



## TOTALIZADOR PROGRAMACION II

*Serán considerados al calificar este examen la eficiencia de las soluciones y del uso de las características del lenguaje C y de la programación estructurada.*

*Para aprobar es necesario obtener al menos 5 puntos*

**Ej 1) a) (Utilizar TDA Cola)** Se tiene una matriz  $T$  de  $N \times N$  naturales que representa un AAM construido sobre un grafo. Desarrollar una solución recursiva sobre la matriz que almacene en la cola los vértices y su grado en el AAM, sólo para aquellos vértices que grado  $> 1$  en el AAM, retornar además el costo del AAM.

- Si la solución tiene algún ciclo el puntaje será 0
- Cada elemento de la cola será un registro con dos campos: *vértice* y *grado*.
- Mostrar la invocación de la solución desarrollada

**b)** Suponer que la Cola está implementada en memoria estática, definir el tipo de la Cola y desarrollar el operador `PoneC` utilizado en la solución del inciso **a)**.

**Ej 2)** Una lista simplemente enlazada se generó luego de haber analizado un texto, en cada nodo:

- ✓ Letra (Ordenada, pueden no estar todas las letras del alfabeto)
- ✓ SubLista de palabras que se hallaron en el texto
  - Palabra (Ordenada)
  - Apariciones ( $>0$ )

Se pide, definir tipos y desarrollar subprogramas para:

**a)** Dada una palabra que fue incorrectamente incluida en lista, eliminarla.

**b)** Generar una lista circular que contenga en cada nodo, sólo las palabras que comienzan con vocal y aparecieron más de 5 veces con el siguiente contenido en cada nodo:

- ✓ Palabra (Ordenada, comienza con vocal)
- ✓ Apariciones ( $>5$ )

**Ej 3) (Utilizar TDA N.Ario)** Dado un árbol  $n$ -ario de enteros, determinar mediante una función int cuántas hojas en el nivel  $K$  contienen un 0.

- $K$  es dato
- Mostrar la invocación de la función desarrollada