

GUÍA DE TRABAJOS PRÁCTICOS

CURSADA 2024 – 1ER CUATRIMESTRE

Práctica II: Listas

En cada caso se debe implementar el TAD descrito y una aplicación que, utilizando el tipo abstracto de dato, permita al usuario el ingreso de una o más instancias (según sea necesario), la aplicación de cualquiera de las operaciones y luego la obtención del resultado.

1. Construir la TAD de listas correspondientes a las implementaciones de “Arreglos”, “Punteros” y “Cursores” teniendo en cuenta:
 - a. Usar las “.h” definidos por la cátedra. Descargarlos de la plataforma.
 - b. Se las debe implementar cada una en un “.c” diferente. Es decir uno para cada tipo de implementación.
 - c. Se las debe probar y testear de forma tal que se pueda asegurar que la TAD funciona correctamente.

Importante !!!

Los siguientes ejercicios deben ser implementados y resueltos en forma genérica, esto significa que se debería poder referenciar cualquiera de las implementaciones de lista (arreglo, puntero o cursor) y los mismos deben seguir en funcionamiento sin problemas.

Para todos los casos se debe tener una interfaz para que se puedan cargar valores manualmente para corroborar los resultados de cada ejercicio.

2. Dada una 2 listas (L1 y L2) cargadas con valores al azar (en la clave) realizar los siguientes ejercicios:
 - a. Que retorne una lista con los valores de L1 que no están en L2.
 - b. Que retorne una lista con los valores de L2 que no están en L1.
 - c. Que retorne una lista con todos los valores comunes a ambas listas (L1 y L2).
 - d. Que retorne los promedios de ambas listas.
 - e. Que retorne el valor máximo de ambas listas y su posición ordinal.
3. Dadas 2 listas (L1 y L2) determinar si L2 es múltiplo de L1. Se considera múltiplo si cada elemento de L2 se divide en forma exacta por el valor L1 de la misma posición. Usar la clave como campo de datos solamente.

Ejemplo: si $L1 = (2, 5, 7, 3)$ y $L2 = (8, 20, 28, 12)$ entonces L2 es múltiplo por L1 porque cada posición de L2 se divide por el valor de L1 de la misma posición en forma exacta (sin decimales). Si el resultado de la división retorna el mismo valor para cada posición se dice que “L2” es múltiplo de “L1” por un “escalar”. Para este caso “4” es el escalar de L1. El algoritmo debe contemplar esta situación.

4. Dadas dos listas L1 y L2, se pide compararlas siempre en el sentido $L1 \rightarrow L2$. Por lo tanto puede suceder que: $L1 > L2$, $L1 = L2$ o $L1 < L2$. La forma de obtener la comparación es por la clave, posición a posición, donde si L1 tiene más cantidad de claves mayores que L2 se considera $L1 > L2$, por el contrario se considera $L1 < L2$, o de lo contrario L1 será igual a L2. Determinar la complejidad algorítmica de la solución empleada.

5. Generar un algoritmo que guarde en una lista enlazada los coeficientes de un polinomio. Realizar una función que dada la lista de coeficientes y un cierto valor de “X” nos devuelva el valor del polinomio para ese punto.

Luego realizar un proceso que dado un rango de valores de “X” y un valor de intervalo “I”, retorne una lista con los valores de “Y” o “F(x)”.

Ejemplo: si el polinomio es $F(x) = 2x + 1$. Se pide retornar los valores de $F(x)$ entre los “X” -1 y 1 de a “0,5”. Es decir se deberían retornar los valores de $F(-1)$, $F(-0,5)$, $F(0)$, $F(0,5)$ y $F(1)$.

6. Generar un algoritmo que determine si una lista es sublista de otra. Se considera que es una sublista si todos los valores de la segunda se encuentran dentro de la primera sin importar el orden o posición de cada elemento. La comparación es solo por la clave. Se pide además determinar la complejidad algorítmica de la solución.

Ejemplo: si “L1” contiene los elementos (7, 3, 4, 5, 9, 2) y “L2” contiene los elementos (4, 7, 2) se dice que “L2” es sublista de “L1”.

IMPORTANTE !!!

- Los prácticos podrán ser resueltos en grupos de 4 **(cuatro)** a 7 **(siete)** alumnos. Condición indispensable para que los TPs sean evaluados.
- Todos los prácticos son obligatorios.
- Los prácticos deben ser entregados en tiempo y forma en las fechas que los docentes estipulen (serán determinadas según la cursada y avisadas mediante la plataforma virtual).
- Los prácticos NO se recuperan por lo que si son entregados en condiciones incorrectas de funcionamiento, fuera de término o fuera de formalidad (abajo descripto) se consideran desaprobados.
- Se deben presentar todos los ejercicios del tp. Se deben tener como mínimo un 60% de los ejercicios correctos para aprobar la práctica, con 2 de los 3 últimos ejercicios funcionando correctamente. Se evalúan tanto los errores de lógica como los de ingreso de datos. Deben controlar los ingresos de datos, tales como blancos, ingreso de números correctos, etc.
- Los prácticos deben presentarse con la formalidad:
 - Carátula Impresa o PDF por mail con: Fecha de Entrega, Integrantes de Grupo, Título del Práctico a Entregar.
 - Se tomará asistencia en el momento de entrega del TP.
 - El código fuente debe estar disponible en GIT HUB, con acceso a los docentes.
 - Si se entregan los (.h) se deben respetar.
 - Los ejercicios van a ser probados con cualquiera de las implementaciones de listas (arreglos, apuntadores y cursores).

FECHA DE ENTREGA: 17/04/2024 (17 hs).

El lenguaje de programación será “C”. (Usar el foro para consultas sobre el TP)



Profesor a Cargo de la Asignatura: Mario Perello.

Trabajos Prácticos a Cargo de: Claudia Reinaudi, José Racker, Pablo Chale y Mariano Goldman.