

Apellido y nombre: \_\_\_\_\_ Legajo: \_\_\_\_\_ Curso con Prof: \_\_\_\_\_

Cantidad de hojas entregadas: \_\_\_\_\_ Nota: \_\_\_\_\_ Evaluó Prof: \_\_\_\_\_

- Si luego de la lectura del examen, durante la resolución tiene alguna duda, escriba hipótesis de trabajo, las cuales también serán evaluadas.
- Los puntos que solicitan codificación puede ser respondidos en C, C++, o en Pascal, pero debe indicar el lenguaje utilizado.
- En C y C++ prototipo refiere a la declaración de la función, es decir tipo de dato retornado, nombre de la función, y tipos de los parámetros; en Pascal el concepto prototipo es análogo al encabezado del procedimiento o de la función previa al bloque que lo define.

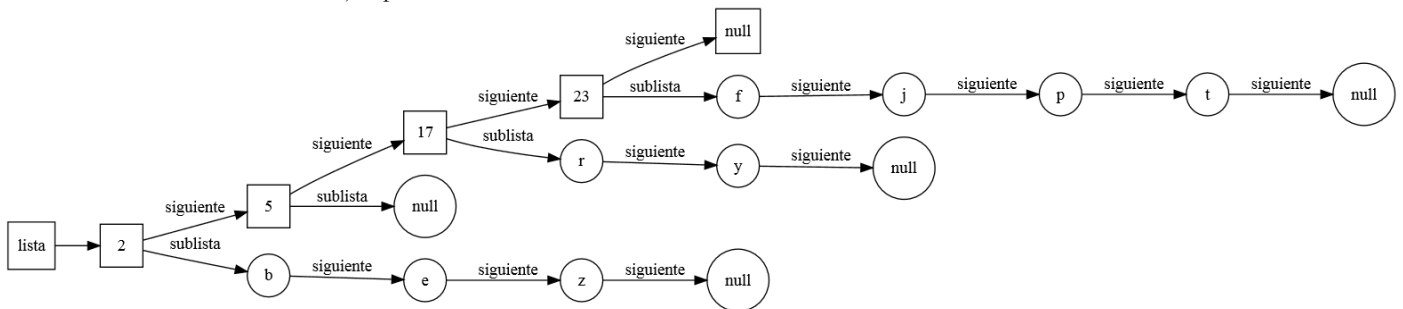
### 1. Fusión en una estructura abstracta de un archivo de registros con una lista enlazada

Temas evaluados: Corte de control, fusión, apareo, listas enlazadas, abstracción procedural, archivos, y flujos.

Diseñe o codifique un subprograma que fusione en una estructura abstracta el contenido de un archivo con el de una lista enlazada de listas enlazadas. Los datos a tratar son secuencias de valores del tipo carácter que se agrupan bajo valores del tipo entero. El prototipo debe ser:

**Pascal** `procedure Fusionar(n:String, l: Lista, var p: Puntero)` **C/C++** `void Fusionar(const string& n, const Lista& l, Puntero& p);`

Asuma que el archivo y la lista de listas están ordenados de menor a mayor. Los registros del archivo tienen el campo grupo que es un entero y campo valor que es un carácter. El primer nivel de la lista de listas, es una lista enlazada de enteros, y el segundo nivel es una lista enlazada de caracteres. Por ejemplo:



Asuma que los caracteres en los grupos de la lista están siempre en el rango [a-m] y que los que están en el archivo están en el rango [n-z]; lo que implica que primero deben agregarse los valores de la lista, y luego los del archivo.

El subprograma recibe una cadena con el nombre del archivo, y debe utilizar la biblioteca adjunta para recorrer ambas secuencias.

El parámetro p es de *salida*, los otros dos son de entrada.

La estructura que contendrá la fusión del archivo con la lista enlazada se debe manipular con las siguientes operaciones.

**Pascal** `function Crear():Puntero`

**C/C++** `Puntero Crear();`

Crea la estructura.

**Pascal** `procedure InsertarGrupo(var p: Puntero, g:Integer)`

**C/C++** `void InsertarGrupo(Puntero &p, int g);`

Inserta en la estructura apuntada por p un nuevo grupo con identificador g. Si la estructura ya contenía ese grupo, no lo inserta.

**Pascal** `procedure InsertarValorEnGrupo(p: Puntero, g:Integer, v:Character)`

**C/C++** `void InsertarValorEnGrupo(Puntero p, int g, char v);`

Inserta en la estructura apuntada por p el carácter v en el grupo g. Si g ya contiene v, no lo inserta. Si p no contiene g, lo inserta junto con v.

### 2. Selección de estructura eficiente

Temas evaluados: Estructuras de datos, abstracción de datos, y abstracción procedural.

Describa y justifique cual es la mejor estructura concreta para la estructura abstracta descrita en el punto 1.—

Apellido y nombre: \_\_\_\_\_ Legajo: \_\_\_\_\_ Curso con Prof: \_\_\_\_\_

Cantidad de hojas entregadas: \_\_\_\_\_ Nota: \_\_\_\_\_ Evaluó Prof: \_\_\_\_\_

- Si luego de la lectura del examen, durante la resolución tiene alguna duda, escriba hipótesis de trabajo, las cuales también serán evaluadas.
- Los puntos que solicitan codificación puede ser respondidos en C, C++, o en Pascal, pero debe indicar el lenguaje utilizado.
- En C y C++ prototipo refiere a la declaración de la función, es decir tipo de dato retornado, nombre de la función, y tipos de los parámetros; en Pascal el concepto prototipo es análogo al encabezado del procedimiento o de la función previa al bloque que lo define.

## 1. Lectura de Polinomios

*Temas evaluados: Estructuras enlazadas, abstracción, archivos, flujos, y lenguaje de programación*

### Problema

Evaluar los polinomios que se encuentran en un archivo y registrar los valores resultantes en una cola, para luego mostrarlos por pantalla.

**1a1.** Codifique la declaración de la estructura o registro del archivo de polinomios. El archivo contiene un registro para cada polinomio. Los polinomios son de grado tres como máximo. Un registro contiene el argumento con el cual se va a evaluar el polinomio, el grado del polinomio, y un arreglo con los coeficientes de todos los términos, inclusive los de coeficiente cero y del término independiente (o nulo). Por ejemplo, si el archivo contiene los siguientes cuatro polinomios:

$$P_1\left(x = \frac{1}{2}\right) = 5x^2 + \frac{1}{8}; P_2(x = 2) = \frac{1}{4}x^3 + 8x^2 + \frac{1}{5}x + 1; P_3(x = 2) = x; P_4(x = 21) = 7$$

entonces su representación es en el archivo es una secuencia 18 valores numéricos con la siguiente interpretación:

P1						P2						P3						P4					
Argumento	Grado	Coeficientes				Argumento	Grado	Coeficientes				Argumento	Grado	Coeficientes				Argumento	Grado	Coeficientes			
0.500	2	0.000	5.000	0.000	0.125	2.000	3	0.250	8.000	0.200	1.000	2.000	1	0.000	0.000	1.000	0.000	21.000	0	0.000	0.000	0.000	7.000

**1a2** Codifique el prototipo de la función *LeerPolinomio* que reciba un flujo (archivo) *f* abierto con polinomios, almacene los coeficientes en la pila *p*, y retorne si pudo leer o no. El término independiente o nulo debe quedar en la cima de *p*.

**1a3** Diseñe o codifique un subprograma para el anterior prototipo. La pila *p* debe manipularse con las operaciones de la biblioteca adjunta.

**1b1** Escriba el prototipo del subprograma *VaciarColaEnPantalla* que envíe a pantalla los contenidos de una cola de valores, uno por línea, la cola debe quedar vacía.

**1b2** Diseñe o codifique un subprograma para el anterior prototipo. Debe utilizar la biblioteca adjunta para manipular la cola correctamente.

**1c** Diseñe o codifique un programa que resuelva el problema planteado mediante la invocación repetida de *LeerPolinomio*, *EvaluarPolinomio*. Asuma que la función *EvaluarPolinomio(pilaDeCoeficientes, argumento)* está disponible. El resultado de la evaluación de cada polinomio se debe almacenar en la cola *q*, la cual deben manipularse con las operaciones de la biblioteca adjunta. Luego de evaluados todos los polinomios, se debe invocar a *VaciarColaEnPantalla*.

## 2. Estructura de datos eficiente

*Temas evaluados: Estructuras de datos, y eficiencia.*

Indique si la siguiente afirmación es verdadera o falsa; independientemente del valor de verdad, justifique claramente:

Tanto los arreglos como las listas enlazadas son igual de eficientes para representar secuencias ordenadas de elementos con posibles repeticiones.—

Apellido y nombre: \_\_\_\_\_ Legajo: \_\_\_\_\_ Cursó con Prof: \_\_\_\_\_

Cantidad de hojas entregadas: \_\_\_\_\_ Nota: \_\_\_\_\_ Evaluó Prof: \_\_\_\_\_

- Si luego de la lectura del examen, durante la resolución tiene alguna duda, escriba hipótesis de trabajo, las cuales también serán evaluadas.
- Los puntos que solicitan codificación puede ser respondidos en C, C++, o en Pascal, pero debe indicar el lenguaje utilizado.
- En C y C++ prototipo refiere a la declaración de la función, es decir tipo de dato retornado, nombre de la función, y tipos de los parámetros; en Pascal el concepto prototipo es análogo al encabezado del procedimiento o de la función previa al bloque que lo define.

## 1. Densidad poblacional

*Temas evaluados: Resolución de problemas, estructuras de datos, y lenguaje de programación*

### Contexto

En el contexto de estadísticas poblacionales mundiales, se subdivide el planisferio en 64.800 sectores rectangulares, 360 en longitud, y 180 en latitud. Cada sector contiene la densidad poblacional.

### Problema

Seleccionar la estructura de datos más eficiente para almacenar esta información en memoria principal. Notar que un sector puede incluir océano o continente.

**1a** Justifique con razones claras y pragmáticas la elección de la estructura, considere velocidad de acceso, actualización, y espacio utilizado.

**1b** Codifique la declaración de la estructura de datos.

## 2. Depuración de mediciones incorrectas

*Temas evaluados: Resolución de problemas, estructuras de datos, estructuras enlazadas, operaciones sobre secuencias, y lenguaje de programación*

### Problema

En un laboratorio de mediciones se requiere depurar registros erróneos. Se cuenta con una lista de mediciones de *ph* de una solución. Cada fecha se realizan tantas mediciones como horas tiene el día.

Luego de una revisión, se detectó que ciertas fechas pueden contener errores y, de ser así, deben descartarse. Para eso, analistas armaron una pila de fechas con una medición de referencia asociada, si la medición promedio para esa fecha es mayor a la de referencia, la fecha completa debe eliminarse, pero antes se debe informar la fecha eliminada junto con su valor de referencia y sus mediciones por hora.

### Solución

Diseñar un subprograma que reciba la lista enlazada de fechas con mediciones y la pila para que descarte las fechas con medición promedio superior a la de referencia e imprima el listado.

**2a1.** Codifique la declaración para los *entry* o *info* de la pila. Cada elemento contiene un *día*, *mes*, *año*, y *referencia*.

**2a2.** Codifique la declaración para los *entry* o *info* de la lista enlazada. Cada elemento contiene un *día*, *mes*, *año*, y un arreglo *mediciones*.

**2b1.** Escriba el prototipo de la función *Corregir* que reciba la lista *l*, la pila *p*, vacíe la pila *p*, actualice la lista *l*, envíe por pantalla el listado y retorne la cantidad de fechas eliminadas. Elija correctamente el mecanismo de transferencia de los argumentos.

**2b2.** Diseñe o Codifique la anterior función. La pila *p* debe manipularse solo con las operaciones de la biblioteca adjunta, la lista *l* debe operarse sin utilizar la biblioteca adjunta. El listado debe tener el siguiente formato:

```
dd/mm/aaaa
ref
00 medición
...
23 medición
```

—