

Nombre: _____

R.U.N: _____

Estructura de Datos

Evaluación Integradora

Docentes: Aaron Pavez – Gabriela Aguayo

26 de Marzo 2020

Instrucciones:

1. Lea cuidadosamente cada pregunta antes de responder.
2. Escriba su nombre en cada hoja de respuesta.
3. Utilice solamente las hojas de respuestas.
4. Para cualquier implementación de código, utilice el lenguaje de programación C.

Importante

Para cada ejercicio que deba implementar código, se pedirá un archivo “.c”. Se debe incluir en el archivo la función y un ejemplo de ejecución en el Main.

Es posible crear funciones de ayuda como crearNodo, agregarNodo e imprimirNodo, etc. (Dichas funciones no serán evaluadas)

Si el ejercicio tiene errores de sintaxis o se cae en tiempo de ejecución el puntaje será cero.

Nombre: _____

R.U.N: _____

Punteros

(2 Pts)

Se pide desarrollar una función la cual deberá recibir 2 arreglos de punteros por parámetro. Retorne un nuevo arreglo que contenga (**Sin repetir**) todos los números que se encuentran disponibles tanto en el arreglo 1 como en el arreglo 2 a la vez. Imprima el resultado en el Main.

Utilice el siguiente prototipo de función. (No es posible modificar)

```
int* generarArrayComun(int *a, int an, int *b, int bn,  
int *nNuevo)
```

Observación 1: Puede crear funciones de ayuda para la función principal.

Observación 2: **an** y **bn** corresponde a la dimensión respectiva del arreglo a y b.

Observación 3: En nNuevo se debe guardar la nueva dimensión del arreglo que se debe retornar.

Observación 4: Está estrictamente prohibido el uso de arreglos con "[]". Todo debe ser con aritmética de punteros.

Ejemplo:

A: 5 1 3 4 2 3 1 4 2

B: 3 0 1 9 1 10

Retorno: 3 1

Recuerde que su respuesta debe ser entregada en un archivo ".c". Incluya un ejemplo de ejecución en el archivo.

Nombre: _____

R.U.N: _____

Listas Enlazadas

(2 Pts)

Utilice la siguiente estructura para resolver el problema (No es posible modificar)

```
typedef struct nodo
{
    int valor;
    struct nodo *next;
} * Nodo;
```

Implemente una función que reciba dos listas e indique a través de un 0(false) o 1(true) si la lista B está contenida dentro de la lista A.

Observación 1: Se dice que está contenida cuando todos los elementos de B se encuentran en A.

Observación 2: Puede crear funciones de ayuda para la función principal.

Utilice el siguiente prototipo de función. No es posible modificar.

```
int esContenido(Nodo listaA, Nodo listaB)
```

Ejemplo 1:

Lista A: 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> NULL

Lista B: 3 -> 4 -> 5 -> NULL

Resultado: 1

Ejemplo 2:

Lista A: 1-> 2 -> 3 -> 4 -> 5 -> NULL

Lista B: 2 -> 3 -> 5 -> NULL

Resultado: 0

Ejemplo 3:

Lista A: NULL

Lista B: NULL

Resultado: 1

Recuerde que su respuesta debe ser entregada en un archivo “.c”. Incluya un ejemplo de ejecución en el archivo.

Nombre: _____

R.U.N: _____

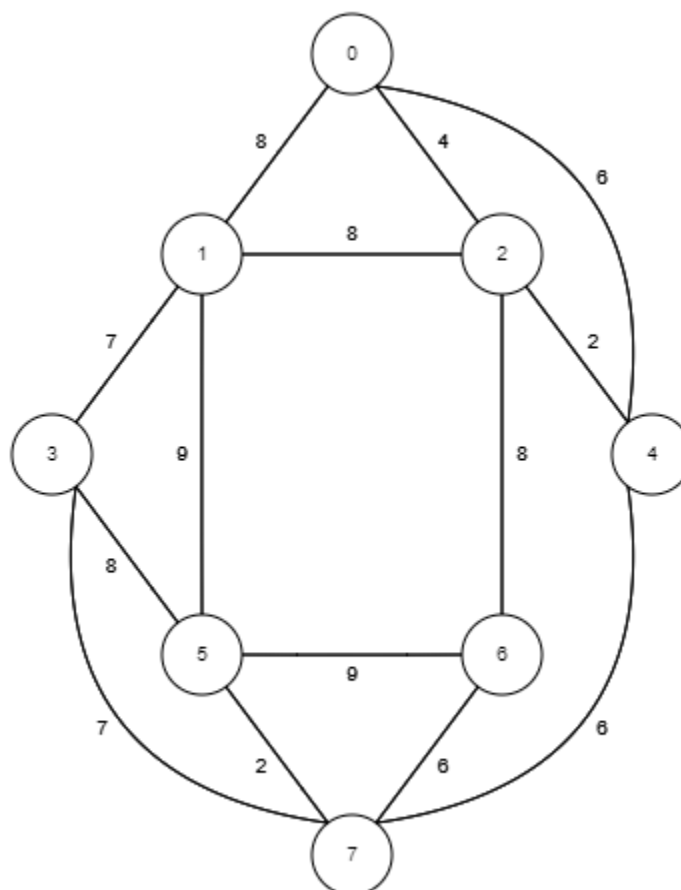
Dijkstra

(1Pts)

Utilizando el algoritmo Dijkstra, complete la siguiente tabla de valores.

Indicar la ruta de **Menor costo** desde el Vértice **1** a cualquier otro Nodo.

Origen	Destino	Camino	Costo
1	0		
1	2		
1	3		
1	4		
1	5		
1	6		
1	7		



Nombre: _____

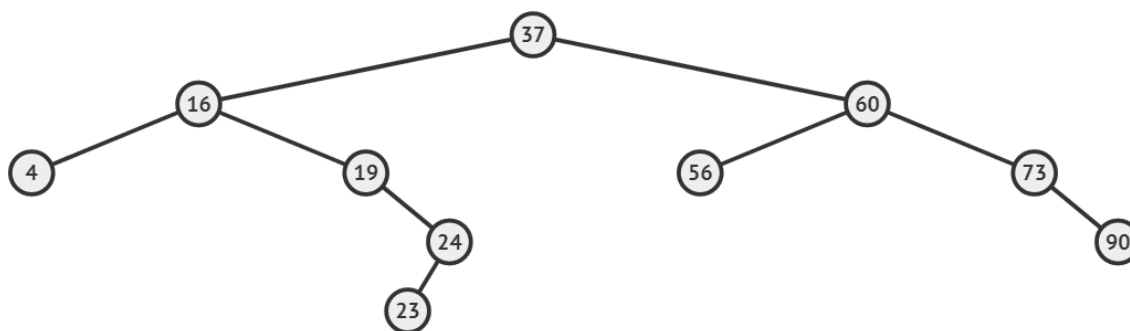
R.U.N: _____

AVL

(1 Pts)

Balancear el siguiente árbol.

Observación: Para cada acción que realice en el árbol deberá dibujar su nuevo estado. **Una rotación doble se deberá representar en dos estados.** (La cantidad de cuadrantes no representa la cantidad de pasos a realizar). (0.5 Pts)



Nombre: _____

R.U.N: _____

Balancear el siguiente árbol.

Observación: Para cada acción que realice en el árbol deberá dibujar su nuevo estado. Una rotación doble se deberá representar en dos estados. (La cantidad de cuadrantes no representa la cantidad de pasos a realizar). (0.5 Pts)

