ITL-ISC 1ª. OPORT. ESTRUCTURA DE DATOS NOMBRE :\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

EXAMEN DE COMPETENCIA U-4. Crear soluciones aplicando estructuras no lineales

COMPETENCIA 1.- Conocer los conceptos fundamentales de la estructura de datos: arbol binario.

1. (40 ptos.) Dibuja el árbol binario para la secuencia: 1, 64, -33, -22, 33, 44, 90, -44, -28, 1

2. (60 ptos.) Escribe en preorden, inorden y posorden el árbol generado en 1.

COMPETENCIA 2.- Implementacion de las clases Arbol, Nodo. Atributos, propiedades y metodos fundamentales: Constructor, inicializacion, Agregar, Buscar, Recorrer perorden, Inorden, Posorden, Eliminar..

1. (20 ptos.) Escribe la clase Arbol, sus atributos, solamente. Escribe el constructor de la clase Arbol.

2. (60 ptos.) Escribe el método Agregar() en la clase Arbol. NOTA: la sobrecarga suponga que ya está hecha.

3. (20 ptos.) Escribe la definición de la clase Nodo, sus atributos y propiedades.

COMPETENCIA 3.- Definicion de otras propiedades y metodos en la clase Arbol.

1. (100 ptos.) Escribe un método en la clase Arbol, que retorne una referencia al mayor elemento en el árbol binario. Si no existe retorna un null. NOTA: la sobrecarga suponga que ya está hecha.

COMPETENCIA 4.- Solucion de problemas usando arboles binarios.

1. (100 ptos.) Se tienen 2 arboles binarios arbol1 y arbol2. La lectura y visualización de loa arboles ya está codificada. Agrega un botón que visualice si el árbol seleccionado está balanceado o no lo está. Un árbol está balanceado, si el número de nodos del subárbol derecho y el número de nodos en el subárbol izquierdo difieren en 1 o en 0.

ITL-ISC 1ª. OPORT. ESTRUCTURA DE DATOS NOMBRE :\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

EXAMEN DE COMPETENCIA U-4. Crear soluciones aplicando estructuras no lineales

COMPETENCIA 1.- Conocer los conceptos fundamentales de la estructura de datos: arbol binario.

1. (40 ptos.) Dibuja el árbol binario para la secuencia: 1, 64, -33, -22, 33, 44, 90, -44, -28, 1

2. (60 ptos.) Escribe en preorden, inorden y posorden el árbol generado en 1.

COMPETENCIA 2.- Implementacion de las clases Arbol, Nodo. Atributos, propiedades y metodos fundamentales: Constructor, inicializacion, Agregar, Buscar, Recorrer perorden, Inorden, Posorden, Eliminar..

1. (20 ptos.) Escribe la clase Arbol, sus atributos, solamente. Escribe el constructor de la clase Arbol.

2. (60 ptos.) Escribe el método Agregar() en la clase Arbol. NOTA: la sobrecarga suponga que ya está hecha.

3. (20 ptos.) Escribe la definición de la clase Nodo, sus atributos y propiedades.

COMPETENCIA 3.- Definicion de otras propiedades y metodos en la clase Arbol.

1. (100 ptos.) Escribe un método en la clase Arbol, que retorne una referencia al mayor elemento en el árbol binario. Si no existe retorna un null. NOTA: la sobrecarga suponga que ya está hecha.

COMPETENCIA 4.- Solucion de problemas usando arboles binarios.

1. (100 ptos.) Se tienen 2 arboles binarios arbol1 y arbol2. La lectura y visualización de loa arboles ya está codificada. Agrega un botón que visualice si el árbol seleccionado está balanceado o no lo está. Un árbol está balanceado, si el número de nodos del subárbol derecho y el número de nodos en el subárbol izquierdo difieren en 1 o en 0.