

ESTRUCTURA DE DATOS

UNIDAD 2

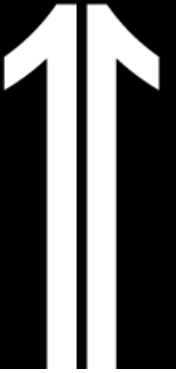
Mg. Cinthia J. Calderon Aquino

Semana 09

UPN.EDU.PE

PRESENTACIÓN DE LA SESIÓN

Logro de la Sesión y Temario

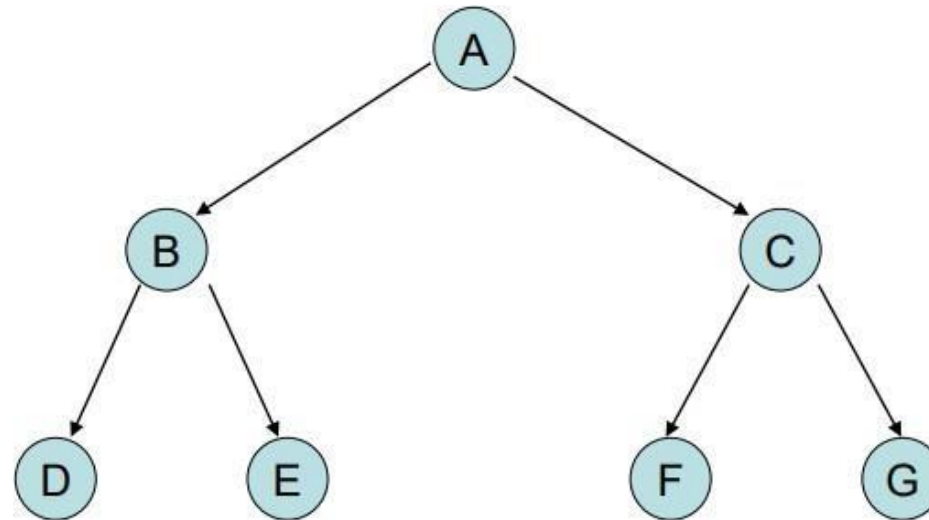


Al finalizar el curso, el estudiante desarrolla un proyecto aplicando estructuras de datos dinámicas en el lenguaje C# con entorno gráfico; demostrando capacidad de análisis, pensamiento lógico y buenas prácticas de programación e implementa funciones definidas de usuarios y sus diferentes tipos

Temario

- **Árboles:**
- Generalidades. Árboles binarios. Operaciones: Raíz, hoja, tallo, recorrido inorden, postorden, preorden

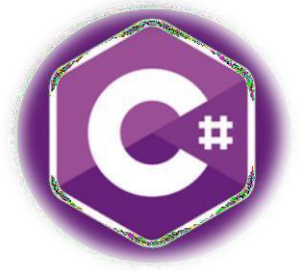
ÁRBOL BINARIO



Preorden: A, B, D, E, C, F, G

Inorden: D, B, E, A, F, C, G

Postorden: D, E, B, F, G, C, A



DATOS IMPORTANTES DE LOS ÁRBOLES



Nodos: Se le llama Nodo a cada elemento que contiene un Árbol.

Nodo Raíz: Se refiere al primer nodo de un Árbol, Solo un nodo del Árbol puede ser la Raíz.

Nodo Padre: Se utiliza este termino para llamar a todos aquellos nodos que tiene al menos un hijo.

Nodo Hijo: Los hijos son todos aquellos nodos que tiene un padre.

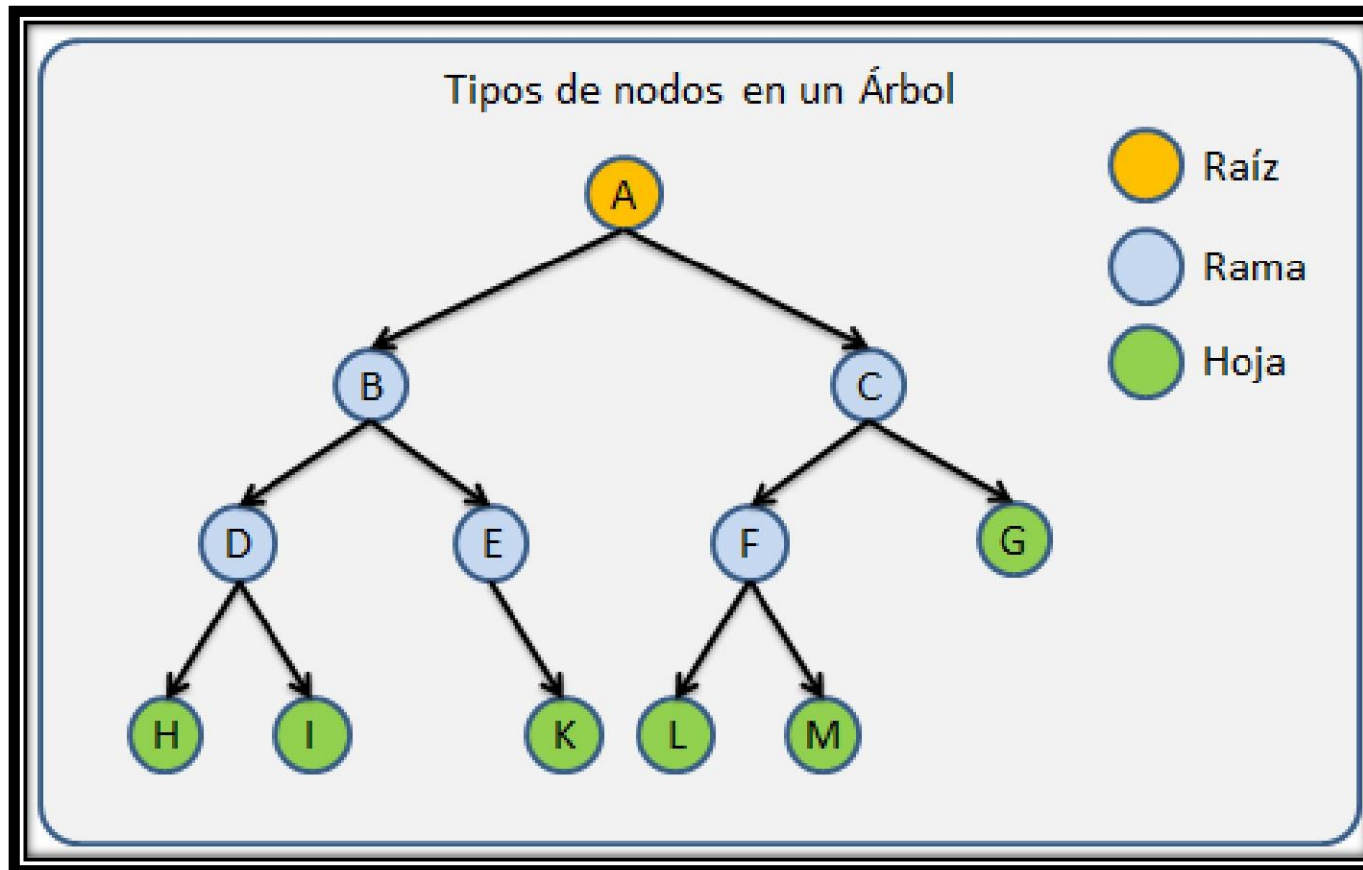
Nodo Hermano: Los nodos hermanos son aquellos nodos que comparte a un mismo padre en común dentro de la estructura.

Nodo Hoja: Son todos aquellos nodos que no tienen hijos, los cuales siempre se encuentran en los extremos de la estructura.

Nodo Rama: Estos son todos aquellos nodos que no son la raíz y que además tiene al menos un hijo.



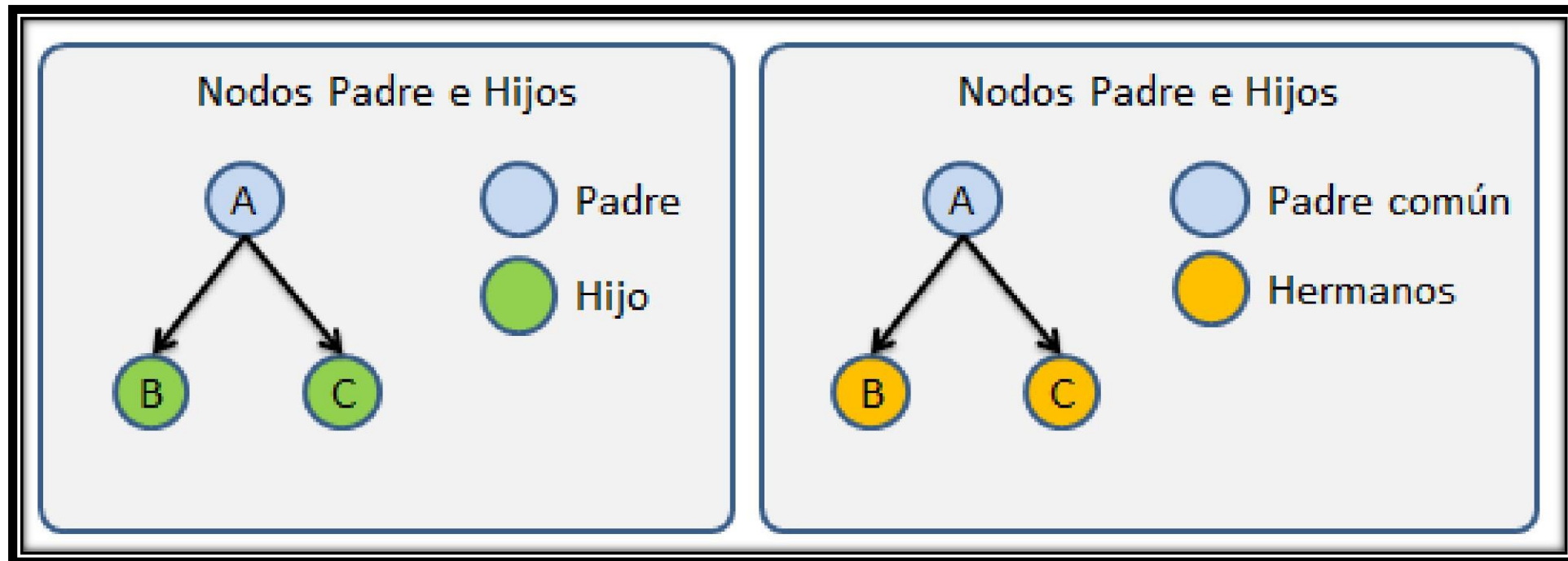
DATOS IMPORTANTES DE LOS ÁRBOLES



La imagen muestra de forma gráfica cuales son los nodos Raíz, Rama, Hoja.



DATOS IMPORTANTES DE LOS ÁRBOLES



La siguiente imagen muestra de forma gráfica los nodos Padre, Hijo y Hermanos.

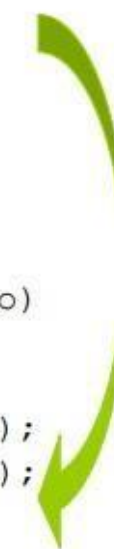



ÁRBOL BINARIO



- Preorder:
 - Visitar nodo
 - Visitar árbol izquierdo
 - Visitar árbol derecho
- Inorder:
 - Visitar árbol izquierdo
 - Visitar nodo
 - Visitar árbol derecho
- Postorder:
 - Visitar árbol izquierdo
 - Visitar árbol derecho
 - Visitar nodo

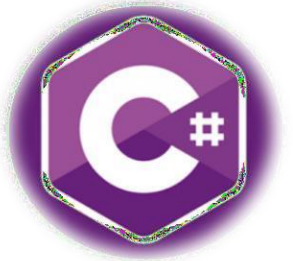
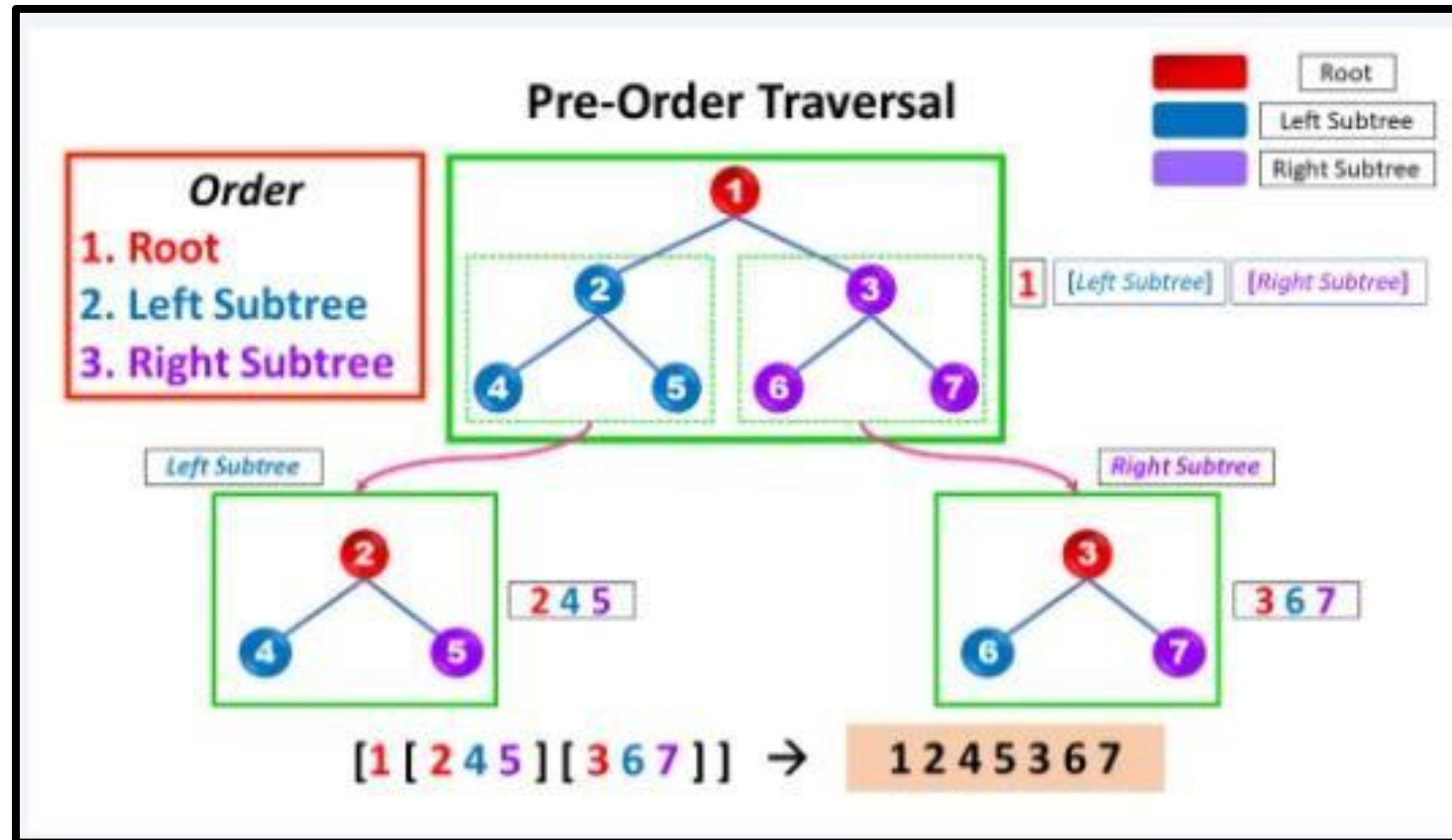
```
void inorder(NodoArbol *nodo) {  
    if (nodo != NULL) {  
        inorder(nodo->izq);  
        visitar(nodo);  
        inorder(nodo->der);  
    }  
}  
  
void postorder(NodoArbol *nodo)  
{  
    if (nodo != NULL) {  
        postorder(nodo->izq);  
        postorder(nodo->der);  
        visitar(nodo);  
    }  
}
```



ÁRBOL BINARIO



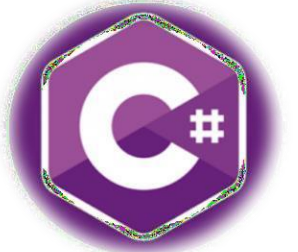
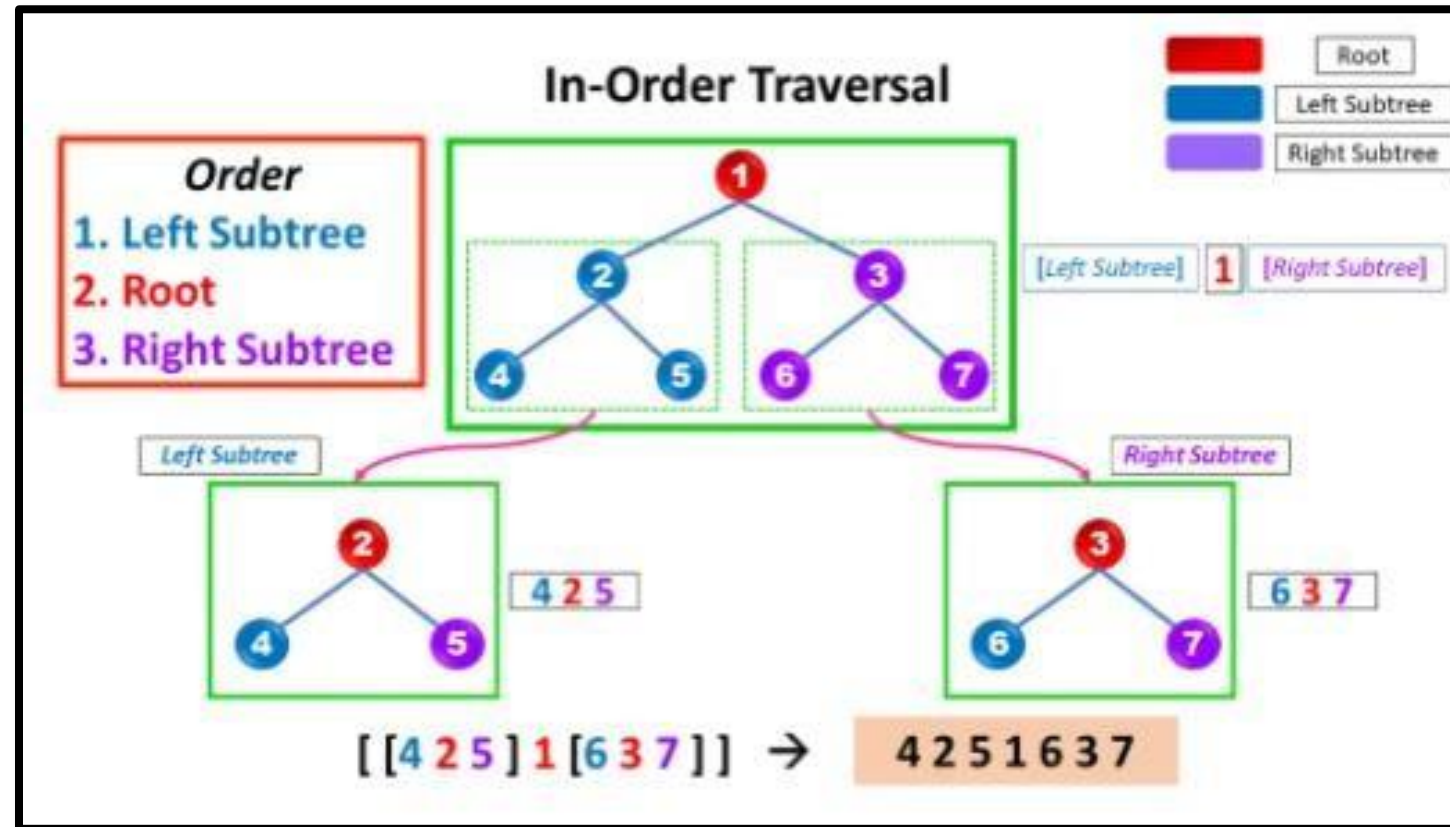
En el recorrido **PreOrder**, se visita primero el **nodo raíz**, luego el **subárbol izquierdo**, y finalmente el **subárbol derecho**.



ÁRBOL BINARIO



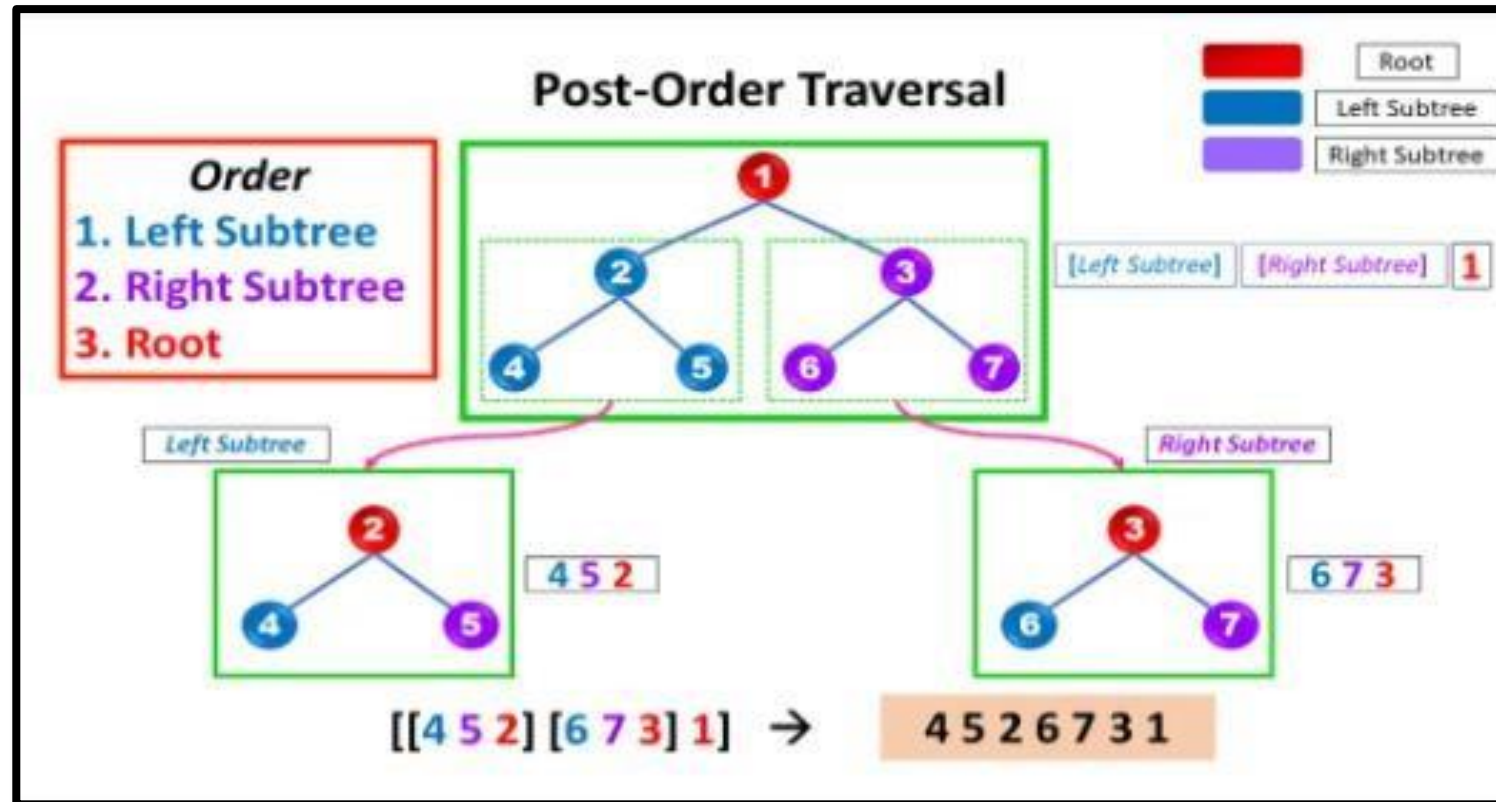
En el recorrido **InOrder**, primero se visita el **subárbol izquierdo**, luego el **nodo raíz** y finalmente el **subárbol derecho**.



ÁRBOL BINARIO



En el recorrido **PostOrder**, se visita primero el **subárbol izquierdo**, luego el **subárbol derecho** y finalmente el **nodo raíz**.

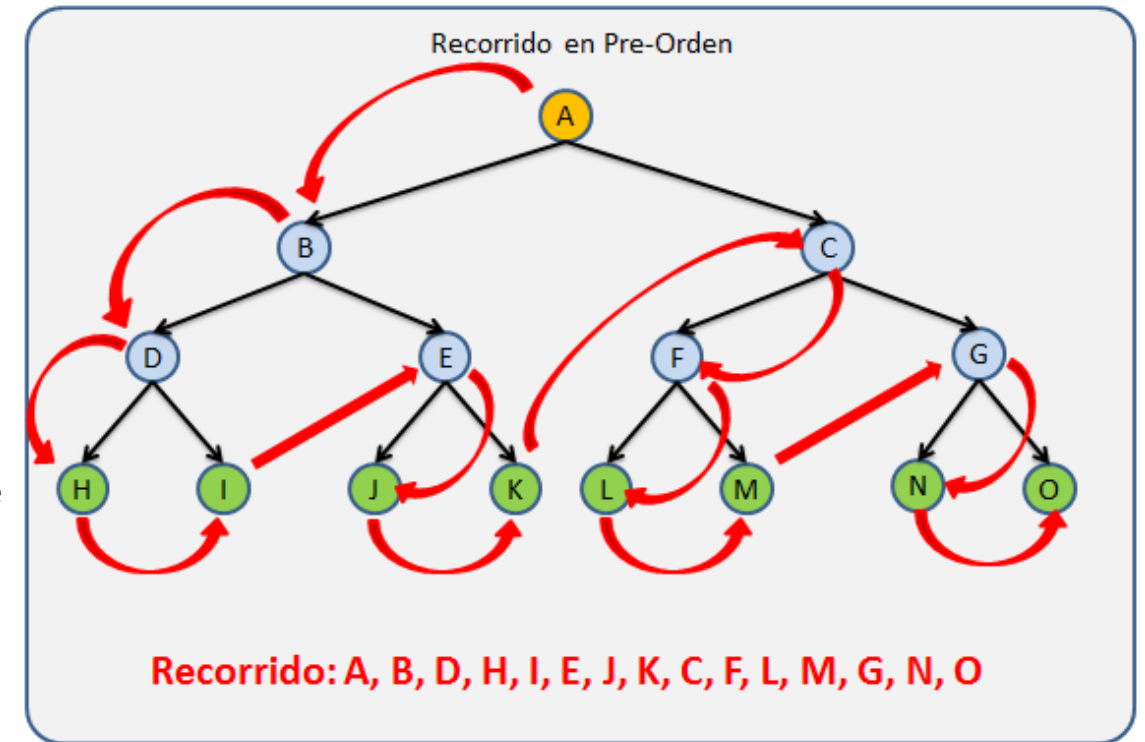


BÚSQUEDA EN PROFUNDIDAD



Recorrido Pre-orden: El recorrido inicia en la Raíz y luego se recorre en pre-orden cada uno de los sub-árboles de izquierda a derecha.

Esta definición puede ser un poco compleja de entender por lo que mejor les dejo la siguiente imagen.



En la imagen podemos ver el orden en que es recorrido el árbol iniciando desde la Raíz.

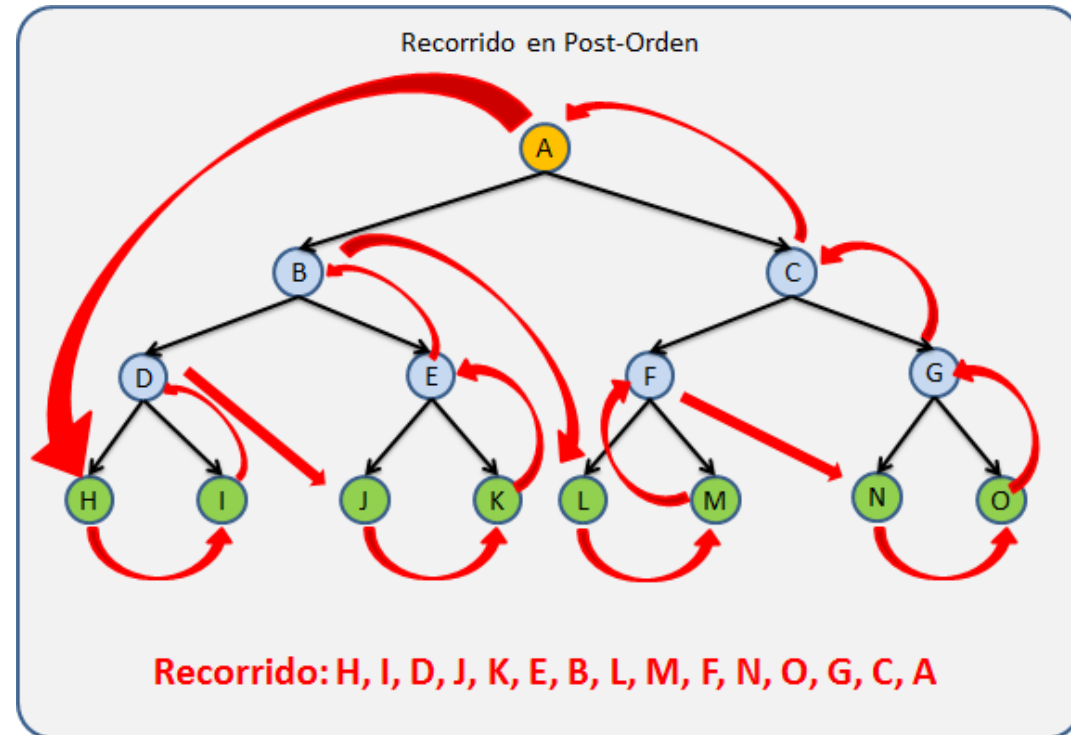


BÚSQUEDA EN PROFUNDIDAD



Recorrido Pos-orden: Se recorre el pos-orden cada uno de los sub-árboles y al final se recorre la raíz.
Para comprender mejor esta definición observemos la siguiente imagen:

En la imagen podemos observar como se realiza el recorrido en Pos-Orden, Sin embargo es importante notar que el primer nodo que se imprime no es la Raíz pues en este recorrido la Raíz de cada Sub-Árbol es procesado al final, ya que toda su descendencia ha sido procesada.

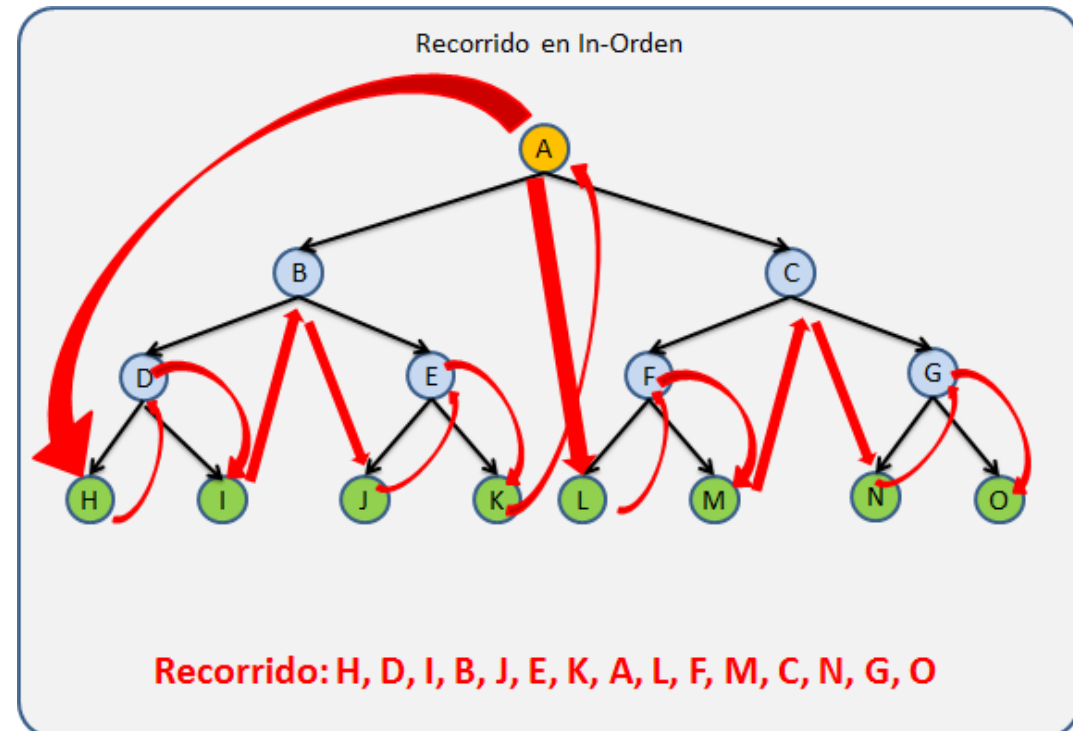


BÚSQUEDA EN PROFUNDIDAD



Recorrido in-orden: Se recorre en in-orden el primer sub-árbol, luego se recorre la raíz y al final se recorre en in-orden los demás sub-árboles:

En la imagen se muestra como es el recorrido In-Orden, Podemos apreciar que la Raíz no es el primero elemento en ser impreso pues este recorrido recorre su rama izquierda, luego la raíz del sub-árbol y luego la ramaderecha.

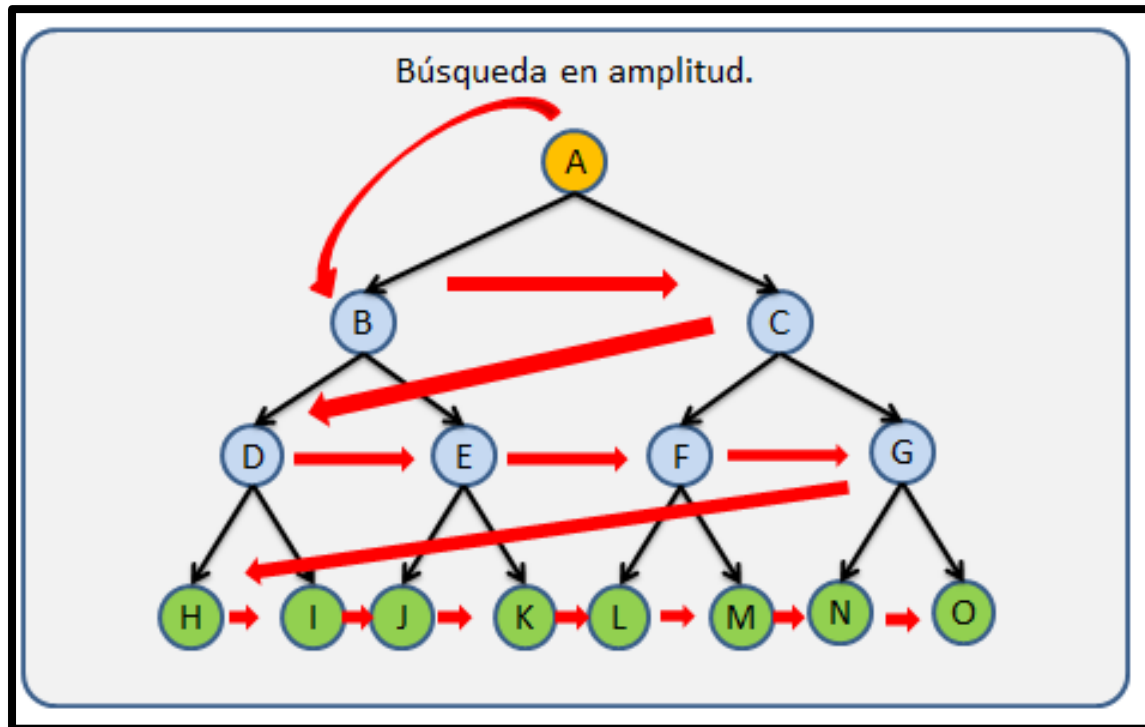


BÚSQUEDA EN AMPLITUD



Se recorre primero la raíz, luego se recorren los demás nodos ordenados por el nivel al que pertenecen en orden de izquierda a derecha.

Este tipo de búsqueda se caracteriza por que la búsqueda se hace nivel por nivel y de izquierda a derecha.



En la imagen se observa como es que un nodo es buscado mediante la búsqueda en profundidad.

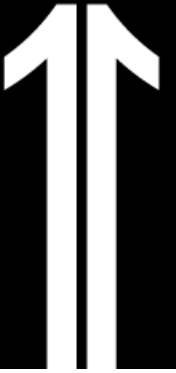




ACTIVIDAD DE CLASE

PROBLEMAS

PROBLEMAS



- Resolver la actividad planteada en la plataforma virtual de aprendizajes

PROBLEMAS

Problema 1- Arbol



1. Crear un menú con Arboles Binarios que permita generar el registro nuevo de nodo con recorrido de pre-orden, post-orden e InOrden y generar una búsqueda del nodo ingresado.

```
Console.WriteLine("| -----|");
Console.WriteLine("|           Arboles Binarios|");
Console.WriteLine("| -----|-----|");
Console.WriteLine("| 1. Registrar un nuevo nodo|");
Console.WriteLine("| 2. Recorrido en pre-orden|");
Console.WriteLine("| 3. Recorrido en post-orden|");
Console.WriteLine("| 4. Recorrido en In-orden|");
Console.WriteLine("| 5. Buscar nodo|");
Console.WriteLine("| -----|-----|");
Console.WriteLine(" Escoja una opción: ");
```

PROBLEMAS

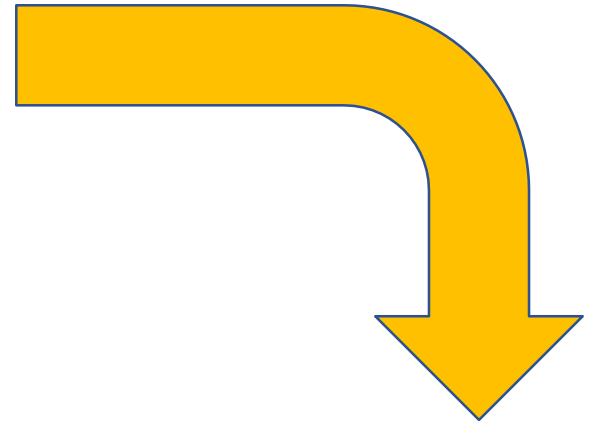
Problema 2- Arbol - Form



Crear un ejercicio con **ARBOL BINARIO** en el lenguaje de programación en c# en forma gráfica con formulario que tenga clase **Árbol** con métodos (**preOrden**, **inOrden**, **postOrden**, **insertar**, **verArbol**, **buscar**, **eliminar**, **esHoja**, **eliminarhoja**, **unhijo**, **doshijos**, **reemplazoderecho**) clase **Libros** con atributos (**código**, **nombre**, **copias**) y **Nodo**. **No se permite get; set; ni List<>**

Desarrollar los siguientes métodos:

- ☐ Ingresar el nuevo Árbol
- ☐ Mostrar en el Árbol
- ☐ Búsqueda por copias
- ☐ Eliminar por copias
- ☐ Ingresar PreOrden, InOrden y PostOrden



PROBLEMAS

Problema 1- Arbol

- ❑1 MenuStrip
- ❑3 TextTBox
- ❑3 Label
- ❑2 DataGridView

| Control | Nombre sugerido | Función |
|---------------------|-----------------|---|
| TextBox | txtCodigo | Ingresar código del libro |
| TextBox | txtNombre | Ingresar nombre del libro |
| TextBox | TxtCopias | Permite ingresar la cantidad de copias del libro. Este campo se usa también para buscar y eliminar nodos |
| DataGridView | dgvLista | Muestra los resultados de los recorridos (PreOrden, InOrden, PostOrden). Cada fila representa un nodo con su código, nombre y copias. |
| DataGridView | dgvArbol | Representa visualmente la estructura del árbol, colocando cada nodo en su nivel correspondiente. Útil para observar la organización jerárquica del árbol. |
| MenuStrip | TsbNuevo | Inserta un nuevo libro en el árbol binario |
| | TsblArbol | Muestra gráficamente la estructura del árbol en el dgvArbol. |
| | TsmBuscar | Busca un nodo por el campo copias |
| | TsbEliminar | Elimina un nodo del árbol por el número de copias. |
| | TsmPreOrden | Realizan y muestran los recorridos del árbol |
| | TsmInOrden | Realizan y muestran los recorridos del árbol |
| | TsmPostOrden | Realizan y muestran los recorridos del árbol |
| 2 Label | | Se utilizan para indicar visualmente al usuario qué información debe ingresar en cada TextBox |



PROBLEMAS

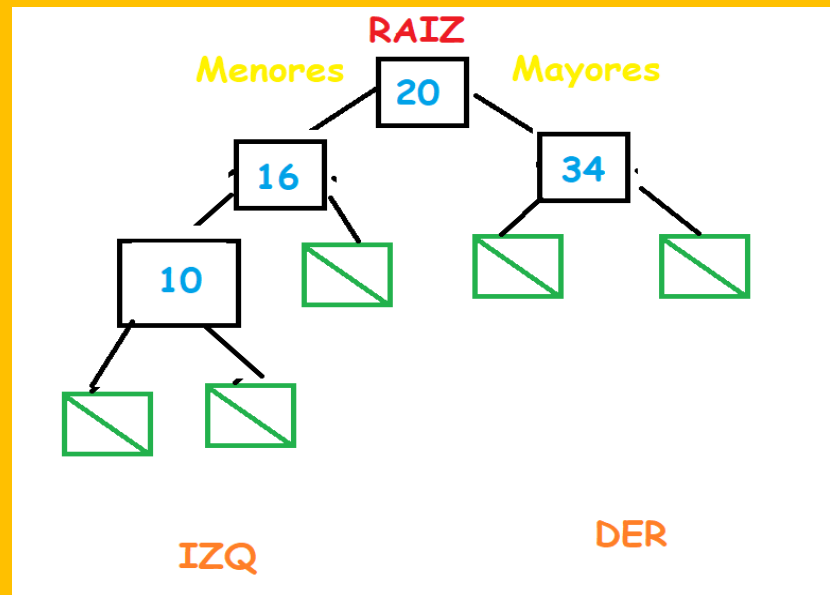
Problema 2- Arbol

- ❑ 1 MenuStrip
- ❑ 3 TextBox
- ❑ 3 Label
- ❑ 2 DataGridView

The screenshot shows a Windows Form titled "Form1" with a standard Windows XP-style title bar. Below the title bar is a menu strip with six items: "Nuevo", "Arbol", "Buscar", "Eliminar", "PreOrden", and "PostOrden". The "Arbol" menu item is currently selected. Below the menu strip, there are three input fields: "Código:" followed by a text box, "Nombre:" followed by a text box, and "Copias:" followed by a text box. Below these input fields, there are two large, empty rectangular areas, likely intended for displaying a tree structure or search results. The form has a white background and a black border.

CONCLUSIONES

Un árbol se puede definir como una **estructura jerárquica y en forma no lineal, aplicada sobre una colección de elementos u objetos llamados nodos.** (Cairó & Guardati, 2006).



CASO:

Cada caso tendrá 2pts

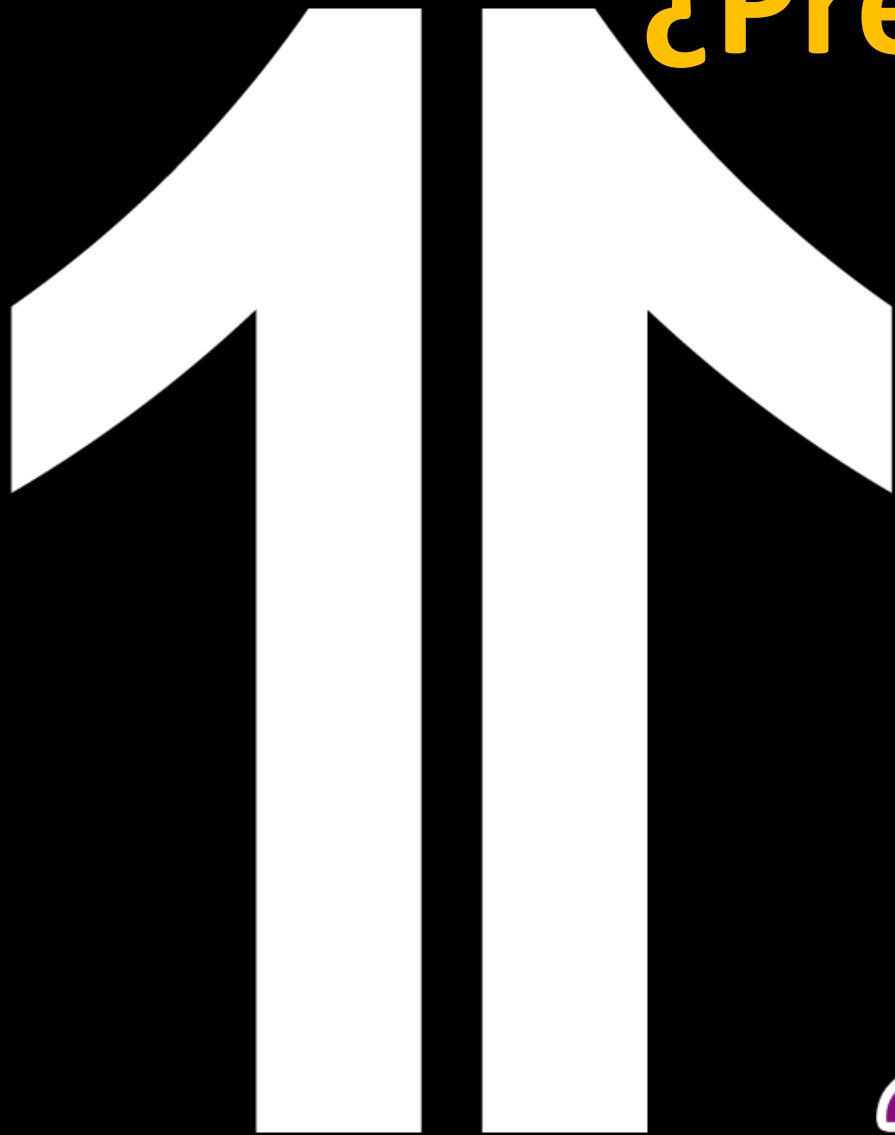
CASO01:

Se solicita una simulación de proceso en un hospital para todas las personas que tiene una mala alimentación en estos tiempos. Se pide un ingreso el número de arreglo de 5 a 20. Con un recorrido del árbol InOrden y PreOrden. Se pide seguir los pasos a continuación

- Ingresar la cantidad de personas
- Recorrido del árbol InOrden
- Recorrido del árbol PreOrden.
- Mostrar el total de b y c.

| Ingreso de personas | <u>InOrden</u> | <u>PreOrden</u> |
|---------------------|----------------|-----------------|
| 4 | 5 | 13 |
| | 6 | 12 |
| | 7 | 10 |
| | 10 | 7 |
| | 11 | 6 |
| | 12 | 5 |
| | 13 | 4 |
| TOTAL | 64 | 57 |

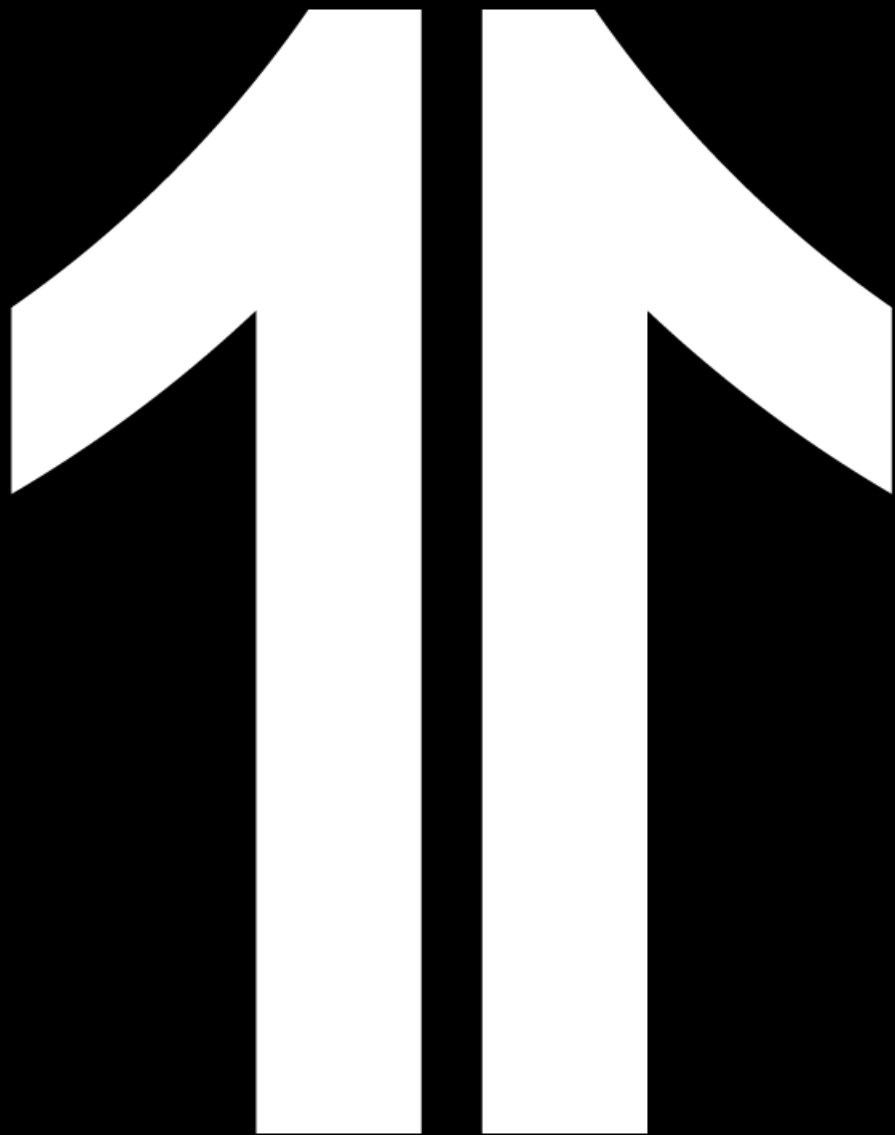
¿Preguntas o Comentarios?





REFERENCIAS BIBLIOGRÁFICAS

- HerbertSchildtFundamentosdeC#3.0
- <https://pdfcoffee.com/fundamentos-de-c-30-4-pdf-free.html>



GRACIAS

