

1) Una obra social dispone de un árbol binario de búsqueda con la información de sus afiliados. De cada afiliado se conoce: Número de afiliado, Nro de DNI, Plan (1..5) y año de ingreso a la obra social. El árbol se encuentra ordenado por número de afiliado. Se solicita:

Implementar un módulo que reciba el árbol de afiliados, Nro de DNI Num1, Nro de DNI Num2 y un número de Plan, y retorne un vector ordenado por Nro de DNI del afiliado. El vector debe contener el número de afiliado y DNI de aquellos afiliados cuyo Nro de DNI se encuentra comprendido entre Num1 y Num2 (siendo Num1 menor que Num2) y el Plan se corresponda con el recibido por parámetro. Por norma de la obra social, cada Plan puede contar con a lo sumo 500 afiliados.

2) Una obra social dispone de un árbol binario de búsqueda con la información de sus afiliados. De cada afiliado se conoce: Número de afiliado, Nro de DNI, Categoría (1..10) y año de ingreso a la obra social. El árbol se encuentra ordenado por número de afiliado. Se solicita:

Implementar un módulo que reciba el árbol de afiliados, Nro de DNI Num1, Nro de DNI Num2 y un número de categoría, y retorne un vector ordenado por Nro de DNI del afiliado. El vector debe contener el número de afiliado y DNI de aquellos afiliados cuyo Nro de DNI se encuentra comprendido entre Num1 y Num2 (siendo Num1 menor que Num2) y la categoría se corresponda con la recibida por parámetro. Por norma de la obra social, cada categoría puede contar con a lo sumo 200 afiliados.

3) Una obra social dispone de un árbol binario de búsqueda con la información de sus afiliados. De cada afiliado se conoce: Número de afiliado, Nro de CUIL, Plan (1..10) y año de ingreso a la obra social. El árbol se encuentra ordenado por número de afiliado. Se solicita:

Implementar un módulo que reciba el árbol de afiliados, Nro de CUIL Num1, Nro de CUIL Num2 y un número de Plan, y retorne un vector ordenado por Nro de CUIL del afiliado. El vector debe contener el número de afiliado y CUIL de aquellos afiliados cuyo Nro de CUIL se encuentra comprendido entre Num1 y Num2 (siendo Num1 menor que Num2) y el Plan se corresponda con el recibido por parámetro. Por norma de la obra social, cada Plan puede contar con a lo sumo 250 afiliados.

4) Una obra social dispone de un árbol binario de búsqueda con la información de sus afiliados. De cada afiliado se conoce: Número de afiliado, DNI, Plan (1..15) y año de ingreso a la obra social. El árbol se encuentra ordenado por número de afiliado. Se solicita:

Implementar un módulo que reciba el árbol de afiliados, número de afiliado Num1, número de afiliado Num2 y un número de Plan, y retorne un vector ordenado por número de afiliado. El vector debe contener el número de afiliado y DNI de aquellos afiliados cuyo número de afiliado se encuentra comprendido entre los números de afiliados recibidos (Num1 y Num2, siendo Num1 menor que Num2) y el plan se corresponda con el recibido

por parámetro. Por norma de la obra social, cada plan puede contar con a lo sumo 300 afiliados.

5) Una obra social dispone de un árbol binario de búsqueda con la información de sus afiliados. De cada afiliado se conoce: Número de afiliado, DNI, Categoría (1..10) y año de ingreso a la obra social. El árbol se encuentra ordenado por número de afiliado. Se solicita:

Implementar un módulo que reciba el árbol de afiliados, número de afiliado Num1, número de afiliado Num2 y un número de categoría, y retorne un vector ordenado por número de afiliado. El vector debe contener el número de afiliado y DNI de aquellos afiliados cuyo número de afiliado se encuentra comprendido entre los números de afiliados recibidos (Num1 y Num2, siendo Num1 menor que Num2) y la categoría se corresponda con la recibida por parámetro. Por norma de la obra social, cada categoría puede contar con a lo sumo 200 afiliados.

6) Una ciudad necesita un sistema para administrar la limpieza de sus esquinas. El sistema necesita un módulo que lea la información de cada esquina que se limpia: número de avenida (1..10), altura (un número entero), cantidad de flores recolectadas y cantidad de papeles recolectados. A partir de esa información, debe generar una estructura que almacene las esquinas agrupadas por número de avenida. Para cada avenida, las esquinas deben almacenarse de manera ordenada por altura. La lectura finaliza con el número de avenida 0, el cual no se procesa.

Se pide implementar un programa en Pascal que incluya:

a) Un módulo Cargar que (de manera aleatoria) genere datos de las esquinas y los almacene en una estructura adecuada.

b) Un módulo que reciba la estructura generada en a) y, usando la técnica de merge acumulador, genere una lista que contenga para cada altura, la cantidad total de flores y de papeles recolectados.

7) Una ciudad necesita un sistema para administrar la limpieza de sus esquinas. El sistema necesita un módulo que lea la información de cada esquina que se limpia: horario de recolección (1..10) número de calle, número de avenida y cantidad de bolsas recolectadas. A partir de esa información, debe generar una estructura que almacene las esquinas agrupadas por horario. Para cada horario de recolección, las esquinas deben almacenarse de manera ordenada por número de avenida. La lectura finaliza con el horario 0, el cual no se procesa.

Se pide implementar un programa en Pascal que incluya:

a) Un módulo Cargar que (de manera aleatoria) genere datos de las esquinas y los

almacene en una estructura adecuada.

b) Un módulo que reciba la estructura generada en a) y, usando merge acumulador, genere una lista que contenga para cada número de avenida cuál fue la cantidad máxima de bolsas recolectada entre sus esquinas y la ubicación de la esquina (es decir, su número de avenida y calle).

8) Una ciudad necesita un sistema para administrar la limpieza de sus esquinas. Para ello, el sistema necesita un módulo que lea la información de cada esquina que se limpia: número de calle, número de avenida, cantidad de objetos recolectados y número identificador del camión (1..10) que se encargó de la limpieza de la esquina. A partir de esa información, debe generar una estructura que almacene las esquinas agrupadas por número de camión.

Para cada camión, las esquinas deben ordenarse por número de avenida. La lectura finaliza con el número de camión 0, el cual no se procesa.

Se pide implementar un programa en Pascal que incluya:

a) Un módulo Cargar que (de manera aleatoria) genere datos de las esquinas y los almacene en una estructura adecuada.

b) Un módulo que reciba la estructura generada en a) y (usando merge acumulador) obtenga para cada avenida el total de objetos recolectados (entre todos los camiones). Genere un ABB que contenga número de avenida y el total de objetos recolectados en la avenida, ordenado por total de objetos recolectados.

9) Una ciudad necesita un sistema para administrar la limpieza de sus esquinas. Para ello, el sistema necesita un módulo que lea la información de cada esquina que se limpia: número de calle, número de avenida, cantidad de objetos recolectados y número identificador del camión (1..10) que se encargó de la limpieza de la esquina. A partir de esa información, debe generar una estructura que almacene las esquinas agrupadas por número de camión.

Para cada camión, las esquinas deben ordenarse por número de calle. La lectura finaliza con el número de camión 0, el cual no se procesa.

Se pide implementar un programa en Pascal que incluya:

a) Un módulo Cargar que (de manera aleatoria) genere datos de las esquinas y los almacene en una estructura adecuada.

b) Un módulo que reciba la estructura generada en a) y (usando merge acumulador) obtenga para cada calle, el total de esquinas que se limpiaron en ella (entre todos los

camiones). Genere un ABB que contenga número de calle y total de esquinas limpiadas, ordenado por total de esquinas limpiadas.

10) Una ciudad necesita un sistema para administrar la limpieza de sus esquinas. Para ello, el sistema necesita un módulo que lea la información de cada esquina que se limpia: número de calle, número de avenida, cantidad de objetos recolectados y número identificador del camión (1..10) que se encargó de la limpieza de la esquina. A partir de esa información, debe generar una estructura que almacene las esquinas agrupadas por número de camión.

Para cada camión, las esquinas deben ordenarse por número de calle. La lectura finaliza con el número de camión 0, el cual no se procesa.

Se pide implementar un programa en Pascal que incluya:

a) Un módulo Cargar que (de manera aleatoria) genere datos de las esquinas y los almacene en una estructura adecuada.

b) Un módulo que reciba la estructura generada en a) y (usando merge acumulador) obtenga para cada calle, el total de esquinas que se limpiaron en ella (entre todos los camiones). Genere un ABB que contenga número de calle y total de esquinas limpiadas, ordenado por total de esquinas limpiadas.

11) Un psicólogo necesita un sistema para administrar a sus pacientes. De cada paciente registra: dni, cód. de paciente, obra social (1: ioma 2: pami 3: osde 4: galeno 5: no tiene) y costo abonado por sesión. Implemente un programa que:

a) Genere un ABB ordenado por dni. Para ello, genere información hasta el paciente con dni 0.

A partir del ABB, realice módulos (uno por inciso) para:

b) Generar una estructura con dni y cód de paciente de los pacientes de ioma, ordenados por dni descendente.

c) Dado un dni, modificar la obra social de dicho paciente a una recibida. Considere que el paciente puede no existir.

NOTA: Realice el programa principal que invoque a los módulos desarrollados.

12) Una aerolínea dispone de un árbol binario de búsqueda con la información de sus

empleados. De cada empleado se conoce: Número de legajo, Dni, Categoría (1..20) y año de ingreso a la empresa. El árbol se encuentra ordenado por número de dni. Se solicita:

a) Implementar un módulo que reciba el árbol de empleados, número de DNI "A", número de DNI "B" y un número de categoría, y retorne un vector ordenado por número de DNI. El vector debe contener el número de DNI y número de legajo de aquellos empleados cuyo número de DNI se encuentra comprendido entre los números de DNI recibidos ("A" y "B", siendo "A" menor que "B") y la categoría se corresponda con la recibida por parámetro. Por norma de la empresa, cada categoría puede contar con a lo sumo 250 empleados

b) Implementar un módulo recursivo que reciba la información generada en "a" y retorne el promedio de los números de legajo.

13) Una aerolínea dispone de un árbol binario de búsqueda con la información de sus empleados. De cada empleado se conoce: Número de legajo, Dni, Categoría (1..20) y año de ingreso a la empresa. El árbol se encuentra ordenado por número de dni. Se solicita:

a) Implementar un módulo que reciba el árbol de empleados, número de DNI "A", número de DNI "B" y un número de categoría, y retorne un vector ordenado por número de DNI. El vector debe contener el número de DNI y número de legajo de aquellos empleados cuyo número de DNI se encuentra comprendido entre los números de DNI recibidos ("A" y "B", siendo "A" menor que "B") y la categoría se corresponda con la recibida por parámetro. Por norma de la empresa, cada categoría puede contar con a lo sumo 50 empleados.

b) Implementar un módulo **recursivo** que reciba la información generada en "a" y retorne el máximo número de legajo.

14) Una aerolínea dispone de un árbol binario de búsqueda con la información de sus empleados. De cada empleado se conoce: Número de legajo, Dni, Categoría (1..20) y año de ingreso a la empresa. El árbol se encuentra ordenado por número de dni. Se solicita:

a) Implementar un módulo que reciba el árbol de empleados, número de DNI "A", número de DNI "B" y un número de categoría, y retorne un vector ordenado por número de DNI. El vector debe contener el número de DNI y número de legajo de aquellos empleados cuyo número de DNI se encuentra comprendido entre los números de DNI recibidos ("A" y "B", siendo "A" menor que "B") y la categoría se corresponda con la recibida por parámetro. Por norma de la empresa, cada categoría puede contar con a lo sumo 150 empleados.

b) Implementar un módulo recursivo que reciba la información generada en "a" y retorne el mínimo número de legajo.

15) Una aerolínea dispone de un árbol binario de búsqueda con la información de sus empleados. De cada empleado se conoce: Número de legajo, Dni, Categoría (1..20) y año de ingreso a la empresa. El árbol se encuentra ordenado por número de legajo. Se solicita:

a) Implementar un módulo que reciba el árbol de empleados, número de legajo "A", número de legajo "B" y un número de categoría, y retorne un vector ordenado por número de legajo. El vector debe contener el número de legajo y Dni de aquellos empleados cuyo número de legajo se encuentra comprendido entre los números de legajo recibidos ("A" y "B", siendo "A" menor que "B") y la categoría se corresponda con la recibida por parámetro. Por norma de la empresa, cada categoría puede contar con a lo sumo 50 empleados.

b) Implementar un módulo recursivo que reciba la información generada en "a" y retorne el máximo número de Dni.

16) Una aerolínea dispone de un árbol binario de búsqueda con la información de sus empleados. De cada empleado se conoce: Número de legajo, Dni, Categoría (1..20) y año de ingreso a la empresa. El árbol se encuentra ordenado por número de legajo. Se solicita:

a) Implementar un módulo que reciba el árbol de empleados, número de legajo "A", número de legajo "B" y un número de categoría, y retorne un vector ordenado por número de legajo. El vector debe contener el número de legajo y Dni de aquellos empleados cuyo número de legajo se encuentra comprendido entre los números de legajo recibidos ("A" y "B", siendo "A" menor que "B") y la categoría se corresponda con la recibida por parámetro. Por norma de la empresa, cada categoría puede contar con a lo sumo 150 empleados.

b) Implementar un módulo **recursivo** que reciba la información generada en "a" y retorne el mínimo número de Dni.

17) Una aerolínea dispone de un árbol binario de búsqueda con la información de sus empleados. De cada empleado se conoce: Número de legajo, Dni, Categoría (1..20) y año de ingreso a la empresa. El árbol se encuentra ordenado por número de legajo. Se solicita:

a) Implementar un módulo que reciba el árbol de empleados, número de legajo "A", número de legajo "B" y un número de categoría, y retorne un vector ordenado por número de legajo. El vector debe contener el número de legajo y Dni de aquellos empleados cuyo

número de legajo se encuentra comprendido entre los números de legajo recibidos ("A" y "B", siendo "A" menor que "B") y la categoría se corresponda con la recibida por parámetro. Por norma de la empresa, cada categoría puede contar con a lo sumo 250 empleados.

b) Implementar un módulo recursivo que reciba la información generada en "a" y retorne el promedio de los números de Dni.

18) Un psicólogo necesita un sistema para administrar a sus pacientes. De cada paciente registra: dni, cód. de paciente, obra social (1: ioma 2: pami 3: osde 4: galeno 5: no tiene) y costo abonado por sesión. Implemente un programa que:

a) Genere un ABB ordenado por dni. Para ello, genere información hasta el paciente con dni 0.

A partir del ABB, realice módulos (uno por inciso) para:

b) Generar una estructura con dni y cód de paciente de los pacientes de ioma, ordenados por dni descendente.

c) Dado un dni, modificar la obra social de dicho paciente a una recibida. Considere que el paciente puede no existir.

NOTA: Realice el programa principal que invoque a los módulos desarrollados.

19) Un psicólogo necesita un sistema para administrar a sus pacientes. De cada paciente registra: dni, cód. de paciente, obra social (1: ioma 2: pami 3: osde 4: galeno 5: no tiene) y costo abonado por sesión. Implemente un programa que:

a) Genere un ABB ordenado por cód. de paciente. Para ello, genere información hasta el paciente con dni 0.

A partir del ABB, realice módulos (uno por inciso) para:

b) Generar una estructura con código y dni de los pacientes de osde, sabiendo que cómo máximo son 100. La estructura debe ordenarse por código ascendentemente .

c) Dado un código de paciente, aumentar el costo abonado por sesión en un monto recibido. Considere que el paciente puede no existir.

NOTA: Realice el programa principal que invoque a los módulos desarrollados.

20) Se lee información acerca de las ventas de productos realizadas en las sucursales de una empresa. De cada venta se conoce fecha, código de producto, cantidad vendida y monto total de la venta. La lectura finaliza cuando se lee el código de producto -1, el cual no se procesa. Implementar un programa para que a partir de la información leída, resuelva los siguientes ítems:

a) Generar un árbol binario de búsqueda ordenado por código de producto, donde cada nodo contenga el código del producto y el monto total vendido. El código de producto no puede repetirse en el árbol.

b) Realizar un módulo que reciba el árbol generado en a. y un código de producto y retorne la suma de todos los montos vendidos para los códigos de productos mayores al código recibido. Nota: La información se lee en forma desordenada. Puede existir más de una venta para un mismo código de producto.

21) Se lee información acerca de las ventas de productos realizadas en las 5 sucursales de una empresa. Cada sucursal realizó a lo sumo 200 ventas. De cada venta se conoce el código de producto, cantidad vendida y monto total de la venta. Las ventas de cada sucursal se leen de manera consecutiva y ordenadas por código de producto. La lectura por cada sucursal finaliza al completar las 200 ventas o cuando se lee el código de producto -1, el cual no se procesa. Implementar un programa para que a partir de la información leída, resuelva los siguientes ítems:

a) Utilizando la técnica de merge o merge acumulador según corresponda, generar una lista que contenga la cantidad total vendida para cada código de producto, ordenada por código de producto.

b) Realizar un módulo recursivo que reciba la lista generada en el punto a y retorne la cantidad de productos para los cuales la cantidad total vendida supera las 500 unidades.

22) Se lee información acerca de las ventas de productos realizadas en las 5 sucursales de una empresa. Cada sucursal realizó a lo sumo 200 ventas. De cada venta se conoce el código de producto, cantidad vendida y monto total de la venta. Las ventas de cada sucursal se leen de manera consecutiva y ordenadas por código de producto. La lectura por cada sucursal finaliza al completar las 200 ventas o cuando se lee el código de producto -1, el cual no se procesa. Implementar un programa para que a partir de la información leída, resuelva los siguientes ítems:

a) Utilizando la técnica de merge o merge acumulador según corresponda, generar una lista que contenga la cantidad total vendida para cada código de producto, ordenada por código de producto.

b) Realizar un módulo recursivo que reciba la lista generada en el punto a y retorne la cantidad de productos para los cuales la cantidad total vendida supera las 500 unidades.

23) Se lee información acerca de las ventas de productos realizadas en las 5 sucursales de una empresa. Cada sucursal realizó a lo sumo 180 ventas. De cada venta se conoce el código de producto, cantidad vendida y monto total de la venta. Las ventas de cada sucursal se leen de manera consecutiva y ordenadas por código de producto. La lectura por cada sucursal finaliza al completar las 180 ventas o cuando se lee el código de producto -1, el cual no se procesa. Implementar un programa para que a partir de la información leída, resuelva los siguientes ítems:

a) Utilizando la técnica de merge o merge acumulador según corresponda, generar una lista que contenga el monto total vendido para cada código de producto, ordenada por código de producto.

b) Realizar un módulo recursivo que reciba la lista generada en el punto a y retorne la cantidad de productos para los cuales el monto total vendido es inferior a 300.000 pesos.

24) Se lee información acerca de las ventas de productos realizadas en las 5 sucursales de una empresa. Cada sucursal realizó a lo sumo 250 ventas. De cada venta se conoce el código de producto, cantidad vendida y monto total de la venta. Las ventas de cada sucursal se leen de manera consecutiva y ordenadas por código de producto. La lectura por cada sucursal finaliza al completar las 250 ventas o cuando se lee el código de producto -1, el cual no se procesa. Implementar un programa para que a partir de la información leída, resuelva los siguientes ítems:

a) Utilizando la técnica de merge o merge acumulador según corresponda, generar una lista que contenga la cantidad total vendida para cada código de producto, ordenada por código de producto.

b) Realizar un módulo recursivo que reciba la lista generada en el punto a y retorne la suma de las cantidades vendidas para los productos con código menor que 80.

25) Un psicólogo necesita un sistema para administrar a sus pacientes. De cada paciente registra: dni, cód. de paciente, obra social (1: ioma 2: pami 3: osde 4: galeno 5: no tiene) y costo abonado por sesión. Implemente un programa que:

a) Genere un ABB ordenado por dni. Para ello, genere información hasta el paciente con dni 0.

A partir del ABB, realice módulos (uno por inciso) para:

b) Generar una estructura con dni y cód de paciente de los pacientes de ioma, ordenados por dni descendente.

c) Dado un dni, modificar la obra social de dicho paciente a una recibida. Considere que el paciente puede no existir.

NOTA: Realice el programa principal que invoque a los módulos desarrollados.

26) Un psicólogo necesita un sistema para administrar a sus pacientes. De cada paciente registra: dni, cód. de paciente, obra social (1: ioma 2: pami 3: osde 4: galeno 5: no tiene) y costo abonado por sesión. Implemente un programa que:

a) Genere un ABB ordenado por cód. de paciente. Para ello, genere información hasta el paciente con dni 0.

A partir del ABB, realice módulos (uno por inciso) para:

b) Generar una estructura con código y dni de los pacientes de osde, sabiendo que cómo máximo son 100. La estructura debe ordenarse por código ascendentemente .

c) Dado un código de paciente, aumentar el costo abonado por sesión en un monto recibido. Considere que el paciente puede no existir.

NOTA: Realice el programa principal que invoque a los módulos desarrollados.

27) Un psicólogo necesita un sistema para administrar a sus pacientes. De cada paciente registra: dni, cód. de paciente, obra social (1: ioma 2: pami 3: osde 4: galeno 5: no tiene) y costo abonado por sesión. Implemente un programa que:

a) Genere un ABB ordenado por dni. Para ello, genere información hasta el paciente con dni 0.

A partir del ABB, realice módulos (uno por inciso) para:

b) Generar una estructura con los códigos de los pacientes de osde cuyo dni está comprendido entre dos valores.

c) Aumentar el costo abonado por sesión de todos los pacientes en un monto recibido.

NOTA: Realice el programa principal que invoque a los módulos desarrollados.

28) Un psicólogo necesita un sistema para administrar a sus pacientes. De cada paciente registra: dni, cód. de paciente, obra social (1: ioma 2: pami 3: osde 4: galeno 5: no tiene) y costo abonado por sesión. Implemente un programa que:

a) Genere un ABB ordenado por cód. de paciente. Para ello, genere información hasta el paciente con dni 0.

A partir del ABB, realice módulos (uno por inciso) para:

b) Generar una estructura con la cantidad de pacientes de cada obra social.

c) Calcular la cantidad de pacientes cuyo costo abonado por sesión está comprendido entre dos valores.

NOTA: Realice el programa principal que invoque a los módulos desarrollados.

29) Un psicólogo necesita un sistema para administrar a sus pacientes. De cada paciente registra: dni, cód. de paciente, obra social (1: ioma 2: pami 3: osde 4: galeno 5: no tiene) y costo abonado por sesión. Implemente un programa que:

a) Genere un ABB ordenado por dni. Para ello, genere información hasta el paciente con dni 0.

A partir del ABB, realice módulos (uno por inciso) para:

b) Generar una estructura con dni y código de los pacientes de galeno, ordenados por dni ascendente.

c) Informar los datos de los pacientes cuyo dni está comprendido entre dos valores.

NOTA: Realice el programa principal que invoque a los módulos desarrollados

30) Un psicólogo necesita un sistema para administrar a sus pacientes. De cada paciente

registra: dni, cód. de paciente, obra social (1: ioma 2: pami 3: osde 4: galeno 5: no tiene) y costo abonado por sesión. Implemente un programa que:

a) Genere un ABB ordenado por cód. de paciente. Para ello, genere información hasta el paciente con dni 0.

A partir del ABB, realice módulos (uno por inciso) para:

- b) Generar una estructura con código y dni de los pacientes de ioma, sabiendo que cómo máximo son 50. La estructura debe ordenarse por código descendentemente.
- c) Informar los datos de los pacientes cuyo código está comprendido entre dos valores.

NOTA: Realice el programa principal que invoque a los módulos desarrollados.

31) Un psicólogo necesita un sistema para administrar a sus pacientes. De cada paciente registra: dni, cód. de paciente, obra social (1: ioma 2: pami 3: osde 4: galeno 5: no tiene) y costo abonado por sesión. Implemente un programa que:

a) Genere un ABB ordenado por dni. Para ello, genere información hasta el paciente con dni 0.

A partir del ABB, realice módulos (uno por inciso) para:

- b) Generar una estructura con dni y cód de paciente de los pacientes de ioma, ordenados por dni descendente.
- c) Dado un dni, modificar la obra social de dicho paciente a una recibida. Considere que el paciente puede no existir.

NOTA: Realice el programa principal que invoque a los módulos desarrollados.

32) Un psicólogo necesita un sistema para administrar a sus pacientes. De cada paciente registra: dni, cód. de paciente, obra social (1: ioma 2: pami 3: osde 4: galeno 5: no tiene) y costo abonado por sesión. Implemente un programa que:

a) Genere un ABB ordenado por dni. Para ello, genere información hasta el paciente con dni 0.

A partir del ABB, realice módulos (uno por inciso) para:

b) Generar una estructura con dni y cód de paciente de los pacientes de ioma, ordenados por dni descendente.

c) Dado un dni, modificar la obra social de dicho paciente a una recibida. Considere que el paciente puede no existir.

NOTA: Realice el programa principal que invoque a los módulos desarrollados.

33) Un psicólogo necesita un sistema para administrar a sus pacientes. De cada paciente registra: dni, cód. de paciente, obra social (1: ioma 2: pami 3: osde 4: galeno 5: no tiene) y costo abonado por sesión. Implemente un programa que:

a) Genere un ABB ordenado por cód. de paciente. Para ello, genere información hasta el paciente con dni 0.

A partir del ABB, realice módulos (uno por inciso) para:

b) Generar una estructura con código y dni de los pacientes de osde, sabiendo que cómo máximo son 100. La estructura debe ordenarse por código ascendentemente .

c) Dado un código de paciente, aumentar el costo abonado por sesión en un monto recibido. Considere que el paciente puede no existir.

NOTA: Realice el programa principal que invoque a los módulos desarrollados.

34) Un psicólogo necesita un sistema para administrar a sus pacientes. De cada paciente registra: dni, cód. de paciente, obra social (1: ioma 2: pami 3: osde 4: galeno 5: no tiene) y costo abonado por sesión. Implemente un programa que:

a) Genere un ABB ordenado por cód. de paciente. Para ello, genere información hasta el paciente con dni 0.

A partir del ABB, realice módulos (uno por inciso) para:

- b) Generar una estructura con la cantidad de pacientes de cada obra social.
- c) Calcular la cantidad de pacientes cuyo costo abonado por sesión está comprendido entre dos valores.

NOTA: Realice el programa principal que invoque a los módulos desarrollados.

35) Un psicólogo necesita un sistema para administrar a sus pacientes. De cada paciente registra: dni, cód. de paciente, obra social (1: ioma 2: pami 3: osde 4: galeno 5: no tiene) y costo abonado por sesión. Implemente un programa que:

- a) Genere un ABB ordenado por dni. Para ello, genere información hasta el paciente con dni 0.

A partir del ABB, realice módulos (uno por inciso) para:

- b) Generar una estructura con los códigos de los pacientes de osde cuyo dni está comprendido entre dos valores.
- c) Aumentar el costo abonado por sesión de todos los pacientes en un monto recibido.

NOTA: Realice el programa principal que invoque a los módulos desarrollados

36) Un psicólogo necesita un sistema para administrar a sus pacientes. De cada paciente registra: dni, cód. de paciente, obra social (1: ioma 2: pami 3: osde 4: galeno 5: no tiene) y costo abonado por sesión. Implemente un programa que:

- a) Genere un ABB ordenado por cód. de paciente. Para ello, genere información hasta el paciente con dni 0.

A partir del ABB, realice módulos (uno por inciso) para:

- b) Generar una estructura con la cantidad de pacientes de cada obra social.
- c) Calcular la cantidad de pacientes cuyo costo abonado por sesión está comprendido entre dos valores.

NOTA: Realice el programa principal que invoque a los módulos desarrollados.

37) Un psicólogo necesita un sistema para administrar a sus pacientes. De cada paciente registra: dni, cód. de paciente, obra social (1: ioma 2: pami 3: osde 4: galeno 5: no tiene) y costo abonado por sesión. Implemente un programa que:

- a) Genere un ABB ordenado por cód. de paciente. Para ello, genere información hasta el paciente con dni 0.

A partir del ABB, realice módulos (uno por inciso) para:

- b) Generar una estructura con código y dni de los pacientes de ioma, sabiendo que cómo máximo son 50. La estructura debe ordenarse por código descendentemente.
- c) Informar los datos de los pacientes cuyo código está comprendido entre dos valores.

NOTA: Realice el programa principal que invoque a los módulos desarrollados.

38) Se lee información acerca de las ventas de productos realizadas en las sucursales de una empresa. De cada venta se conoce fecha, código de producto, cantidad vendida y monto total de la venta. La lectura finaliza cuando se lee el código de producto -1, el cual no se procesa. Implementar un programa para que a partir de la información leída, resuelva los siguientes ítems:

- a) Generar un árbol binario de búsqueda ordenado por código de producto, donde cada nodo contenga el código del producto y la cantidad total vendida. El código de producto no puede repetirse en el árbol.

b) Realizar un módulo que reciba el árbol generado en a. y una cantidad, y retorne la cantidad de

códigos de producto cuya cantidad total vendida superan la cantidad recibida.

Nota: La información se lee en forma desordenada. Puede existir mas de una venta para un mismo código de producto.

39) Se lee información acerca de las ventas de productos realizadas en las sucursales de una empresa. De

cada venta se conoce fecha, código de producto, cantidad vendida y monto total de la venta. La lectura

finaliza cuando se lee el código de producto -1, el cual no se procesa. Implementar un programa para

que a partir de la información leída, resuelva los siguientes ítems:

a) Generar un árbol binario de búsqueda ordenado por código de producto, donde cada nodo

contenga el código del producto y el monto total vendido. El código de producto no puede repetirse en el árbol.

b) Realizar un módulo que reciba el árbol generado en a. y un código de producto y retorne la suma

de todos los montos vendidos para los códigos de productos mayores al código recibido.

Nota: La información se lee en forma desordenada. Puede existir más de una venta para un mismo código de producto.

40) Se lee información acerca de las ventas de productos realizadas en las sucursales de una empresa. De

cada venta se conoce fecha, código de producto, cantidad vendida y monto total de la venta. La lectura

finaliza cuando se lee el código de producto -1, el cual no se procesa. Implementar un programa para

que a partir de la información leída, resuelva los siguientes ítems:

a) Generar un árbol binario de búsqueda ordenado por código de producto, donde cada nodo contenga el código del producto y la cantidad total vendida. El código de producto no puede repetirse en el árbol.

b) Realizar un módulo que reciba el árbol generado en a. y dos códigos de producto y retorne la suma de todas las cantidades vendidas para los códigos de productos comprendidos en el rango.

Nota: La información se lee en forma desordenada. Puede existir mas de una venta para un mismo código de producto.

41) Se lee información acerca de las ventas de productos realizadas en las 5 sucursales de una empresa.

Cada sucursal realizó a lo sumo 200 ventas. De cada venta se conoce el código de producto, cantidad

vendida y monto total de la venta. Las ventas de cada sucursal se leen de manera consecutiva y

ordenadas por código de producto. La lectura por cada sucursal finaliza al completar las 200 ventas o

cuando se lee el código de producto -1, el cual no se procesa. Implementar un programa para que a

partir de la información leída, resuelva los siguientes ítems:

a) Utilizando la técnica de merge o merge acumulador según corresponda, generar una lista que

contenga la cantidad total vendida para cada código de producto, ordenada por código de producto.

b) Realizar un módulo recursivo que reciba la lista generada en el punto a y retorne la cantidad de productos para los cuales la cantidad total vendida supera las 500 unidades.

42) Se lee información acerca de las ventas de productos realizadas en las 5 sucursales de una empresa.

Cada sucursal realizó a lo sumo 180 ventas. De cada venta se conoce el código de producto, cantidad

vendida y monto total de la venta. Las ventas de cada sucursal se leen de manera consecutiva y

ordenadas por código de producto. La lectura por cada sucursal finaliza al completar las 180 ventas o

cuando se lee el código de producto -1, el cual no se procesa. Implementar un programa para que a

partir de la información leída, resuelva los siguientes ítems:

a) Utilizando la técnica de merge o merge acumulador según corresponda, generar una lista que

contenga el monto total vendido para cada código de producto, ordenada por código de producto.

b) Realizar un módulo recursivo que reciba la lista generada en el punto a y retorne la cantidad de

productos para los cuales el monto total vendido es inferior a 300.000 pesos.

44) Se lee información acerca de las ventas de productos realizadas en las 5 sucursales de una empresa.

Cada sucursal realizó a lo sumo 250 ventas. De cada venta se conoce el código de producto, cantidad

vendida y monto total de la venta. Las ventas de cada sucursal se leen de manera consecutiva y

ordenadas por código de producto. La lectura por cada sucursal finaliza al completar las 250 ventas o

cuando se lee el código de producto -1, el cual no se procesa. Implementar un programa para que a

partir de la información leída, resuelva los siguientes ítems:

a) Utilizando la técnica de merge o merge acumulador según corresponda, generar una lista que

contenga la cantidad total vendida para cada código de producto, ordenada por código de

producto.

b) Realizar un módulo recursivo que reciba la lista generada en el punto a y retorne la suma de las

cantidades vendidas para los productos con código menor que 80.
