

Serán considerados al calificar este examen la eficiencia de las soluciones y la utilización adecuada de las características del lenguaje C y de la programación estructurada. LOS RESULTADOS PRESENTADOS DEBEN SER PROPIOS, ORIGINALES y RESUELTOS DE MODO INDIVIDUAL.

*TIEMPO DE RESOLUCION y SUBIDA de ARCHIVO/S: 130'. **No se aceptarán entregas por otro medio***

Para aprobar es necesario obtener al menos 5 puntos en este examen y al menos 4,25 deben obtenerse entre los ejercicios 2, 3 y 4.

En todos los ejercicios que corresponda, mostrar las invocaciones (incluyendo su contexto: declaraciones e inicializaciones) de las soluciones desarrolladas.

Subir archivos .c y/o .doc. El tiempo máximo para resolver este examen es 135'

() Cuando este examen está aprobado, la nota FINAL se obtiene así: $CURSADA * 0.3 + TOTALIZADOR * 0.7$*

(1,5p) Ej 1.- Indicar Verdadero o Falso para cada una de las siguientes afirmaciones. Justificar o ejemplificar la respuesta dada (toda respuesta sin justificar o ejemplificar tendrá puntaje cero)

a) Una cola circular, podría implementarse efectuando corrimientos, lo que implicaría modificar tanto el algoritmo de inserción como el de eliminación.

b) Si luego de ejecutar el algoritmo de Floyd, la matriz final A contiene infinitos significa que el grafo es no-conexo

(3,5p) Ej 2.- Suponer una lista doblemente enlazada LDE en la que cada elemento contiene un número N entero positivo (no se repiten) y una sublista con valores enteros que se supone son divisores de N. Se pide, definir tipos e implementar:

i) Dado K, verificar si todos los elementos de la sublista del nodo que contiene K son sus divisores. Eliminar los que no lo sean. Informar la situación.

ii) Dado X, generar un archivo de texto DIVI.TXT con los valores de la lista que sean pares y tienen exactamente X divisores. Colocar un valor por línea.

(2,5p) Ej 3.- (Utilizar TDA N.Ario) Se tiene un árbol N-Ario AN de números enteros, determinar en qué nivel se encuentra la primera aparición de K en AN según el recorrido inorden del árbol (K dato de entrada)

(2,5p) Ej 4.- Dado un digrafo con aristas ponderadas implementado en una lista de adyacencia, y una pila P en la que cada elemento contiene un *peso* y un *vértice*. Se pide definir tipos y desarrollar una solución recursiva sobre la Pila que elimine de la misma aquellos *vértices* cuyo *peso* sea menor que el costo total de las aristas que se vinculan con él que no son bucles.