

Práctica 8 – Listas

Contenidos

- **Listas simples**
- **Multiestructuras**

Consultar bibliografía y apuntes de clase para contestar las siguientes preguntas:

- ¿Qué es una lista?
- Armar un ejemplo que ponga y saque datos de una lista.
- Pensar varios ejemplos de la vida real en que el comportamiento sea el de una lista.
- ¿Qué operaciones de acceso se usan para almacenar y acceder a elementos individuales de datos?
- Compare pilas y colas con listas. ¿Qué diferencias o similitudes encuentra?
- ¿Qué diferencia hay entre una lista y un arreglo?

Utilizando listas resolver los siguientes ejercicios.

Ejercicio 1

Se tiene una lista cuyos nodos almacenan:

- Nombre de Ciudad
- Cantidad de mujeres
- Cantidad de varones

Se pide recorrer la lista y calcular e informar:

- Cuántas ciudades tienen mayor porcentaje de hombres
- Nombre de la ciudad con mayor población
- Porcentaje de ciudades donde la cantidad de mujeres supera el promedio de mujeres de todas las ciudades
- Indicar si la lista está ordenada alfabéticamente

Ejercicio 2

Se tiene una cola que almacena palabras, se pide escribirlas ordenadas alfabéticamente utilizando una lista como estructura auxiliar. Las palabras de la cola deben quedar en su orden original.

Ejercicio 3

Dada una lista de enteros, generar dos listas de salida: una con los impares, ordenada en forma descendente, y la otra con los pares, ordenada en forma ascendente. La lista original está ordenada ascendentemente. (Se debe utilizar los nodos de la lista original y establecer nuevos enlaces; no crear nuevos nodos)

Ejercicio 4

A partir de dos listas de números ordenados en forma ascendente, mezclarlas generando una única lista con orden descendente. (Se debe utilizar los nodos de las listas originales y establecer nuevos enlaces; no crear nuevos nodos)

Ejercicio 5

Ídem Ejercicio 4, pero la lista resultante debe quedar en orden ascendente.

Ejercicio 6

Dada una lista ordenada de números y un arreglo de N números ordenados, incorporar los números del arreglo a la lista respetando el orden de la misma.

Ejercicio 7

Dado un archivo que representa un polinomio (cada registro tiene un campo exponente, y otro coeficiente y no viene ordenado), generar un archivo de salida que represente el mismo polinomio derivado y ordenado.

Ejercicio 8

Dada una lista de caracteres desordenados, eliminar de la lista todas las repeticiones de un caracter C recibido como parámetro (debe quedar uno solo en la lista).

Ejercicio 9

Dada una lista de números ordenada, eliminar las repeticiones sin utilizar una estructura auxiliar.

Ejercicio 10

Dada una lista de números desordenada, eliminar las repeticiones sin utilizar una estructura auxiliar.

Ejercicio 11

Dado un archivo binario secuencial con la siguiente estructura de registro:

- Matrícula (cadena de 6)
- Nombre (cadena de 20)
- Notas arreglo de 1..40 de enteros

Desarrollar un algoritmo que saque de la lista a los alumnos recibidos (las 40 notas mayores a 4). La información resultante deberá almacenarse en un archivo binario de salida. Utilizar una lista como estructura auxiliar, ya que el archivo de salida deberá estar ordenado por matrícula (el de entrada no tiene ninguna clase de ordenamiento)

Ejercicio 12

Dada una lista que contiene una cola en cada nodo, se desea generar una pila de salida que contenga el elemento máximo de cada cola de la lista. Es decir, la pila de salida deberá tener la misma cantidad de elementos que la lista.

Ejercicio 13

Se tiene una lista simplemente enlazada, cada nodo contiene una pila (estática) de letras no vacía.

A partir de una palabra, se pide recorrer la lista verificando si en el tope de cada pila se encuentra una letra de la palabra, en dicho caso se remueve y se sigue evaluando el tope hasta que se vacía la pila o la letra en la cima no pertenece a la palabra.

En el caso que la pila quedara vacía, eliminar el nodo de la lista.

El proceso finaliza al terminar el recorrido de la lista o cuando se han detectado y eliminado todas las letras de la palabra. Implementar la solución como una función void que tenga como parámetros la palabra y la lista, e informe si pudieron encontrarse todas las letras.

Ejercicio 14

Se tiene una lista simple, cada nodo almacena un intervalo real $[A,B]$ y una sublista de números reales, se pide:

- a. Verificar si todos los nodos tienen al menos un número dentro del intervalo.
- b. Calcular cuál es el intervalo con máxima cantidad de números (de su sublista)
- c. Para un intervalo dado, eliminar todos los números de su sublista que no estén incluidos.
- d. Idem c pero si queda vacía, eliminar el nodo.
- e. Dado un número real, buscar el primer intervalo al cual pertenezca e inserta por la cabeza (al principio). No hacerlo si ya hay otro igual.

¿si la sublista está ordenada en forma ascendente, cuáles son los ítems afectados y cómo cambia el código desarrollado?

Ejercicio 15

Dado el siguiente archivo:

ARCHIVO-EMPLEADOS

.Legajo-Empleado	(cadena de 4)
.Nombre	(cadena de 30)
.Sueldo_Basico	entero
.Seccion	(cadena de 10)

Y considerando que se trata de la nómina de empleados de una empresa, hacer un programa, que permita realizar el alta o la baja de dicho empleado (según el tipo de movimiento a realizar). El archivo original no está ordenado, y que para dar de alta o baja a un empleado, la sección es uno de los datos a solicitar. Utilizar la estructura de lista (o listas) que considere más conveniente para que el acceso sea lo más rápido posible. Una vez efectuadas todas las altas y bajas del período genere un nuevo archivo secuencial de salida con los datos actualizados y ordenados por sección.

Sugerencia: cada nodo de la lista auxiliar contendrá sección y sublista de empleados de dicha sección.

Notar que puede agregarse un empleado de una nueva sección, que no figura en el archivo.

Ejercicio 16

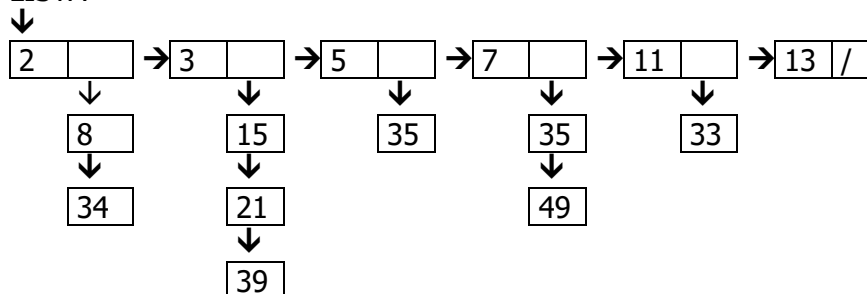
En una lista cada nodo almacena:

- un número primo (en orden ascendente y no repetidos)
- una sublista de múltiplos.

Por otro lado una pila contiene números enteros positivos, no primos. Se pide vaciar la pila ubicando cada múltiplo en la sublista correspondiente al primo que lo divida, siempre y cuando no se encuentre ya almacenado en la misma, en cuyo caso se proseguirá la búsqueda de otro divisor para ubicarlo en su sublista. De no encontrar ubicación posible se insertará en una cola.

Ejemplo:

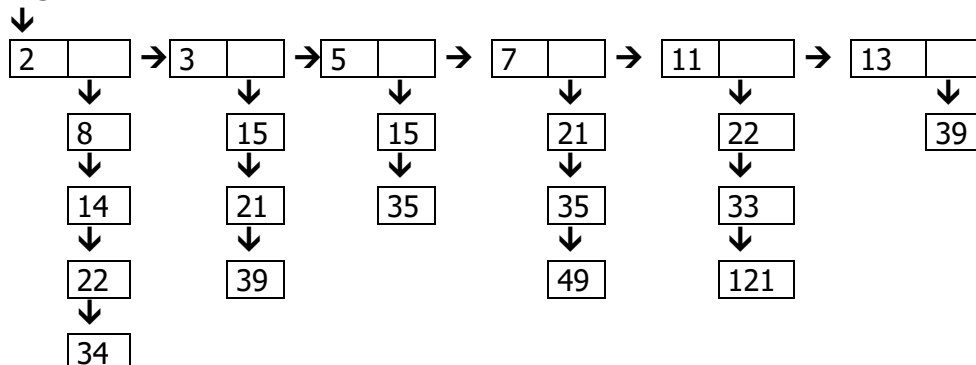
LISTA



PILA

15	22	21	14	49 *	22	121	35 *	39
----	----	----	----	------	----	-----	------	----

LISTA



Los números que en la pila están marcados con un asterisco irán a la cola por no encontrar ubicación en la sublista.

Ejercicio 17

Se tiene una PILA con los siguientes datos de los deportistas intervinientes en una competencia:

- ✓ Nombre y Apellido
- ✓ Código de Deporte (cadena de 6)
- ✓ Tiempo (*en segundos*)

y una lista dónde figura, para cada deporte:

- ✓ Código de Deporte (cadena de 6, *no ordenado*)
- ✓ Tiempo Promedio (*en segundos*)
- ✓ Cantidad de Deportistas
- ✓ Sublista de Deportistas (puede estar vacía)
 - ✓ Nombre y Apellido
 - ✓ Tiempo (*en segundos*)

Se pide sacar de la Pila los datos de los deportistas e insertarlos en la Lista si su tiempo es menor al promedio del deporte correspondiente. La lista tiene un nodo de cada deporte, si no hay deportistas, el primero se inserta directamente. Recalcular el promedio en caso de inserción.