EJERCICIOS Y EJEMPLOS CON ÁRBOLES 2017

Ejemplo 1:

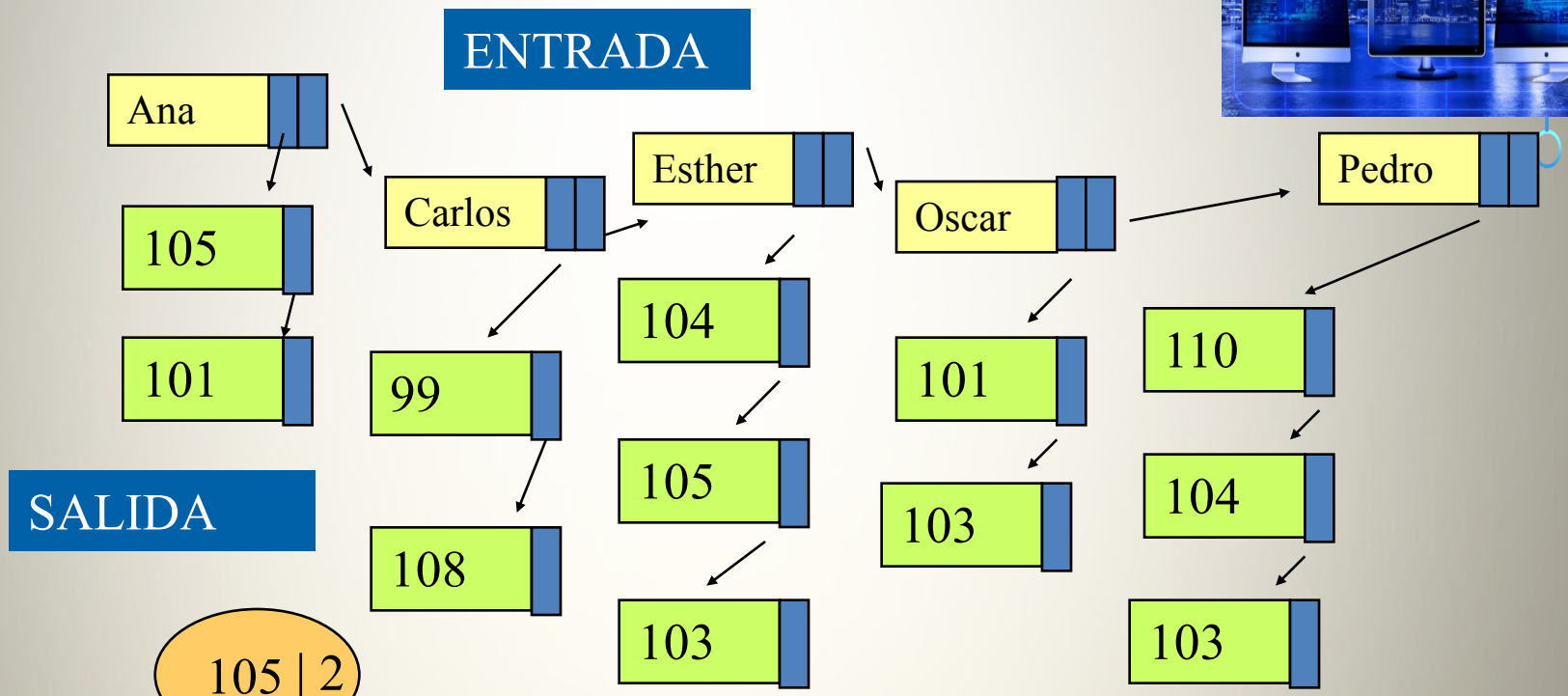
Se tiene una lista de personas anotadas en una maestría ordenada por apellido.

Para cada persona se tiene una lista de los cursos ya aprobados (los cursos aparecen identificados por su código).

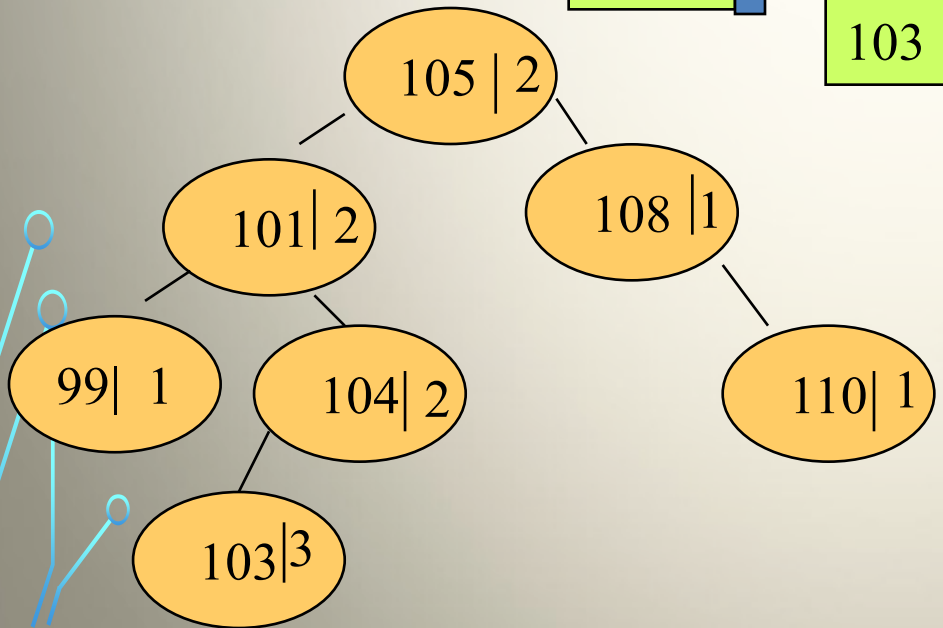
Se pide generar un árbol binario ordenado de cursos donde para cada curso se tenga la cantidad de alumnos que lo aprobó.

Nota: el orden del árbol es por código de curso.

Ejemplo 1: Planteo de la información



SALIDA



¿Qué pasa si el alumno no tiene ninguna materia aprobada?

Ejemplo 1: estrategia de solución

Una solución posible…

{Recorrer la lista de alumnos}

¿Cómo modularizar correctamente este algoritmo?

{ Por cada alumnos tomar la lista de cursos del alumno y hacer:}

{Por cada elemento de la lista insertar en el Ordenado en el árbol binario o sumar uno si ya existe el nodo}

Ejemplo 1: estructuras de datos

```
Type
   ListaCursos = ^nodocurso;

   Nodocurso = record
                     Codigo:string [4];
                      Sig : ListaCursos;
                   End;

  ListaAlumnos = ^Alumno;

  Alumno = record
                  Apellido: string [20];
                  Cursos: ListaCursos;
                  Sig : ListaAlumnos;
                End;

ArbolCursos = ^NodoArbol;

NodoArbol =  record
                     Codigo : string[4];
                     Cant : integer;
                     Izq : ArbolCursos;
                     Der: ArbolCursos;
                  End;
```

Hacer el módulo de creación del árbol binario ordenado

```
Var
   LC, listacur: ListaCursos;
   LA, listaalu: ListaAlumnos;
   AC, arbol : ArbolCursos;
```

Ejemplo 2

a) Realizar un módulo que Informar la cantidad de alumnos que tienen los cursos con código menor a 105.

```
Procedure cantidades (a: arbolcursos);
  begin
    If a <> Nil then
                    If (a^.codigo < 105) then begin
                                                    write (a^.cant);
                                                    Cantidades (a^.der);
                                                    Cantidades (a^.izq);
                                                  End
                                                  Else cantidades (a^.izq);
 End;
```

Ejemplo 3

b) Hacer un módulo que informe cuales son las cantidades de alumnos de los cursos con código comprendidos entre 105 y 110.

```
Procedure intervaloCantidades (a: arbolcursos);
  begin
    If a <> Nil then
                  if (a^.codigo >= 105) then
                        if (a^.codigo <= 110) then begin
                                                    Write (a^.cant);
                                                    intervaloCantidades (a^.izq);
                                                    intervaloCantidades (a^.der);
                                                  End
                                                  Else intervaloCantidades (a^.izq);
                                            Else intervaloCantidades (a^.der);
End;
```

Si bien esta solución informa correctamente, ¿Recorre ramas de más dentro del intervalo?

Ejemplo 3: Solución mejorada

b) Hacer un módulo que informe cuales son las cantidades de alumnos de los cursos con código comprendidos entre 105 y 110.

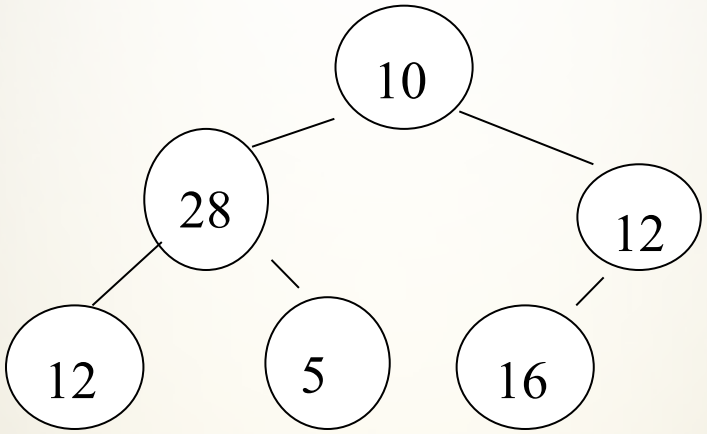
Podemos pensar en evitar recorrer de más dentro del intervalo

```
Procedure intervaloCantidades (a: arbolcursos);
  begin
    If a <> Nil then
                if (a^.codigo >= 105) then
                        if (a^.codigo <= 110) then begin
                                Write (a^.cant);
                                if (a^.codigo = 105) then
                                  intervaloCantidades (a^.der);
                                  else if (a^.codigo = 110)
                                        intervaloCantidades (a^.izq);
                                        else  begin
                                              intervaloCantidades (a^.izq);
                                              intervaloCantidades (a^.der);
                                    end
                              End
                              Else intervaloCantidades (a^.izq);
                      Else intervaloCantidades (a^.der);
End;
```

Saber que encontré el valor de uno de los extremos ya me da información sobre lo que pasa a su izquierda o derecha

Ejemplo 4 Arboles BINARIOS

Dado un árbol binario donde cada uno de sus nodos tiene como dato un número entero. Ejemplo:10 28 12 5 16 12



Observa que NO está ordenado

a) Imprimir el peso de cada rama del árbol en términos de la suma de los valores que contienen sus nodos.

Ejemplo 4 Arboles BINARIOS

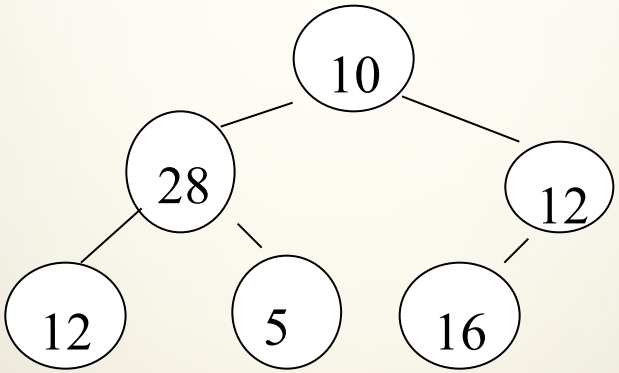
```
type
 Arbol = ^nodo;
  Nodo = record
                Dato: integer;
                Izq: arbol;
                Der: arbol;
              End;
Var
    A: arbol; cant: integer;
    max : integer {para parte (b)}
Procedure Imprimir (a: arbol; cant: integer);
  Begin
      Cant := cant + a^.dato;
      If (a^.izq <> Nil) then Imprimir (a^.izq, cant);
      If (a^.der <> Nil) then Imprimir (a^.der, cant);
      If (a^.izq = Nil) and  (a^.der = Nil) then
                  write ("La suma de la rama es:", cant);
  End;
```

```
{prog. ppal}
Begin
....
Cant := 0;
If (a <> nil) then
    Imprimir (a, cant);
...
End.
```

Ejercitación Arboles BINARIOS

2.b)Supongamos que se desea calcular la rama del árbol que tiene más peso en términos de la suma de los valores que contienen sus nodos. Se pide informar la suma máxima.

Resultado: La rama que tiene máxima suma es la que totaliza 50



Observa que no está ordenado

Ejercitación Arboles BINARIOS: este algoritmo me permite tratar cada rama del árbol

```
Procedure ImprimirModificado (a: arbol; cant: integer;
                                        var  max:integer);
  Begin
     Cant := cant + a^.dato;
     If (a^.izq <> Nil) then ImprimirModificado (a^.izq, cant, max);
     If (a^.der <> Nil) then ImprimirModificado (a^.der, cant, max);
     If (a^.izq = Nil) and  (a^.der = Nil) then
                                        If cant > max then max := cant;
   End;


{Prog Ppal}
  Begin
     ....
     max:= -1; Cant := 0;
     if (a <> nil) then ImprimirModificado (a, cant, max);
     Write (max);
     ...
  End.
```

¿Funciona si el árbol estuviera ordenado?