto se conoce código de producto (único), nombre del producto, descripción, precio de compra, precio de venta y ubicación en depósito.

Se solicita hacer el mantenimiento de este archivo **utilizando la técnica de reutilización de espacio llamada lista invertida**.

Declare las estructuras de datos necesarias e implemente los siguientes módulos:

Agregar producto: recibe el archivo sin abrir y solicita al usuario que ingrese los datos del producto y lo agrega al archivo sólo si el código ingresado no existe. Suponga que existe una función llamada existeProducto que recibe un código de producto y un archivo y devuelve verdadero si el código existe en el archivo o falso en caso contrario. **La función existeProducto no debe implementarla**. Si el producto ya existe debe informarlo en pantalla.

Quitar producto: recibe el archivo sin abrir y solicita al usuario que ingrese un código y lo elimina del archivo solo si este código existe. Puede utilizar la función existeProducto. En caso de que el producto no exista debe informarse en pantalla.

Nota: Los módulos que debe implementar deberán guardar en memoria secundaria todo cambio que se produzca en el archivo.

2 - Árboles

Dado un árbol B de orden 4 y con política izquierda para la resolución de underflow, para cada operación dada debe:

a. Dibujar el árbol resultante.
b. Explicar las decisiones tomadas.
c. Indicar las lecturas y escrituras en el orden de ocurrencia.

Las operaciones a realizar son: +25, -30, -5, -48.

Árbol inicial:

2: 0 (10) 1 (30) 4 (60) 3

0: (5) 1: (15)(20)(23) 4: (42)(48) 3: (70)

3 - Hashing

Dado el archivo dispersado a continuación, grafique los estados sucesivos para las siguientes operaciones: +78, -87, +70, -15. Indique las lecturas y escrituras en cada

operación, y calcule la densidad de empaquetamiento después de la última operación.

Técnica de resolución de colisiones: **Saturación progresiva.**

f(x) = x MOD 11

Dirección	Clave	Clave
0	55	
1	23	12
2	46	
3		
4	70	
5	60	
6	50	
7	84	
8		
9	42	
10	21	65

Redictado FOD 2016 – Primera Fecha de Evaluación Parcial – 15/11/16

1. Dadas las siguientes definiciones, codifique un procedimiento para dar de alta un registro en el archivo. Al realizar el alta se debe comprobar que no se inserten códigos duplicados. Las altas se realizan reutilizando registros borrados, si es que hay. Para reutilizar registros borrados, se mantiene una pila de posiciones borradas con cabecera en el registro cero.

```
Type
    tProducto = Packed Record
        Case activo: Boolean of {discriminante de registros para comprobar si son válidos o
        no (de encadenamiento de posiciones libres)}
        True: ( cod: LongWord;
                nombre: String[23];
                stock: Word;
                precioVta: Single );
        False: ( proxLibre: LongWord ); {el registro es de encadenamiento de espacio
        libre; si proxLibre=0 no hay más posiciones libres en el archivo}
        end;

    tCtlProd = Record
        pp : File of tProducto; // archivo de productos
        p: tProducto; // variable para leer productos
        cab: tProducto // variable para mantener al registro 0 en memoria
        end;

Function Alta ( var a: tCtlProd; const nuevop: tProducto ): Boolean;
{Devuelve False si encuentra código de producto duplicado, o True si se hizo el alta.
El archivo a.pp está abierto para lectura y escritura y no debe cerrarse.
Al abrirse a.pp se leyó el registro 0 en a.cab.}
…
```

2. Dado el estado inicial del siguiente árbol B+, con nodos con capacidad máxima para tres registros en todos los nodos, y mínima 1, especifique los estados sucesivos completos al realizarle las operaciones que se especifican. Además, indique los nodos que se leen y escriben para realizar cada operación.
Nota: Los nodos internos se parten 2-1, y los *underflows* se resuelven con hermanos derechos.

2: 0 (348) 4 (629) 1 (812) 3			
0: 187, 223 | 4	4: 348, 521 | 1	1: 629, 747, 801 | 3	3: 933 | -1

Operaciones: +700, -933, -747.

3. Dado el siguiente estado inicial de un archivo directo estático de bloques con capacidad para dos registros e indicador de desborde, especifique los estados sucesivos completos al realizarle las operaciones que se especifican, e indique los bloques que se leen y escriben para realizar cada una de ellas. Los registros en saturación se organizan con dispersión doble, con función f2(x) = x mod 3 + 1. Para las altas, se debe verificar que no haya claves duplicadas.

0: N |
1: N | 1146, 5431
2: N |
3: N | 2948
4: N | 3659, 814

Operaciones: +644, +8101, +5003.

Resolución

```
Type
    tProducto = Packed Record
        Case activo: Boolean of {discriminante de registros para comprobar si son válidos o
        no (de encadenamiento de posiciones libres)}
        True: ( cod: LongWord;
            nombre: String[23];
            stock: Word;
            precioVta: Single );
        False: ( proxLibre: LongWord ); {el registro es de encadenamiento de espacio
        libre; si proxLibre=0 no hay más posiciones libres en el archivo}
        end;

    tCtlProd = Record
        pp : File of tProducto; // archivo de productos
        p: tProducto; // variable para leer productos
        cab: tProducto // variable para mantener al registro 0 en memoria
        end;

Function Alta ( var a: tCtlProd; const nuevop: tProducto ): Boolean;
{Devuelve False si encuentra código de producto duplicado, o True si se hizo el alta.
El archivo a.pp está abierto para lectura y escritura y no debe cerrarse.
Al abrirse a.pp se leyó el registro 0 en a.cab.}
var l: longWord; // posición libre
begin
// Búsqueda de código duplicado:
seek(a.pp, 1);
repeat
    read(a.pp, a.p)
until (a.p.activo and (a.p.cod = nuevop.cod)) or eof(a.pp);

// Si hay duplicado se sale devolviendo False:
if a.p.activo and (a.p.cod = nuevop.cod) then begin Alta:=False; Exit end;

// Alta normal:
if a.cab.proxLibre <> 0
then begin // hay alguna posición libre:
    l:=a.cab.proxLibre;
    seek(a.pp, l);
    read(a.pp, a.cab);
    seek(a.pp, 0);
    write(a.pp, a.cab)
    end
else l:=FileSize(a.pp); // no hay posición libre
seek(a.pp, l);
write(a.pp, a.nuevop);
Alta:=True
end;
```

Posibles errores:

- Asignar y/o abrir-cerrar el archivo dentro de la función cuando se indica que el archivo ya está abierto y no debe cerrarse: si se abre el archivo con la intención de reposicionarse al comienzo, es un error leve (performance). Si abren y también cierran el error es grave (análisis).

- Posicionamiento en 0 en vez de en 1 para búsqueda de duplicado: error leve (performance). Si no se posicionan en 0 o 1 error muy grave (lógica).
- No buscar duplicado: error grave (análisis).
- Buscar duplicado sin verificar validez de registro: error grave (lógica).
- Encontrar duplicado y seguir leyendo hasta fin de archivo: error moderado (lógica y performance).
- No buscar posición libre y agregar al final: error muy grave (análisis).
- Inconsistencia en el manejo de posiciones libres: muy grave (lógica-algoritmia).

No de admite ningún error muy grave. Se tolera hasta un error grave. Todos los leves son tolerables.

2: 0 (348) 4 (629) 1 (812) 3			
0: 187, 223 | 4	4: 348, 521 | 1	1: 629, 747, 801 | 3	3: 933 | -1

Operaciones: +700, -933, -747.

+700:

7: 2 (747) 6				
2: 0 (348) 4 (629) 1			6: 5 (812) 3	
0: 187, 223 | 4	4: 348, 521 | 1	1: 629, 700 | 5	5: 747, 801 | 3	3: 933 | -1

Costo: [L2]L1 E1E5 E2E6 E7

-933:

7: 2 (747) 6				
2: 0 (348) 4 (629) 1			6: 5 (801) 3	
0: 187, 223 | 4	4: 348, 521 | 1	1: 629, 700 | 5	5: 747 | 3	3: 801 | -1

Costo: [L7]L6L3 L5E5E3 E6

-747:

7: 2 (629) 6			
2: 0 (348) 4		6: 1(747) 5	
0: 187, 223 | 4	4: 348, 521 | 1	1: 629, 700 | 5	5: 801 | -1

Costo: [L7]L6L5 L3E5 L2E2E6 E7
Libres: 3

Posibles errores:

- Al nodo raíz o lo leen en todas las operaciones o no lo leen en ninguna. Cambiar el criterio sería un error leve.
- Cambiar números de nodos: gravísimo.
- No partir nodos internos 2-1 como se especifica sino 1-2: grave.
- No partir las hojas 2-2 sino 3-1 o 1-3: grave.
- Fusionar 5 y 3 liberando al 3 en la baja de 933: moderado.
- Fusionar 2 y 6 liberando al 6 en la baja de 747: grave (si no fusionó en la baja previa no debe hacerlo ahora).
- Liberar al nodo 5 y no al 3 en la baja de 747: si actualiza correctamente el nodo 1 (cambiando referencia al siguiente) con las lecturas y escrituras pertinentes, grave, si no actualiza al 1, gravísimo.
- Errar en lecturas y escrituras en todas las operaciones: gravísimo.

No de admite ningún error muy grave (gravísimo). Se tolera hasta un error grave.

0: N |
1: N | 1146, 5431
2: N |
3: N | 2948
4: N | 3659, 814

Operaciones: +644, +8101, +5003.

+644:

0: N |
1: N | 1146, 5431
2: N | 644
3: N | 2948
4: S | 3659, 814

644 mod 5 = 4; 644 mod 3 +1 = 3, (4 + 3) mod 5 = 2

Costo: L4E4 L2E2

+8101:

0: N |
1: S | 1146, 5431
2: N | 644
3: N | 2948, 8101
4: S | 3659, 814

8101 mod 5 = 1; 8101 mod 3 + 1 = 2, (1 + 2) mod 5 = 3

Costo: L1E1 L3E3

+5003:

0: N |
1: S | 1146, 5431
2: N | 644, 5003
3: S | 2948, 8101
4: S | 3659, 814

5003 mod 5 = 3; 5003 mod 3 + 1 = 3, (3 + 3) mod 5 = 1; (1 + 3) mod 5 = 4; (4 + 3) mod 5 = 2

Costo: L3E3 L1 L4 L2E2

Posibles errores:

- Leer secuencialmente los bloques del archivo hasta llegar al de destino: gravísimo.
- Reflejar cambios correctamente en indicadores de desborde, pero no indicar escrituras correspondientes: grave.
- No usar indicadores de desborde: si no los dibujan y resuelven correctamente marcando registros borrados e indicando bien lecturas y escrituras: grave.
- Errar en lecturas y escrituras en todas las operaciones: gravísimo.

No de admite ningún error muy grave (gravísimo). Se tolera hasta un error grave.

FOD - Segunda Fecha - 27/06/2017
Escribir claramente en la primera hoja del examen: legajo, apellido y nombre, turno (LM, MM, MT) y temas que rinde (por su número). En cada hoja indicar: número de hoja/total.

1. Archivos Secuenciales

a- Defina en **no más de 5 renglones** los conceptos de baja lógica y baja física. Enumere las ventajas y desventajas de la utilización de bajas físicas.

b- Suponga que tiene un archivo que contiene información de los sistemas operativos más utilizados en la actualidad. Cada registro contiene los siguientes campos: nombre_sistema_operativo, cantidad_instalaciones, es_de_codigo_abierto (un sistema operativo puede ser de código abierto o no), tipo_licencia (GPL, BSD, MPL, etc.). Realice los siguientes procedimientos utilizando el registro cabecera para implementar una lista invertida que permita la reasignación de espacio:
b.1 Alta de sistema operativo: recibe un registro con información de un sistema operativo y un archivo. Debe dar el alta el registro en el archivo reutilizando espacio libre en caso de que exista.
b.2 Baja de sistema operativo: recibe un registro con información de un sistema operativo y un archivo. Debe dar de baja el sistema operativo del archivo, en caso de que exista. De lo contrario debe informar que el sistema operativo no existe en el archivo. Tenga en cuenta que deberá dejar preparado el archivo para futuras altas con reasignación de espacio.

Nota: Debe declarar todas las estructuras de datos utilizadas y el campo utilizado para el encadenamiento debe ser cantidad_instalaciones.

2. Árboles en Archivos

a- Suponga que usted es administrador de un servidor de correo electrónico. Los logs del mismo (información guardada acerca de los movimientos que ocurren en el servidor) se encuentran almacenados en el archivo logmail.dat. De los mismos se guarda la siguiente información: número de usuario, nombre de usuario, nombre, apellido y cantidad de mails enviados. Definir la estructura de datos correspondiente a un árbol B de orden M, que almacenará la información correspondiente al archivo logmail.dat. Además, declarar la variable correspondiente al árbol que se va a generar.
b- Dado el siguiente árbol B+ de orden 4 y con política de resolución de underflows a derecha, realice las siguientes operaciones indicando lecturas y escrituras en el orden de ocurrencia. Además, debe describir detalladamente lo que sucede en cada operación. +150, -500.

2: 0 (340) 1 (400) 1 (500) 3
0: (11)(50)(77) 1 1: (340)(350)(360) 1 1: (400)(410)(420) 3 3: (520)(530)-1

3. Archivos Directos

a- Se debe crear y cargar un archivo directo con capacidad para dos registros con dispersión doble para organizar registros en saturación, con los registros cuyas claves se

listan a continuación y de manera que su densidad de empaquetamiento resulte del 70%. Indicar L/E por cada operación.

+1050, +213, +718, +669, +96, +523, +194. $f_2(x) = x \text{ MOD } 3$

b- Realice las bajas de las claves 1050 y 194. Indicar L/E por cada operación

Parcial segunda fecha 29-06-22

1) Archivos secuenciales

Se cuenta con un archivo que contiene información de los diferentes municipios de la Provincia de Buenos Aires. De cada municipio se conoce: nombre de municipio (no puede repetirse), descripción, cantidad de habitantes, extensión en mts2 y año de creación

Este archivo debe ser mantenido realizando bajas lógicas y utilizando la térnica de reutilización de espacio libre llamada **lista invertida con registro cabecera**

Escriba la definición de las estructuras de datos necesarias y los siguientes procedimientos:

ExisteMunicipio: módulo que recibe por parámetro un nombre y devuelve verdadero si el municipio existe en el arhcivo o falso en caso contrario

AltaMunicipio: módulo que lee por teclado los datos de un nuevo municipio y lo agrega al archivo reutilizando espacio disponible en caso de que exista. (El control de unicidad lo debe realizar el módulo anterior). En caso de que el municipio que se quiere agregar ya exista se debe informar "ya existe el municipio en el archivo"

BajaMunicipio: módulo que da de baja lógicamente un municipio cuyo nombre se lee por teclado. Para marcar un municipio como borrado se debe utilizar el campo cantidad de habitantes para mantener actualizada la lista invertida. Para verificar que el municipio a borrar exista debe usar el módulo ExisteMunicipio. En caso de no existir se debe informar "Municipio no existe"

2) Árboles en archivos

Dado un árbol B+ de orden 4 y con política de resolución de underflows a derecha, para cada operación dada:

a. Dibuje el árbol resultante
b. Explique las decisiones tomadas
c. Escriba las lecturas y escrituras en el orden de ocurrencia

Operaciones: +80 +27 -387 -534

4: 0(191)1(387)2(515)3

0:(01)(20)(75)1 1:(191)(230)(300)2 2:(410)(425)(459)3 3:(534)-1

3) Archivos directos

Realice el proceso de dispersión mediante el método de hashing extensible, sabiendo que cada registro tiene capacidad para dos claves. El número natural indica el orden de llegada de las mismas. Deberá explicar los pasos que realiza en cada operación y dibujar los estados sucesivos correspondientes

1	Aconcagua	10100111	2	Kilimanjaro	10101010
3	Mont Blanc	00111110	4	Cervino	01101111
5	Etna	0110101	6	Chañi	11110000
7	Cho Oyu	01011101	8	Vinicunca	01011011
9	Manaslu	00110100	10	Monte Tai	11100011

Redictado FOD 2016 – Tercera Fecha de Evaluación Parcial – 20/12/16

1. Una editorial de diarios y revistas distribuye y vende sus productos (diarios y revistas) al interior del país. La editorial cuenta con 5 distribuidoras. Cada distribuidora envía un listado con los productos vendidos indicando: código de producto y cantidad vendida del mismo.

   La editorial posee un archivo maestro en donde almacena la información de todos los diarios y revistas que distribuye, para ello el archivo maestro cuenta con el código de producto, nombre, descripción y el stock actual de cada producto.

   Escriba el programa principal con la declaración de los tipos de datos necesarios y realice un proceso que reciba los 5 detalles y actualice el archivo maestro con la información proveniente de los archivos de detalle.

   Tanto el maestro como los detalles se encuentran ordenados por el código de producto. No se realizan ventas de ejemplares sin stock.

   Nota: la solución planteada debe ser escalable en cuanto a la cantidad de detalles.

2. Dado el estado inicial del siguiente árbol B+, con nodos con capacidad máxima para tres registros en todos los nodos, y mínima 1. Especifique los estados sucesivos completos al realizarle las operaciones que se detallan a continuación. Además, indique los nodos que se leen y escriben para realizar cada operación, en el orden correspondiente.

   Nota: Los nodos internos se parten 2-1, y los *underflows* se resuelven por defecto con hermanos derechos.

2 0 (348) 4 (747) 1 (812) 3			
0 60, 187, 223 | 4	4 348, 491, 512 | 1	1 747 | 3	3 933 | -1

   Operaciones: +300, -933, +400, -747

3. Dado el siguiente estado inicial de un archivo directo estático de bloques con capacidad para dos registros. Utilizando la función de dispersión h(x) = x mod 7, especifique los estados sucesivos completos al realizarle las operaciones detalladas a continuación, indicando los bloques que se leen y escriben para realizar cada una de ellas en el orden correspondiente, así como la densidad de empaquetamiento de cada estado.

   Los registros en saturación se organizan con dispersión doble, con la función s(x) = x mod 3 + 1.

   Para las altas, se debe verificar que no haya claves duplicadas, y para las bajas, se sustituye la clave del registro a eliminar por -1.

   0: 322, 504
   1: 162
   2: 359, 562
   3: 241, 598
   4: 480
   5: 281, -1
   6: 531, 650

   Operaciones: +510, -562, +301, +442.

Pautas de Corrección

Los errores pueden ser inadmisibles, graves, moderados o leves.

Opciones para aprobar un tema: no tener errores inadmisibles / 1 grave y hasta 2 leves / 4 leves / 2 moderados y hasta 1 leve / 1 moderado y hasta 3 leves. Cualquier otra combinación desaprueba el tema.

Archivos

```
const
    CD = 5; // cantidad de distribuidoras
    NoCod = 999999999;
type
    tProd = record // registro de producto
        cod: longword;
        no, desc: string;
        stock: longword
        end;
    tVta = record // registro de ventas
        cod: longword;
        cant: word
        end;
    tMae = File of tProd; // archivo maestro de productos
    tDet = File of tVta; // archivo detalle de ventas de productos
    tCtlDet = record // registro de control de archivo de detalle
        det: tDet;
        rDet: tVta
        end;
    tDetalles = array[1..CD] of tCtlDet;
procedure leerDet( var det: tDet; var rDet: tVta );
begin
if not eof(det)
then read(det, rDet) else rDet.cod:=NoCod
end;

procedure minimo( var ctlDet: tDetalles; var min: tVta );
var i, imin: byte;
begin
imin:=1;
for i:=2 to CD do if ctlDet[i].rDet.cod < ctlDet[imin].rDet.cod then imin:=i;
min:=ctlDet[imin].rDet;
if min.cod <> NoCod then leerDet(ctlDet[imin].det, ctlDet[imin].rDet)
end;
```

```
procedure actualizar( var ctlDet: tDetalles );
// Recibe los archivos de detalle asignados y sin abrir
var
        mae: tMae; // archivo maestro
        prod: tProd;
        codAct:  longword;
        i: byte;
        min: tVta;
begin
assign(mae, 'productos');
reset(mae);
read(mae, prod);
for i:=1 to CD do begin reset(ctlDet[i].det); leerDet(ctlDet[i].det, ctlDet[i].rDet) end;
minimo(ctlDet, min);
while min.cod <> NoCod do begin
        codAct:=min.cod;
        while prod.cod<>codAct do read(mae, prod); // búsqueda del registro de producto
        while min.cod=codAct do begin
                prod.stock:=prod.stock-min.cant;
                minimo(ctlDet, min)
                end;
        seek(mae, filepos(mae)-1);
        write(mae, prod) // escritura del registro de producto actualizado
        end;
close(mae);
for i:=1 to CD do close(ctlDet[i].det)
end;

var
        nomArch: string;
        detalle: tDetalles;
        i: byte;

begin
for i:=1 to CD do begin
        write('Ingrese nombre de archivo de detalle ', i, ': ');
        readln(nomArch);
        assign(detalle[i].det, nomArch)
        end;
actualizar(detalle)
end.
```

Errores posibles

No asignar y abrir con reset los 5 detalles: inadmisible.

No abrir correctamente (con *reset*) el archivo maestro: grave.

Corte de control principal por eof o valor alto del maestro (procesa registros sin sentido): si la solución es eficaz, moderado, si no, inadmisible.

No realizar un proceso que devuelva mínimo de los detalles: grave.

No inicializar por cada código diferente un contador de revistas: inadmisible.

Volver atrás en detalles: inadmisible.

No buscar en el maestro el registro a actualizar: inadmisible.

No volver atrás en el maestro para escribir: inadmisible.

No actualizar maestro: inadmisible.

No cerrar el maestro: grave.

No procesar en simultáneo detalles y maestro: inadmisible.

Recorrer más de una vez los archivos: inadmisible.

No hacer un procedimiento: leve.

Leer más de una vez al sacar el mínimo (perder registros): inadmisible (solo se avanza en el archivo que tenga el código mínimo).

No utilizar estructura auxiliar para trabajar los 5 detalles y los registros correspondientes a cada detalle: inadmisible, contradice el enunciado, la solución no es escalable.

Árboles

2 0 (348) 4 (747) 1 (812) 3			
0 60, 187, 223 | 4	4 348, 491, 512 | 1	1 747 | 3	3 933 | -1

+300: [L2] L0 E0 E5 E2 E6 E7

7 2 (747) 6				
2 0 (223) 5 (348) 4			6 1 (812) 3	
0 60, 187 | 5	5 223, 300 | 4	4 348, 491, 512 | 1	1 747 | 3	3 933 | -1

-933: [L7] L6 L3 L1 E1 L2 E2 E6 E7

Libres: 3

7 2 (348) 6			
2 0 (223) 5		6 4 (747) 1	
0 60, 187 | 5	5 223, 300 | 4	4 348, 491, 512 | 1	1 747 | -1

+400: [L7] L6 L4 E4 E3 E6

7 2 (348) 6				
2 0 (223) 5		6 4 (491) 3 (747) 1		
0 60, 187 | 5	5 223, 300 | 4	4 348, 400 | 3	3 491, 512 | 1	1 747 | -1

-747: [L7] L6 L3 L1 E3 E1 E6

<table>
<tr><td colspan="5">7
2 (384) 6</td></tr>
<tr><td colspan="2">2
0 (223) 5</td><td colspan="3">6
4 (491) 3 (512) 1</td></tr>
<tr><td>0
60, 187 | 5</td><td>5
223, 300 | 4</td><td>4
348, 400 | 3</td><td>3
491 | 1</td><td>1
512 | -1</td></tr>
</table>

Errores posibles

Renumerar nodos: inadmisible.

Olvidar enlaces en las hojas: inadmisible.

Agregar nodos nuevos en orden incorrecto alta 300: inadmisible (no se entiende el algoritmo). Por el recorrido de los nodos primero se agrega el 5 como hoja, luego el 6 como interno y finalmente el 7 como nueva raíz. Si se aclara la presunción de mantener la raíz en una posición fija, sería aceptable que la raíz permaneciera fija en la posición 2.

Omitir la lectura o escritura de un nodo: grave (se considera un error a cada omisión).

Fusionar en nivel medio baja de 933: moderado.

No reutilizar nodo 3 en el alta 400: grave.

Baja de 747 eliminar el nodo en vez de resolver el *underflow*: si fusionó en nivel medio en la baja de 933, no hay error, si no, inadmisible (no es algorítmico).

Escribir nodos que se liberan: moderado.

Archivos Directos

0: 322, 504
1: 162
2: 359, 562
3: 241, 598
4: 480
5: 281, -1
6: 531, 650

DE=11/14

Las lecturas indicadas en rojo son para validar unicidad de claves.

+510: L6 L0 L1 E1

0: 322, 504
1: 162, 510
2: 359, 562
3: 241, 598
4: 480
5: 281, -1
6: 531, 650

DE=12/14

-562: L2 E2

0: 322, 504
1: 162, 510
2: 359, -1
3: 241, 598
4: 480
5: 281, -1
6: 531, 650

DE=11/14

+301: L0 L2 L4 E2

0: 322, 504
1: 162, 510
2: 359, 301
3: 241, 598
4: 480
5: 281, -1
6: 531, 650

DE=12/14

+442: L1 L3 L5 L0 L2 L4 E5

0: 322, 504
1: 162, 510
2: 359, 301
3: 241, 598
4: 480
5: 281, 442
6: 531, 650

DE=13/14

Errores posibles

Omitir la lectura o escritura de un bloque: grave se cuenta un error por cada omisión).

No verificar que haya claves duplicadas: grave.

Indicar una sola vez la densidad de empaquetamiento: leve.

Hacer mal una sola operación: grave.

No usar la segunda función en caso de colisión: inadmisible.

Alta de 301, si inserta en bloque 4 en vez de bloque 2: grave.

FOD - Tercera Fecha - 11/07/2017

Escribir claramente en la primera hoja del examen: legajo, apellido y nombre, turno (LM, MM, MT) y temas que rinde (por su número). En cada hoja indicar: número de hoja/total.

1. Archivos Secuenciales

En la facultad de Ciencias Jurídicas existe un sistema a través del cual los alumnos del posgrado tienen la posibilidad de pagar las carreras en RapiPago. Cuando el alumno se inscribe a una carrera, se le imprime una chequera con seis códigos de barra para que pague las seis cuotas correspondientes. Existe un archivo que guarda la siguiente información de los alumnos inscriptos: dni_alumno, codigo_carrera y monto_total_pagado.

Todos los días RapiPago manda N archivos con información de los pagos realizados por los alumnos en las N sucursales. Cada sucursal puede registrar cero, uno o más pagos y un alumno puede pagar más de una cuota el mismo día. Los archivos que manda RapiPago tienen la siguiente información: dni_alumno, codigo_carrera, monto_cuota.

a) Se debe realizar un procedimiento que dado el archivo con información de los alumnos inscriptos y los N archivos que envía RapiPago, actualice la información de lo que ha pagado cada alumno hasta el momento en cada carrera inscripto.

b) Realice otro procedimiento que reciba el archivo con información de los alumnos inscriptos y genere un archivo de texto con los alumnos que aún no han pagado nada en las carreras que están inscriptos. El archivo de texto debe contener la siguiente información: dni_alumno, codigo_carrera y la leyenda “alumno moroso”. La organización de la información del archivo de texto debe ser tal de poder utilizarla en una futura importación de datos realizando la cantidad mínima de lecturas.

Precondiciones:

- Cada alumno puede estar inscrito en una o varias carreras.
- Todos los archivos están ordenados, primero por dni_alumno y luego por codigo_carrera.
- En los archivos que envía RapiPago hay información de pagos de alumnos que si o si existen en el archivo de inscriptos.

Nota: Para ambos incisos debe definir todas las estructuras de datos utilizados.

2. Árboles en archivos

a. Se tiene información de los empleados administrados de una agencia de seguros. De los mismos se conocen dni, cuil, apellido y nombre, fecha de nacimiento, estado civil y fecha de ingreso. Los datos de los empleados se encuentran desordenados en un archivo de datos y Ud. debe proveer mecanismos para la búsqueda de empleados por dni mediante un árbol B de orden M. Defina todas las estructuras de datos necesarias para satisfacer estos requerimientos.

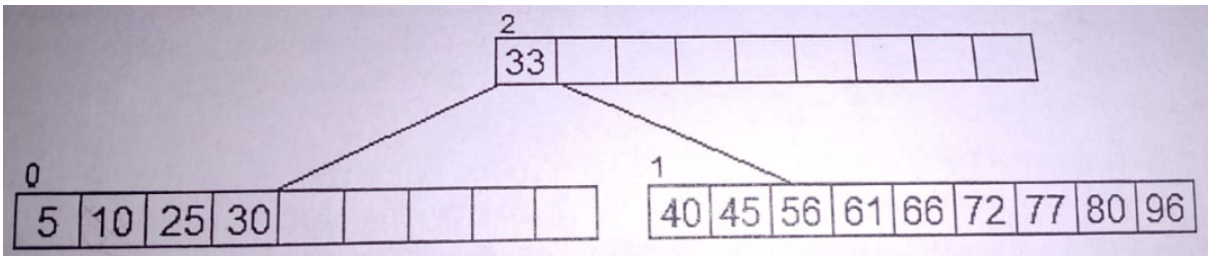
b. Dada la siguiente función:

```
function queHace(A, NRR, clave, NRR_encontrado, pos_encontrada)
begin
        if (A = nulo)
                queHace := false;
        else
                posicionarYLeerNodo(A, nodo, NRR);
```

```
        if (existe(A, nodo, clave, pos)) then begin
                NRR_encontrado := NRR;
                pos_encontrada := pos;
        end
        else
                queHace(A, nodo.hijo[pos], clave, NRR_encontrado, pos_encontrada)
end;
```

b1. Indique que realiza este algoritmo.
b2. Indique para qué tipo/s de árbol/es puede utilizarse. Justifique brevemente.

c. Dado el siguiente árbol B de orden 10 con política de resolución de underflow a izquierda realice la baja de la clave 10, indicando L/E en orden de ocurrencia. Justifique brevemente.



d. Determine los posibles órdenes de un árbol B sabiendo que la cantidad mínima de elementos por nodo es tres (recuerde que la raíz es un no nodo excepcional). Justifique.
e. Determine el orden de un árbol B+ sabiendo que el máximo número de elementos por nodo es 1289.

3. Archivos directos

a. Se debe crear y cargar un archivo directo con capacidad para un registro por cubeta. Realice las operaciones listadas a continuación indicando cada L/E realizada:
+12, +78, +57, +36, +89, +17, +25, +29, -78, -12
Técnica de resolución de colisiones: Saturación progresiva encadenada.

f(x) = x MOD 11.

b. Calcule la densidad de empaquetamiento.

Nota: Debe graficar los estados sucesivos del archivo cuando la operación produzca un desborde.

FOD/IBD 2015 - Primera Fecha del Primer Parcial - 9/6/2015
En la primer hoja del examen escribir claramente legajo, apellido y nombre, turno /mañana: 1 - tarde: 2), temas que rinde (por su número) y cantidad de hojas que entrega. Numerar cada hoja.

1. Archivos Secuenciales
Una cadena de restaurantes posee un archivo de productos que tiene a la venta, de cada producto se registra: código de producto, nombre, descripción, código de barras, categoría de producto, stock actual y stock mínimo. Diariamente el depósito debe efectuar envíos a cada uno de los tres restaurantes que se encuentran en la ciudad de Laprida. Para esto, cada restaurante envía un archivo por mail con los pedidos de productos. Cada pedido contiene: código de producto, cantidad pedida y una breve descripción del producto. Se pide realizar el proceso de actualización del archivo maestro con los tres archivos de detalle, obteniendo un informe de aquellos productos que quedaron por debajo del stock mínimo y sobre estos productos informar la categoría a la que pertenecen. Además , informar aquellos pedidos que no pudieron satisfacerse totalmente por falta de stock, indicando la diferencia que no pudo ser enviada a cada restaurante. Si el stock no es suficiente para satisfacer un pedido en su totalidad, entonces el mismo debe satisfacerse con la cantidad que se disponga.

Nota: Todos los archivos están ordenados por código de producto.

2. Árboles en Archivos
Dado el árbol B+ de orden 4 que se detalla a continuación, indique los estados sucesivos completos al realizar la siguiente secuencia de operaciones: +500, -145, -402. Se debe dejar más carga a la derecha.

2: 0 (238) 3 (547) 1 (729) 4
0: (145) 3 3: (238)(402)(512) 1 1: (614) 4 4: (729) -1

3. Dispersión de Archivos
Dado el archivo dispersado a continuación, indique los estados sucesivos completos para las siguientes operaciones: +13, +23, +42, +75, -31
Método: Saturación progresiva. f(x) = x mod n. n = tamaño tabla
Al finalizar calcule la densidad de empaquetamiento. Indique lecturas y escrituras por cada operación.

Dirección	Registro	Registro
0	66	
1	56	34
2	46	
3		
4	81	

5	16	
6	61	
7	95	
8		
9	31	
10	32	21

Redictado FOD 2016 – Segunda Fecha de Evaluación Parcial – 6/12/16

1. El Gerente General de una cadena de librerías con sucursales en todo el país requiere información sobre ejemplares de libros vendidos en cada sucursal, totalizados por ISBN *(International Standard Book Number)* y por autor (puede haber más de un ISBN de un mismo autor).

   Se dispone de un archivo con registros de ventas compuestos por código de sucursal (codSuc: word), identificador de autor (idAutor: longword), ISBN del libro (isbn: longword) e identificador interno de ejemplar (idEj: word). El archivo está ordenado por codSuc, idAutor e ISBN (puede haber varias ventas del mismo libro en una misma sucursal, pero de distintos ejemplares).

   Definir los tipos de registro (tVta) y del archivo (tArchVtas) y codificar un procedimiento (totalizar) que reciba el archivo asignado y sin abrir y el nombre de un archivo de texto y reporte en el archivo de texto:

   Código de Sucursal: ___

   Identificación de Autor: ___

   ISBN: ___ . Total de ejemplares vendidos del libro: ___

   … (puede haber más ISBN del mismo autor)

   Total de ejemplares vendidos del autor: ___

   … (puede haber más autores)

   Total de ejemplares vendidos en la sucursal: ___

   … (puede haber más sucursales)

   TOTAL GENERAL DE EJEMPLARES VENDIDOS EN LA CADENA: ___

2. Dado el siguiente estado de un Árbol B con nodos de a lo sumo 4 registros, y al menos 2 (excepto el raíz), indicar qué números de nodos se leen o escriben y en qué orden para la siguiente secuencia de operaciones. Además, deberá graficar el estado completo del árbol luego de cada operación: +900, -711, -552.

   Considerar que las bajas en nodos internos se resuelven sustituyendo al registro a eliminar por el siguiente (el menor del subárbol derecho), y que los subflujos *(underflows)* se resuelven por defecto con hermanos derechos (de no haber un hermano derecho, por excepción, se resuelve con el izquierdo).

2: 0 (321) 3 (477) 4 (711) 5 (864) 1				
0: 16, 242, 299	3: 356, 365	4: 552, 628	5: 780, 810, 816	1: 895, 939, 952, 990

3. Dado el siguiente estado de un archivo directo estático con bloques de a lo sumo 3 registros y sin registros en saturación, indicar qué números de bloques se leen o escriben y en qué orden para la siguiente secuencia de operaciones. Se deberá utilizar saturación lineal para resolución de desbordes, empleando como función de dispersión: h(x)=x mod 5, además deberá graficar el estado completo del archivo luego de cada operación: +50, -816, +80.

   Indicar la densidad de empaquetamiento y la cantidad de registros en saturación para cada estado del archivo.

   0: 780, 365, 810

   1: 356, 816, 16

   2: 242, 552

   3: 628

   4: 299

Pautas de Corrección

Los errores pueden ser inadmisibles, graves, moderados o leves.

Para aprobar un tema no se puede tener ningún error inadmisible, a lo sumo un error grave con no más de dos leves, hasta dos moderados (o hasta uno moderado con no más de tres leves) con no más de uno leve si no se tiene ninguno grave, o hasta cuatro leves si no se tiene ninguno moderado ni grave.

Archivo Secuencial

```
type
    tVta = Record
        codSuc: word;
        idAutor, isbn: longword;
        idEj: word
        end;

    tArchVtas = File of tVta;

const
    maxSuc: word = 9999;

procedure leer ( var a: tArchVtas; var v: tVta );
begin
if not eof( a )
    then read( a, v )
    else codSuc:=maxSuc
end;

procedure totalizar ( var a: tArchVtas; const nomArch: string );
var
    reporte: text;
    v: tVta;
    sucActual: word;
    autorActual, isbnActual: longword;
    totISBN, totAutor, totSuc, totGral: longword;
begin
reset( a ); // apertura de archivo de ventas
assign( reporte, nomArch ); rewrite( reporte ); // asignación y apertura de archivo de reportes
totGral:=0;
leer( a, v );
while v.codSuc <> maxSuc do begin
    // Nueva sucursal
    sucActual:=v.codSuc; totSuc:=0; writeln(reporte, 'Codigo de sucursal: ', v.codSuc );
    while v.codSuc = sucActual do begin
        // Nuevo autor
        autorActual:=v.idAutor; totAutor:=0; writeln(reporte, 'Identificacion de autor: ', v.idAutor );
        while (v.codSuc = sucActual) and (v.idAutor = autorActual) do begin
            // Nuevo libro
            isbnActual:=v.isbn; totISBN:=0;
            while (v.codSuc = sucActual) and (v.idAutor = autorActual) and (v.isbn = isbnActual) do begin
                totISBN:=totISBN+1;
```

```
                    leer( a, v )
                    end;
                // Corte de control por cambio de libro
                writeln( reporte, 'ISBN: ', isbnActual, '. Total de ejemplares vendidos: ', totISBN );
                totAutor:=totAutor+totISBN
                end;
            // Corte de control por cambio de autor
            writeln( reporte, 'Total de ejemplares vendidos del autor: ', totAutor );
            totSuc:=totSuc+totAutor
            end;
        // Corte de control por cambio de sucursal
        writeln( reporte, 'Total de ejemplares vendidos en la sucursal: ', totSuc );
        totGral:=totGral+totSuc
        end;
writeln( reporte, 'TOTAL GENERAL DE EJEMPLARES VENDIDOS EN LA CADENA: ', totGral );
close( a ); close( reporte)
end;
```

Errores posibles

No usar la lectura especial para generar un código de sucursal máximo o inexistente cuando se termina el archivo: grave. Si no se usa la lectura especial y no se reportan totales del último isbn, el último autor, la última sucursal y el total genera se reporta inconsistente, el error es inadmisible.

No abrir correctamente (con *reset*) ni cerrar el archivo de ventas: grave.

Reportar en pantalla y no en un archivo de texto: grave.

No abrir correctamente (con *rewrite*) el archivo de texto: grave.

No asignar, abrir ni cerrar el archivo de texto: inadmisible.

Abrir, pero no cerrar archivos: moderado.

Abrir archivo de texto, pero imprimir en pantalla: leve.

No validar igualdad de código de sucursal en ciclos de autor y de libro, o de autor en ciclo de libro: moderado (en cada ciclo que cometa un error).

Usar el procedimiento de lectura especial pero no codificarlo: moderado.

No inicializar un contador o el valor de referencia actual (sucActual, autorActual o isbnActual) o no reportar encabezado en el archivo al iniciar un ciclo: grave.

No reportar un total o hacerlo incorrectamente al producirse un corte de control: grave.

Incrementar todos los totalizadores por unidad en lugar de acumulativamente al haber un corte de control: leve.

Archivo B

2: 0 (321) 3 (477) 4 (711) 5 (864) 1				
0: 16, 242, 299	3: 356, 365	4: 552, 628	5: 780, 810, 816	1: 895, 939, 952, 990

+900: ↓: [L2] L1 - ↑: E1E6 E2E7 E8

8: 2 (711) 7					
2: 0 (321) 3 (477) 4			7: 5 (864) 1 (939) 6		
0: 16, 242, 299	3: 356, 365	4: 552, 628	5: 780, 810, 816	1: 895, 900	6: 952, 990

-711: ↓: [L8] L7 L5 - ↑: E5 E8

8: 2 (780) 7					
2: 0 (321) 3 (477) 4			7: 5 (864) 1 (939) 6		
0: 16, 242, 299	3: 356, 365	4: 552, 628	5: 810, 816	1: 895, 900	6: 952, 990

-552: ↓: [L8] L2 L4 - ↑: L3E3 (Libres: 4) L7E2 (Libres: 4, 7, 8)

2: 0 (321) 3 (780) 5 (864) 1 (939) 6				
0: 16, 242, 299	3: 356, 365, 477, 628	5: 810, 816	1: 895, 900	6: 952, 990

Errores posibles

Renumerar nodos: inadmisible.

Agregar nodos nuevos en orden incorrecto en el alta de 900: inadmisible (no se entiende el algoritmo). Por el recorrido de los nodos primero se agrega el 6 como hoja, luego el 7 como interno y finalmente el 8 como nueva raíz. Si se aclara la presunción de mantener la raíz en una posición fija, sería aceptable que la raíz permaneciera fija en la posición 2.

Omitir la lectura o escritura de un nodo: grave (se considera un error a cada omisión). Por ejemplo, no leer el nodo 7 para llegar al 5 en la baja del 711, o no escribir el 5 o la raíz para persistir su actualización.

Escribir nodos que se liberan: moderado.

No indicar nodos liberados: leve.

Reportar operaciones en orden incorrecto, sin respetar secuencias de descenso hasta hoja y posterior ascenso hasta la raíz, indica desconocimiento del algoritmo: inadmisible.

En la baja de 552, se puede admitir que se liberen los hermanos izquierdos en lugar de los derechos: 3 en lugar del 4, y 2 en lugar del 7; pero ambas liberaciones deben ser consistentes (siempre se debe liberar el izquierdo, o siempre el derecho), de lo contrario se considera error: grave (inconsistencia algorítmica).

Archivo Directo

0: 780, 365, 810
1: 356, 816, 16
2: 242, 552
3: 628
4: 299

Densidad de empaquetamiento (DE): 10/15=2/3.
Registros en saturación: ninguno (enunciado).

+50

Con indicadores de desborde: L0(no hay duplicado) E0L1E1L2E2
0: S|780, 365, 810
1: S|356, 816, 16
2: N|242, 552, 50
3: N|628
4: N|299

Sin indicadores de desborde: L0L1L2(no hay duplicado) E2
0: 780, 365, 810
1: 356, 816, 16
2: 242, 552, 50
3: 628
4: 299

DE: 11/15.
Saturación: 1 registro.

-816

Con indicadores de desborde: L1E1
0: S|780, 365, 810
1: S|356, 16
2: N|242, 552, 50
3: N|628
4: N|299

Sin indicadores de desborde: L1E1
0: 780, 365, 810
1: 356, #, 16
2: 242, 552, 50
3: 628
4: 299

DE: 10/15=2/3.
Saturación: 1 registro.

+80

Con indicadores de desborde: L0L1L2(para comprobar inexistencia de duplicado) E1
0: S|780, 365, 810
1: S|356, 16, 80
2: N|242, 552, 50
3: N|628
4: N|299

Sin indicadores de desborde: L0L1L2L3(para comprobar inexistencia de duplicado) E1
0: 780, 365, 810
1: 356, 80, 16
2: 242, 552, 50
3: 628
4: 299

DE: 11/15.
Saturación: 2 registros.

Errores posibles

Omitir la lectura o escritura de un bloque: grave (se cuenta un error por cada omisión).
Agregar el 80 en un bloque distinto al 1: inadmisible.
No marcar la baja del 816 si no se usan indicadores de desborde: inadmisible.
No reportar registros en saturación o densidad de empaquetamiento: inadmisible si no se reporta ninguno de ambos, grave, si sólo no se reporta registros en saturación. Ídem para reportes incorrectos.
Nota: si no se controla inexistencia de duplicados, sin las lecturas y escrituras correspondientes, la solución es admisible.

Resolución alternativa con optimización de bajas

+50

Con indicadores de desborde: L0(no hay duplicado)
E0L1E1L2E2
0: S|780, 365, 810
1: S|356, 816, 16
2: N|242, 552, 50
3: N|628
4: N|299

Sin indicadores de desborde: L0L1L2(no hay duplicado)
E2
0: 780, 365, 810
1: 356, 816, 16
2: 242, 552, 50
3: 628
4: 299

DE: 11/15.
Saturación: 1 registro.

-816

Con indicadores de desborde: L1L2E2E1
0: S|780, 365, 810
1: S|356, 16, 50
2: N|242, 552
3: N|628
4: N|299

Sin indicadores de desborde: L1L2E2E1
0: 780, 365, 810
1: 356, 50, 16
2: 242, 552, #
3: 628
4: 299

DE: 10/15=2/3.
Saturación: 1 registro.

+80

Con indicadores de desborde: L0L1L2(no hay duplicado)
E2
0: S|780, 365, 810
1: S|356, 16, 50
2: N|242, 552, 80
3: N|628
4: N|299

Sin indicadores de desborde: L0L1L2L3(no hay duplicado)
E2
0: 780, 365, 810
1: 356, 50, 16
2: 242, 552, 80
3: 628
4: 299

DE: 11/15.
Saturación: 2 registros.

FOD/IBD 2016 - Primera Fecha - 07/06/2016

En la primera hoja del examen escribir claramente legajo, apellido y nombre, turno /mañana: 1, tarde 2), temas que rinde (por su número) y cantidad de hojas que entrega. Numerar cada hoja.

1. Archivos secuenciales

Crear un archivo con información de los clientes de un delivery de pizza. Se necesita registrar nombre, apellido, domicilio y teléfono.

Realizar un procedimiento que cargue los clientes ingresandolos desde teclado. La carga finaliza con un número de teléfono igual a 0, el cual no se incorpora al archivo. Los registros se almacenan con longitud variable. Se debe usar el caracter & como separados de registros y $ como separados de campos.

Realizar un procedimiento que, dado un número de teléfono, encuentre la dirección del cliente y la retorne.

Debe definir todas las estructuras y tipos de datos utilizados para la resolución del ejercicio.

Las lecturas y escrituras deben realizarse utilizando blockread y blockwrite respectivamente.

2. Árboles en Archivos

Dado el siguiente árbol B+ de orden 4, mostrar cómo quedaría el mismo luego de realizar las siguientes operaciones: +131, +115, +50, +70, +10, -43, -50, +65, -10, -23, -63.

Por cada operación, justificar brevemente cada operación realizada e indicar las lecturas y escrituras efectuadas. Ante disparidad de carga en el último nivel, el nodo derecho debe quedar más cargado. Ante disparidad de carga en nodos intermedios, el nodo izquierdo debe quedar más cargado. Política de resolución de underflows: izquierda o derecha.

2: 0 (63) 1

0: (23)(43) 1 1: (63) (67) (90)

3. Archivos Directos

Dado el siguiente archivo dispersado a continuación, grafique los estados sucesivos para las siguientes operaciones: +58, +63, +78, +61, +89, -12, -78, -23. Técnica de resolución de colisiones: hashing doble.

Por cada operación indicar las lecturas y escrituras efectuadas.

$f_1(x) = x \text{ MOD } 11$. $f_2(x) = x \text{ MOD } 7 + 1$

Dirección	Clave
0	
1	23
2	

3	47
4	15
5	
6	72
7	
8	30
9	
10	

FOD/IBD 2015 - Primera Fecha del Primer Parcial - 9/6/2015

En la primer hoja del examen escribir claramente legajo, apellido y nombre, turno /mañana: 1 - tarde: 2), temas que rinde (por su número) y cantidad de hojas que entrega. Numerar cada hoja.

1. Archivos Secuenciales

Una cadena de restaurantes posee un archivo de productos que tiene a la venta, de cada producto se registra: código de producto, nombre, descripción, código de barras, categoría de producto, stock actual y stock mínimo. Diariamente el depósito debe efectuar envíos a cada uno de los tres restaurantes que se encuentran en la ciudad de Laprida. Para esto, cada restaurante envía un archivo por mail con los pedidos de productos. Cada pedido contiene: código de producto, cantidad pedida y una breve descripción del producto. Se pide realizar el proceso de actualización del archivo maestro con los tres archivos de detalle, obteniendo un informe de aquellos productos que quedaron por debajo del stock mínimo y sobre estos productos informar la categoría a la que pertenecen. Además , informar aquellos pedidos que no pudieron satisfacerse totalmente por falta de stock, indicando la diferencia que no pudo ser enviada a cada restaurante. Si el stock no es suficiente para satisfacer un pedido en su totalidad, entonces el mismo debe satisfacerse con la cantidad que se disponga.

Nota: Todos los archivos están ordenados por código de producto.

2. Árboles en Archivos

Dado el árbol B+ de orden 4 que se detalla a continuación, indique los estados sucesivos completos al realizar la siguiente secuencia de operaciones: +500, -145, -402. Se debe dejar más carga a la derecha.

2: 0 (238) 3 (547) 1 (729) 4

0: (145) 3 3: (238)(402)(512) 1 1: (614) 4 4: (729) -1

3. Dispersión de Archivos

Dado el archivo dispersado a continuación, indique los estados sucesivos completos para las siguientes operaciones: +13, +23, +42, +75, -31

Método: Saturación progresiva. f(x) = x mod n. n = tamaño tabla

Al finalizar calcule la densidad de empaquetamiento. Indique lecturas y escrituras por cada operación.

Dirección	Registro	Registro
0	66	
1	56	34
2	46	
3		
4	81	

5	16	
6	61	
7	95	
8		
9	31	
10	32	21

2. Dado un árbol B+ de orden 4 y con política izquierda o derecha, para cada operación dada:

a. Dibuje el árbol resultante.

b. Explique brevemente las decisiones tomadas.

c. Escriba las lecturas y escrituras,

Operaciones: +15, +45, -30, -70

nodo 7: 1 i 2(70)6

nodo 2: 2 i 0(30)1(50)3

nodo 6: 1 i 4(90)5

nodo 0: 3 h(5)(10)(20)->1

nodo 1: 3 h(30)(35)(40)->3

nodo 3: 1 h(60)->4

nodo 4: 2 h(70)->5

nodo 5: 2 h(90)->-1

Tema 1 - Árboles

Dado un árbol B de orden 4 y con política izquierda o derecha, para cada operación dada:

a. Dibuje el árbol resultante

b. Explique brevemente las decisiones tomadas

c. Escriba las lecturas y escrituras siguiendo el orden de ejecución

Operaciones: +6, +58, -40, -75

7: i 2(75)6

2: i 0(40)1(60)5

6: i 3(100)4

0: h(4)(13)(23)

1: h(41)(45)(59)

5: h(62) 3: h(80)

4: h(145)

RECUERDE QUE DEBE ADJUNTAR UNA FOTO CON SU RESOLUCIÓN DE PUÑO Y LETRA DENTRO DE LOS 25 MINUTOS DE LA TAREA, LUEGO DE CERRADA LA TAREA NO PODRÁ ADJUNTAR SU SOLUCIÓN.

Dado un árbol B de orden 5 y con política izquierda, para cada operación dada:

a. Dibuje el árbol resultante
b. Explique las decisiones tomadas
c. Escriba las lecturas y escrituras

Operaciones: -76, -400, +900, +12

nodo 2: 3 i 0(76)4(300)1(600)3
nodo 0: 4 h(21)(45)(46)(70)
nodo 4: 2 h(100)(200)
nodo 1: 1 h(400)
nodo 3: 2 h(700)(800)

1)

1. Archivos Secuenciales

Una cadena de restaurantes posee un archivo de productos que tiene a la venta, de cada producto se registra: código de producto, nombre, descripción, código de barras, categoría de producto, stock actual y stock mínimo. Diariamente el depósito debe efectuar envíos a cada uno de los tres restaurantes que se encuentran en la ciudad de Laprida. Para esto, cada restaurante envía un archivo por mail con los pedidos de productos. Cada pedido contiene: código de producto, cantidad pedida y una breve descripción del producto. Se pide realizar el proceso de actualización del archivo maestro con los tres archivos de detalle, obteniendo un informe de aquellos productos que quedaron por debajo del stock mínimo y sobre estos productos informar la categoría a la que pertenecen. Además, informar aquellos pedidos que no pudieron satisfacerse totalmente por falta de stock, indicando la diferencia que no pudo ser enviada a cada restaurante. Si el stock no es suficiente para satisfacer un pedido en su totalidad, entonces el mismo debe satisfacerse con la cantidad que se disponga.

Nota: Todos los archivos están ordenados por código de producto

2)

1. Archivos Secuenciales

Dada la siguiente estructura:

```
Type
   tProducto = record
        código: integer;
        nombre: string[50];
        presentacion: string[100];
   end;
tArchProductos = file of tProducto ;
```

Las bajas se realizan apilando registros borrados y las altas reutilizando registros borrados. El registro 0 se usa como cabecera de la pila de registros borrados: 0 en el registro 0 implica que no hay registros borrados y N en el registro 0 indica que el próximo registro a reutilizar es el N, siendo éste un número relativo de registro válido. El encadenamiento de registros debe realizarse con el campo código de producto pero especificando los números de registro referenciados con signo negativo.

Implemente los siguientes módulos:

```
{Abre el archivo y agrega el producto recibido como parámetro manteniendo la política
descripta anteriormente}
procedure agregar (var a: tArchProductos; producto: tProducto);


{Abre el archivo y elimina el producto recibido como parámetro manteniendo la
política descripta anteriormente}
procedure eliminar (var a: tArchProductos; producto: tProducto);
```

3)

1. Archivos Secuenciales

Un supermercado tiene 25 cajas que registran diariamente las ventas de productos. De cada venta se dispone: número de ticket, código del producto y cantidad de unidades vendidas del producto.
Al finalizar el día, los archivos correspondientes a las cajas se ordenan por código de producto para luego actualizar el archivo de productos. Los registros del archivo de productos contienen el código del producto, la descripción, la cantidad en existencia, el stock mínimo y el precio de venta actual. Implementar un programa que permita:

a) Dada la cantidad de cajas, actualizar la existencia de cada producto registrando la cantidad vendida en la jornada. Tenga en cuenta que el stock no puede quedar por debajo de cero.

b) Informar aquellos productos que dispongan unidades en existencia y no hayan sido vendidos.

c) Informar aquellos productos vendidos que quedaron por debajo del stock mínimo.

d) Informar para cada código de producto, el nombre y el monto total vendido, y también informar el monto total facturado en el día para todos los productos.

NOTA: No debe implementar el ordenamiento de los archivos. Todos los archivos deben recorrerse una única vez

4)

1. El Gerente General de una cadena de librerías con sucursales en todo el país requiere información sobre ejemplares de libros vendidos en cada sucursal, totalizados por ISBN *(International Standard Book Number)* y por autor (puede haber más de un ISBN de un mismo autor).

   Se dispone de un archivo con registros de ventas compuestos por código de sucursal (codSuc: word), identificador de autor (idAutor: longword), ISBN del libro (isbn: longword) e identificador interno de ejemplar (idEj: word). El archivo está ordenado por codSuc, idAutor e ISBN (puede haber varias ventas del mismo libro en una misma sucursal, pero de distintos ejemplares).

   Definir los tipos de registro (tVta) y del archivo (tArchVtas) y codificar un procedimiento (totalizar) que reciba el archivo asignado y sin abrir y el nombre de un archivo de texto y reporte en el archivo de texto:

   Código de Sucursal: ___

   Identificación de Autor: ___

   ISBN: ___ . Total de ejemplares vendidos del libro: ___

   ... (puede haber más ISBN del mismo autor)

   Total de ejemplares vendidos del autor: ___

   ... (puede haber más autores)

   Total de ejemplares vendidos en la sucursal: ___

   ... (puede haber más sucursales)

   TOTAL GENERAL DE EJEMPLARES VENDIDOS EN LA CADENA: ___

5)

1. Archivos Secuenciales

Una cadena de tiendas de indumentaria posee un archivo maestro **no ordenado** con la información correspondiente a las prendas que se encuentran a la venta. De cada prenda se registra: cod_prenda, descripción, colores, tipo_prenda, stock y precio_unitario. Ante un eventual cambio de temporada, se deben actualizar las prendas a la venta. Para ello reciben un archivo conteniendo: cod_prenda de las prendas que quedarán obsoletas. Deberá implementar un procedimiento que reciba ambos archivos y realice la baja lógica de las prendas, para ello deberá modificar el stock de la prenda correspondiente a valor negativo.
Por último, una vez finalizadas las bajas lógicas, deberá efectivizar las mismas compactando el archivo. Para ello no podrá utilizar ninguna estructura auxiliar, debe resolverlo dentro del mismo archivo. Solo deben quedar en el archivo las prendas que no fueron borradas, una vez realizadas todas las bajas físicas.

6)

1. Archivos Secuenciales

Una empresa que comercializa fármacos recibe de cada una de sus 30 sucursales un resumen mensual de las ventas y desea analizar la información para la toma de futuras decisiones.
El formato de los archivos que recibe la empresa es: cod_farmaco, nombre, fecha, cantidad_vendida, forma_pago (campo String indicando contado o tarjeta).
Los archivos de ventas están ordenados por: cod_farmaco y fecha.
Cada sucursal puede vender cero, uno o más veces determinado fármaco el mismo día, y la forma de pago podría variar en cada venta. Realizar los siguientes procedimientos:

a) Recibe los 30 archivos de ventas e informa por pantalla el fármaco con mayor cantidad_vendida.

NO → b) Recibe los 30 archivos de ventas e informa por pantalla la fecha en la que se produjeron más ventas al contado mostrando fecha y cantidad de pagos contado.

c) Recibe los 30 archivos de ventas y guarda en un archivo de texto un resumen de ventas por fecha y fármaco con el siguiente formato: cod_farmaco, nombre, fecha, cantidad_total_vendida. (el archivo de texto deberá estar organizado de manera tal que al tener que utilizarlo pueda recorrer el archivo realizando la menor cantidad de lecturas posibles). **Nota**: en el archivo de texto por fecha, cada fármaco aparecerá a lo sumo una vez. Además de escribir cada procedimiento deberá declarar las estructuras de datos utilizadas.

Parciales

ARCHIVOS

1) Una federación de competidores de running organiza distintas carreras al mes. Cada carrera cuenta con DNI de corredor, apellido, nombre, kms que corrió y si ganó o no la carrera (valor 1 indica que ganó, valor 0 indica que no ganó
puede haber distintas cantidades de carreras en el mes. Para el mes de abril se organizaron 5 carreras
escriba el programa principal con la declaración de tipos necesaria y realiza un proceso que reciba los 5 archivos y genere un archivo maestro con la siguiente información: DNI, apellido, nombre kms totales y carreras ganadas.
todos los archivos se encuentran ordenados por DNI del corredor. Cada persona puede haber recorrido una o más carreras
2) Una cadena de restaurantes posee un archivo de productos que tiene a la venta, de cada producto se registra: código de producto, nombre, descripción, código de barras, categoría de producto, stock actual y stock mínimo. Diariamente el depósito debe efectuar envíos a cada uno de los tres restaurantes que se encuentran en la ciudad de Laprida. Para esto, cada restaurante envía un archivo por mail con los pedidos de productos. Cada pedido contiene: código de producto, cantidad pedida y una breve descripción del producto. Se pide realizar el proceso de actualización del archivo maestro con los tres archivos de detalle, obteniendo un informe de aquellos productos que quedaron por debajo del stock mínimo y sobre estos productos informar la categoría a la que pertenecen. Además informar aquellos pedidos que no pudieron satisfacerse totalmente por falta de stock, indicando la diferencia que no pudo ser enviada a cada restaurante. Si el stock no es suficiente para satisfacer un pedido en su totalidad, entonces el mismo debe satisfacerse con la cantidad que se disponga. Nota: totos los archivos están ordenados por código de producto
3) Crear un archivo con información de los clientes de un delivery de pizza. Se necesita registrar nombre, apellido, domicilio y teléfono. Realizar un procedimiento que cargue los clientes ingresándolos desde teclado. La carga finaliza con un número de teléfono igual a 0, el cual no se incorpora al archivo. Los registros se almacenan con longitud variable. Se debe usar el carácter & como separados de registros y $ como separados de campos.
Realizar un procedimiento que, dado un número de teléfono, encuentre la dirección del cliente y la retorne.
Debe definir todas las estructuras y tipos de datos utilizados para la resolución del ejercicio.
4) Suponga que usted es administrador de un servidor de correo electrónico. Los logs del mismo (información guardada acerca de los movimientos que ocurren en el server) se encuentran en la ruta: /var/log/logsmall.dat, donde se guarda la siguiente información: número de usuario, nombre de usuario, nombre, apellido y cantidad de mails enviados. Diariamente el servidor de correo genera un archivo con la siguiente información: número de usuario, cuenta destino y cuerpo del mensaje. Este archivo representa todos los correos enviados por los usuarios de un día determinado. Ambos archivos están ordenados por número de usuario, se sabe que un usuario puede enviar cero, uno o mas mails por día y los usuarios que aparecen en el archivo de envíos diarios seguro existen el logmail.dat.
   a. Dado el archivo 6junio2017.dat el cual contiene los movimientos del 6 de junio de 2017, realice el procedimiento necesario para actualizar la información del log.
   b. Dado un archivo detalle 6junio2017.dat del punto anterior, realice un procedimiento que genere un archivo de texto llamado listado.txt con la siguiente información:
   nroUsuario nombreUsuario cantidadMensajesEnviados
   ...
   nroUsuario nombreUsuario cantidadMensajesEnviados
   nroUsuario nombreUsuario cantidadMensajesEnviados

   Debe realizar el programa principal con un menú que muestre las opciones necesarias para llevar a cabo ambos puntos. Debe incluir la declaración de todas las estructuras de datos y variables-necesarias.
   Nota: Tener en cuenta que el listado.txt deberán aparecer todos los usuarios que existen en el sistema
5) En la facultad de Ciencias Jurídicas existe un sistema a través del cual los alumnos del posgrado tienen la posibilidad de pagar las carreras en Rapipago. Cuando el alumno se inscribe a una carrera, se le imprime una chequera con seis códigos de barra para que pague las seis cuotas correspondientes. Existe un archivo que guarda la siguiente información de los alumnos inscriptos: dni_alumno, codigo_carrera y monto_total_pagado. Todos los días RapiPago manda N archivos con información de los pagos realizados por los alumnos en las N sucursales. Cada sucursal puede registrar cero, uno o más pagos y un alumno puede pagar más de una cuota el mismo día. Los archivos que manda RapiPago tienen la siguiente información: dni_alumno, codigo_carrera, monto_cuota.
   a. Se debe realizar un procedimiento que dado el archivo con información de los alumnos inscriptos y los N archivos que envía Rapipago, actualice la información de lo que ha pagado cada alumno hasta el momento en cada carrera inscripto.

b. realice otro procedimiento que reciba el archivo con información de los alumnos inscriptos y genere un archivo de texto con los alumnos que aún no han pagado nada en las carreras que están inscriptos. El archivo de texto debe contener la siguiente información: dni_alumno, codigo_carrera y la leyenda "alumno moroso". La organización de la información del archivo de texto debe ser tal de poder utilizarla en una futura importación de datos realizando la cantidad mínima de lecturas.

Precondiciones:

- Cada alumno puede estar inscrito en una o varias carreras.
- Todos los archivos están ordenados, primero por dni_alumno y luego por codigo_carrera.
- En los archivos que envía Rapipago hay información de pagos de alumnos que si o si existen en el archivo de inscriptos.

Nota: Para ambos incisos debe definir todas las estructuras de datos utilizados

6) Se desea modelar un sistema para manejar las ventas realizadas en una cadena de repuestos para autos. La cadena cuenta con 3 sucursales y una casa matriz. Todas las semanas la casa matriz recibe un arhcivo binario de cada sucursal con el detalle de las ventas realizadas. Cada archivo cuenta con la siguiente información: cod_repuesto, nombre_repuesto, cantidad_vendida y fechaYhora_venta. Cada repuesto en los detalles puede no aparecer o aparecer en más de una o varias ventas y todos los archivos están ordenados por cod_repuesto

Escriba un programa (Programa principal, estructuras y módulos) que datos los archivos detalle de cada sucursal, genere un nuevo archivo binario llamado total_repuestos_vendidos.dat con la suma de ventas de cada repuesto en todas las sucursales. El nuevo archivo deberá tener el siguiente formato: cod_repuesto, nombre_repuesto y cantidad_total_vendida

Los archivos deben recorrerse solo una vez

7) A partir de un siniestro ocurrido se perdieron las actas de nacimiento y fallecimientos de toda la provincia de Buenos Aires de los últimos 10 años. En pos de recuperar dicha información, se deberá procesar 2 archivos por cada una de las 50 delegaciones distribuidas en la provincia, un archivo de nacimientos y otro de fallecimientos y crear el archivo maestro reuniendo dicha información

Los archivos detalles con nacimientos, contendrán la siguiente información: nro. de partida de nacimiento, nombre, apellido, dirección detallada (calle, nro., piso, depto., ciudad), matrícula del médico, nombre y apellido de la madre, DNI de la madre, nombre y apellido del padre, DNI del padre

En cambio los 50 archivos de fallecimientos tendrán: nro. partida de nacimiento, DNI, nombre y apellido del fallecido, matrícula del médico que firma el deceso, fecha y hora del deceso y lugar

Realizar un programa que cree el archivo maestro a partir de toda la información los archivos. Se debe almacenar en el maestro: nro. de partida de nacimiento, nombre, apellido, dirección detallada, matrícula del médico, nombre y apellido de la madre, DNI de la madre, nombre y apellido del padre, DNI del padre y si falleció, además matrícula del médico que firma del deceso, fecha y hora del deceso y lugar. Se deberá además listar en un archivo de texto la información recolectada de cada persona

8) Se desea modelar un sistema para manejar los accesos realizados a diferentes procesos por los servidores de la Facultad de Informática. La facultad cuenta con 3 servidores. Todas las semanas cada servidor genera un archivo binario con el detalle de los accesos realizados. Cada archivo cuenta con la siguiente información: cod_proceso, nombre y fechaYHora_acceso. Cada servidor puede no acceder a un proceso o acceder una o varias veces y todos los archivos están ordenados por cod_proceso.

Escriba un programa (programa principal, estructuras y módulos) que datos los archivos detalle de cada servidor, genere un nuevo archivo binario llamado accesos, datos.dat con la suma de accesos a cada proceso en todos los servidores. El nuevo archivo deberá tener el siguiente formato: cod_proceso, nombre y cantidad_total_accesos

9) Se desea modelar un sistema para manejar la cantidad de bebidas consumidas en un festival de folklore realizado en la ciudad de Laprida. El festival cuenta con 3 puestos de bebidas. Al finalizar el festival cada puesto genera un archivo binario de ventas con las siguiente información: cod_bebida, nombre, cantidad_vendida y cod_vendedor. Cada bebida puede no venderse o aparecer en una o varias ventas y todos los archivos están ordenados por cod_bebida

Escriba un programa(programa principal, estructuras y módulos) que dados los archivos detalle de cada puesto, genere un nuevo archivo binario llamado total.dat con la suma de ventas de cada bebida en todos los puestos. El nuevo archivo deberá tener el siguiente formato: cod_bebida, nombre y cantidad_total_vendida

10) La gerencia general de una empresa de ventas de artículos para el hogar con dos sucursales en la ciudad recibe mensualmente de cada sucursal un archivo ordenado por códigos de categoría, de marca y de modelo de productos vendidos en el mes, con la cantidad de unidades vendidas de cada uno. Codificar un procedimiento que reciba como parámetros a los dos archivos de ventas y a un archivo de texto, todos asignados y sin abrir, y reporte en el archivo de texto al total de unidades vendidas por categoría, por categoría y marca, y por categoría, marca y modelo. Una misma marca puede tener productos de distintas categorías y distintas marcas pueden tener productos con el mismo código de modelo

11) Un supermercado tiene 25 cajas que registran diariamente las ventas de productos. De cada venta se dispone:

número de ticket, código del producto y cantidad de unidades vendidas del producto. Al finalizar el día, los archivos correspondientes a las cajas se ordenan por código de producto para luego actualizar el archivo de productos. Los registros del archivo de productos contienen el código del producto, la descripción, la cantidad en existencia, el stock mínimo y el precio de venta actual. Implementar un programa que permita:

a. Dada la cantidad de cajas, actualizar la existencia de cada producto registrando la cantidad vendida en la jornada. Tenga en cuenta que el stock no puede quedar por debajo de cero
b. Informar aquellos productos que dispongan de unidades en existencia y no hayan sido vendidos
c. Informar aquellos productos que dispongan unidades en existencia y no hayan sido vendidos
d. Informar aquellos productos vendidos que quedaron por debajo el stock mínimo
e. informar para cada código de producto, el nombre y el monto total vendido y también informar el monto total facturado en el día de todos los productos

12) Una cadena de tiendas de indumentaria posee un archivo maestro **no ordenado** con la información correspondiente a las prendas que se encuentran a la venta. De cada prenda se registra: cod_prenda, descripción, colores, tipo_prenda, stock y precio_unitario. Ante un eventual cambio de temporada, se deben actualizar las prendas a la venta. Para ello reciben un archivo conteniendo: cod_prenda de las prendas que quedarán obsoletas. Deberá implementar un procedimiento que recibe ambos archivos y realice la baja lógica de las prendas, para ello deberá modificar el stock de la prenda correspondiente a valor negativo.
Por ello no podrá usar ninguna estructura auxiliar, debe resolverlo dentro del mismo archivo. Solo deben quedar en el archivo las prendas que no fueron borradas, una vez realizadas todas las bajas físicas

13) Una empresa que comercializa fármacos recibe de cada una de sus 30 sucursales un resumen mensual de las ventas y desea analizar la información para la toma de futuras decisiones. El formato de los archivos que recibe la empresa es: cod_farmaco, nombre, fecha, cantidad_vendida, forma_pago (campo string). Los archivos de ventas están ordenados por: cod_farmaco y fecha. Cada sucursal puede vender cero, uno o más veces determinado fármaco el mismo día y la forma de pago podría variar en cada venta. Realizar los siguientes procedimientos:

a. Recibe los 30 archivos de ventas e informa por pantalla el fármaco con mayor cantidad_vendida
b. Recibe los 30 archivos de ventas y guarda en un archivo de texto un resumen de ventas por fecha y fármaco con el siguiente formato: cod_farmaco, nombre, fecha, cantidad_total_vendida (el archivo de texto deberá estar organizado de manera que al tener que utilizarlo pueda recorrer el archivo realizando la menor cantidad de lecturas posibles. Nota: en el archivo de texto por fecha, cada fármaco aparecerá a lo sumo una vez. Además de escribir cada procedimiento deberá declarar las estructuras de datos usadas

14) Se cuenta con un archivo que almacena información sobre especies en vía de extinción para ello se almacena: código, nombre de especie, familia de ave, descripción y zona geográfica. El archivo no está ordenado por ningún criterio. Realice un programa que elimine especies de aves, para ello se recibe por teclado las especies a eliminar. Deberá realizar todas las declaraciones necesarias, implementar todos los procedimientos que requiera y una alternativa para borrar los registros. Para ello deberá implementar dos procedimientos, uno que marque los registros a borrar y otro posteriormente que comparte el archivo, quitando los registros marcados. Para quitar los registros se deberá copiar el último registro del archivo en la posición del registro a borrar y luego eliminar del archivo el último registro de forma de evitar registros duplicados. Las bajas deben finalizar el recibir el código 100000

ÁRBOLES

1) dado el siguiente árbol B+ de orden 5. La raíz se debe mantener en la posición 0 del archivo. Política de derecha. Realizar las operaciones: -52 +21 -90 -78. Indicar el costo de cada operación (lecturas y escrituras)
nodo 0: 4,i,1(23)2(52)3(73)4(88)5
nodo 1: 4,h,(5)(9)(15)(20) --> 2
nodo 2: 3,h, (23)(32)(41)-->3
nodo 3:2,h,(53)(62)-->4
nodo 4:1,h,(78)-->5
nodo 5:1,h,(90)-->1

2) Dado el árbol B+ de orden 4 que se detalla a continuación, indique los estados sucesivos completos al realizar la siguiente secuencia de operaciones: +500, -145, -402. Se debe dejar más carga a la derecha.
2: 0 (238) 3 (547) 1 (729) 4

0: (145) 3 3: (238)(402)(512) 1 1: (614) 4 4: (729) -1

3) Dado el siguiente árbol B+ de orden 4, mostrar cómo quedaría el mismo luego de realizar las siguientes operaciones: +131, +115, +50, +70, +10, -43, -50, +65, -10, -23, -63.
Por cada operación, justificar brevemente cada operación realizada e indicar las lecturas y escrituras efectuadas. Ante disparidad de carga en el último nivel, el nodo derecho debe quedar más cargado. Ante disparidad de carga en nodos intermedios, el nodo izquierdo debe quedar más cargado. Política de resolución de underflow: izquierda o derecha.
2: 0 (63) 1
0: (23)(43) 1 1: (63) (67) (90)

4) Dado el siguiente árbol B de orden 5, mostrar cómo quedaría el mismo luego de realizar las siguientes operaciones: +445, -490, -507, -608. Justificar detalladamente cada operación indicando lecturas y escrituras en orden de ocurrencia. Para la resolución de underflow debe utilizar la política a izquierda. Graficar cada operación por separado
2: 0 (320) 1 (490) 4(555) 5(641) 3
0: (13)(153) 1: (325)(341)(431)(460) 4: (500)(507) 5: (608)(633) 3: (790)(923)

5) Dado el árbol B+ de orden 4 que se detalla a continuación, indique los estados sucesivos completos al realizar la siguiente secuencia de operaciones: +450 -804 -725
2: 0(150)3(601)1(804)4
0:(100)3 3(150)(306)(480)1 1:(705)(725)4 4:(804)-1

6) Dado el árbol B+ de orden 4 que se detalla a continuación, indique los estados sucesivos completos al realizar la siguiente secuencia de operaciones: +418 -710 -915
Justifique brevemente cómo realiza cada operación
2: 0(240) 3(690) 1(901)4
0(52)3 3(240)(315)471)1 1(710)4 4(901)(915)-1

7) Dado el siguiente árbol B+ de orden 6, mostrar cómo quedaría el mismo luego de realizar las siguientes operaciones: -301 -363 -130 -222 -209 -80
Política de resolución de underflow: izquierda
2: 0(80)3(101)4(180)1(301)6(505)5
0:(15)(49)(66)3
1:(180)(209)(222)(256)(287)6
3:(80)(91)(95)4
4:(101)(120)(145)153)(177)1
5:(512)(580)(613)-1
6:(301)(383)(404)5

8) Dado el siguiente estado de un árbol B con nodos de a lo sumo 4 registros y al menos 2 (excepto la raíz), indicar qué números de nodos se leen o escriben y en qué orden para la siguiente secuencia de operaciones. Además, deberá graficar el estado completo del árbol luego de cada operación: +900 -711 -552. Usando política derecha
2:0(321)3(477)4(711)5(864)1
0:(16)(242)(299) 3:(356)(365) 4:(552)(628) 5:(780)(810)(816) 1:(895)(939)(952)(990)

9) Dado el árbol B+ que se detalla más abajo, con capacidad para 5 registros en nodos hoja y 5 claves en nodos internos y carga mínima de 3 registros en nodos hoja y 2 claves en nodos internos, muestre los estados sucesivos completos al realizar las siguientes operaciones +90 -20 -50. Indique la secuencia de números de nodos que se leen o escriben para cada operación. Al balancear cantidades impares de registros o claves, dejar más cargado el nodo de la izquierda. Ante un underflow, se resuelve el problema con un único nodo hermano, por defecto el derecho
2: 0(32)3(50)1(72)4(120)6(200)5
0:(10)(20)(24)3
3:(32)(39)(44)1
1:(50)(62)(69)4
4:(72)(80)(87)(100)(115)6
6:(120)(150)(170)(180)5
5:(200)(300)(400)-1

10) Dado el siguiente árbol B+ de orden 4 y con política de resolución de underflow a derecha. Realice las operaciones detalladas a continuación. Deberá dibujar los estados sucesivos del árbol a partir del resultado de cada operación e indicar nodos leídos y escritos en el orden de ocurrencia. Operaciones: +100 -540 -67 -422 -445 -556
2: 0(241)1(422)3(540)4
0:(3)(35)(67)1
1:(241)(300)(329)3
3:(422)(445)4
4:(556)(600)-1

11) Dado el siguiente árbol B de orden 6 y con política de underflow a izquierda. Realice las siguientes operaciones: -70 +109 +330 -321
2: 0(48)1(70)3(93)4(267)5(323)6
0: (17)(33)
1: (50)(64)
3:(85)(86)
4:(105)(169)(193)(205)(210)
5:(321)(322)
6(325)(335)(336)(340)(3341)

12) Dado el árbol B+ con orden 6. Realice las siguientes operaciones +159 -5 -190. Política underflow derecha
2: 0(10)1(60)3(115)4(145)5(179)6
0:(1)(5)->1
1:(34)(44)->3
3(60)(113)->4
4:(120)(125)(131)(139)->5
5:(145)(153)(158)(160)(177)->6
6:(179)(190)->-1

13) Dado un árbol B+ de orden 4 y con política izquierda o derecha, para cada operación dada:
a. Dibuje el árbol resultante.
b. Explique brevemente las decisiones tomadas.
c. Escriba las lecturas y escrituras,
Operaciones: +15, +45, -30, -70
nodo 7: 1 i 2(70)6
nodo 2: 2 i 0(30)1(50)3
nodo 6: 1 i 4(90)5
nodo 0: 3 h(5)(10)(20)->1
nodo 1: 3 h(30)(35)(40)->3
nodo 3: 1 h(60)->4
nodo 4: 2 h(70)->5
nodo 5: 2 h(90)->-1

14) Dado un árbol B+ de orden 5 y con política izquierda o derecha, para cada operación dada, dibuje el árbol resultante y escriba las lecturas y escrituras. Operaciones: +240 -300 -40
nodo 8: 2(70)7
nodo 2: 0(50)4
nodo 7: 5(90)6(120)3(210)9(300)1
nodo 0:(40)->4
nodo 4: (50)->5
nodo 5: (70)(80)->6
nodo 6: (90)(100) ->3
nodo 3: (120)(200)->9
nodo 9:(210)(220)(230)(240)->1
nodo 1: (400)(500)->1

HASHING

1) Realice el proceso de dispersión mediante el método de hashing extensible, sabiendo que cada registro tiene capacidad para dos claves. El número natural indica el orden de llegada de las claves. Deberá explicar los pasos que realiza en cada operación y dibujar los estados sucesivos correspondientes. Justifique brevemente

1	Java	10100111	2	C	10101010
3	Ruby	111110	4	Haskell	1101111
5	Kotlin	110101	6	Python	11110000
7	PHP	1011101	8	SQL	1011011
9	Pascal	110100	10	Smalltalk	11100001

2) Dado el archivo dispersado a continuación, indique los estados sucesivos completos para las siguientes operaciones: +13, +23, +42, +75, -31
Método: Saturación progresiva. f(x) = x mod n. n = tamaño tabla
Al finalizar calcule la densidad de empaquetamiento. Indique lecturas y escrituras por cada operación

dirección	registro	registro
0	66	
1	56	34
2	46	
3		
4	81	
5	16	
6	61	
7	95	
8		
9	31	
10	32	21

3) Dado el siguiente archivo dispersado a continuación, grafique los estados sucesivos para las siguientes operaciones: +58, +63, +78, +61, +89, -12, -78, -23. Técnica de resolución de colisiones: hashing doble.
Por cada operación indicar las lecturas y escrituras efectuadas.
$f1(x) = x\ MOD\ 11$
$f2(x) = x\ MOD\ 7 + 1$

dirección	clave
0	
1	23
2	
3	47
4	15
5	

6	72
7	
8	30
9	
10	

4) Realice el proceso de dispersión mediante el método de hashing extensible, sabiendo que cada registro tiene capacidad para dos claves. El número natural indica el orden de llegada de las mismas.

1	Berlín	00000001	2	Praga	01100011
3	Estocolmo	11110101	4	Londres	01010110
5	Bucarest	00101000	6	Oslo	00110100
7	Copenhague	10110010	8	Roma	01111110
9	Riga	01111100	10	Varsovia	001100000

5) Se debe crear y cargar un archivo directo con capacidad para dos registros con dispersión doble para organizar registros en saturación con los registros cuyas claves se listan a continuación y de manera que su densidad de empaquetamiento resulte del 70% +1050 +213 +718 +669 +96 +523 +194 f2(x)= x MOD3
   a. realice las bajas de las claves 1050 y 194. Indicar L/E por cada operación

6) Se debe crear y cargar un archivo directo con capacidad para un registro por cubeta. Realice las operaciones listadas a continuación indicando cada L/E realizada: +12, +78, +57, +36, +89, +17, +25, +29, -78, -12
Técnica de resolución de colisiones: Saturación progresiva encadenada.
f(x) = x MOD 11. Calcule la densidad de empaquetamiento

7) Para las claves siguientes, realice el proceso de dispersión mediante el método de hashing extensible, sabiendo que cada nodo tiene capacidad para dos registros. El número indica el orden de llegada de las claves

1	Australia	00000001	6	Dinamarca	00110100
2	Alemania	01100011	7	Finlandia	10110010
3	Suiza	11110101	8	Polonia	01111110
4	Holanda	01010110	9	Italia	01111110
5	Bélgica	00101000	10	Francia	00110000

8) Dado el siguiente archivo dispersado más abajo, dibuje los estados sucesivos para +78 +45 +47. Técnica de colisiones: saturación progresiva encadenada. Calcular DE

dirección	enlace	clave
0	-1	22
1	-1	34
2	-1	46
3	-1	
4	-1	59
5	-1	
6	-1	
7	-1	40
8	-1	
9	-1	
10	-1	

9) Dado el estado de un archivo con direcciones con capacidad para dos registros y organizado mediante dispersión doble. Muestre los estados sucesivos completos al realizar la siguiente secuencia de operaciones +197 +287. Segunda función de dispersión: $f2(x) = x \; mod \; 5 + 1$

dirección	clave	clave
0	203	49
1	78	155
2	65	
3	339	
4	221	158
5	54	
6	412	

10) Dado el siguiente archivo dispersado a continuación, grafique los estados sucesivos para las siguientes operaciones: +58 +63 +78 +61 +89 -12 -78 -23. Técnica de resolución: hashing doble. Para cada operación indicar las lecturas y escrituras efectuadas. $f1(x) = x \; MOD \; 11, f2(x) = x \; MOD \; 7 + 1$

dirección	clave
0	
1	23
2	
3	47
4	15
5	
6	72
7	
8	30
9	
10	

11) Dado el siguiente estado de un archivo directo estático con bloques de a lo sumo 3 registros y sin registros en satuación, indicar qué números de bloques se leen o escriben y en qué orden para la siguiente secuencia de operaciones: +50 -816 + 80. Se deberá usar satuación encadenada para la resolución de desbordes. Indicar DE

dirección	clave	clave	clave
0	780	365	810
1	356	816	16
2	242	552	
3	628		
4	299		

12) Realice el proceso de dispersión mediante el método de hashing extensible, sabiendo que cada registro tiene capacidad para dos llaves. El número natural indica el orden de llegada de las mismas. Se debe explicar cada operación detalladamente y graficar el estado completo del archivo por cada operación

1	Alfa	00001001	2	Beta	01110100
3	Gamma	11100010	4	Delta	01011111
5	Épsilon	00110000	6	Rho	01101011
7	Pi	10100110	8	Tau	01101101
9	Psi	01110001	10	Omega	00110111

13) Dado el archivo dispersado a continuación, indique los estados sucesivos completos para las siguientes operaciones: +13 +23 +42 +75 -31. Satuación progresiva $f(x) = x \bmod n, n = tamaño\ de\ la\ tabla$. Calcular densidad de empaquetamiento e indique las lecturas y escrituras por cada operación

dirección	registro	registro
0	66	
1	56	34
2	46	
3		
4	81	
5	16	
6	61	
7	95	
8		
9	31	
10	32	21

14) Dado el archivo dispersado a continuación, grafique los estados sucesivos para las siguientes operaciones +12 +45 +89 + 59 +26 -45. Satuación progresiva encadenada. Indicar las L/E

dirección	enlace	clave
0	-1	
1	-1	78
2	-1	
3	-1	47
4	-1	
5	-1	27
6	-1	
7	-1	51
8	-1	
9	-1	53
10	-1	

15) Realice el proceso de dispersión mediante el método de hashing extensible sabiendo que cada registro tiene capacidad para dos claves

1	Alfa Romero	101001111	2	Peugeot	10101010
3	Audi	00111110	4	Nissan	01101111
5	BMW	01101011	6	Toyota	11110000
7	Fiat	01011101	8	Suzuki	01011011
9	Ford	00110100	10	Renault	11100011

16) Para las claves siguientes, realice el proceso de dispersión mediante el método de hashing extensible, sabiendo que cada nodo tiene capacidad para dos registros

1	mecánica	00000001	2	eléctrica	01100011
3	cinética	11110100	4	térmica	01010110
5	potencial	00101001	6	química	00110101
7	gravitacional	10110000	8	magnética	01111110
9	sonora	01110000	10	nuclear	00110100