

Apellido y nombre: _____ Legajo: _____ Curso con Prof: _____

Cantidad de hojas entregadas: _____ Nota: _____ Evaluó Prof: _____

1. Intersección¹

Temas evaluados: Listas enlazadas, abstracción procedural, estructuras de control, y lenguaje de programación.

1a. Escriba en Pascal, (puede ser C o C++, si así lo decidiera) el encabezado de una función o procedimiento que reciba dos listas enlazadas de enteros, a y b, y genere una tercera, c, con la intersección de a y b. Asuma que las listas a y b están ordenadas y que no contienen repeticiones. La lista c debe generarse ordenada. Escriba la definición de los tipos de datos no primitivos utilizados. Utilice el pasaje por referencia correctamente.

1b. Diseñe un algoritmo para la función 1a.

2. Capicúa

Temas evaluados: Pila y selección de estructura de datos eficiente.

Problema:

Determinar si una secuencia de dígitos ingresada por teclado es capicúa.

2a. Diseñe un algoritmo que resuelva el problema. Respete las siguientes restricciones:

- La longitud máxima de la secuencia es de N dígitos, valor conocido *a priori*.
- La longitud de la secuencia M, se ingresa por teclado antes de la secuencia y se cumple $M \leq N$.
- Cada dígito se ingresa teclado por separado.
- Para la resolución del problema se debe aplicar una pila y manejarla con las operaciones de pila listadas al dorso.

2b. Indique si es posible utilizar una representación contigua (arreglo) para la pila. Si la respuesta es afirmativa, indique para este caso porqué la utilizaría por sobre una representación enlazada.

3. Temperaturas del mundo

Temas evaluados: Registros, archivos, arreglos y resolución de problemas.

Enunciado:

Se dispone del archivo medidas.dat con las temperaturas del mundo del año pasado. Cada medida se compone de cinco componentes:

- día del año, un entero en el rango 1 a 366.
- hora, un entero en el rango 0 a 23.
- latitud, entero en el rango [-90, 90]. La latitud va de norte a Sur, en grados. El Sur está en el rango [-90,0) y el Norte en (0,90]; el ecuador es el cero.
- longitud, entero en el rango [-180, 180].
- temperatura, un número real.

3a. Diseñar un algoritmo que:

- Lea las temperaturas y las acumule en un arreglo de registros. El arreglo debe poseer tantas posiciones como latitudes; cada posición es una estructura con dos componentes, la temperatura acumulada y la cantidad de medidas.
- Informe al final cuál es el hemisferio más cálido².

¹ Recuerde el concepto de intersección, por definición debe pertenecer a ambos conjuntos.

² Se valorará particularmente la determinación del hemisferio aplicando promedios de forma precisa para el problema planteado (que es el promedio de los promedios de cada zona)

Apellido y nombre: _____ Legajo: _____ Curso con Prof: _____

Cantidad de hojas entregadas: _____ Nota: _____ Evaluó Prof: _____

Para la resolución del examen, los siguientes procedimientos y funciones *pueden* ser invocados sin ser implementados. Por cada procedimiento o función que *decida* invocar **debe reescribir su prototipo** reemplazando correctamente las palabras *subrayadas y en cursiva* (llamadas variables o no terminales); la **invocación debe ser precisa y concordar con el prototipo**.

Biblioteca genérica de plantillas

Operaciones sobre Archivos

```
function LeerEspecial(var archivo: TipoArchivo; var registro: TipoRegistro): Boolean
Lee un registro desde archivo y almacena en registro. Retorna pudo leer? True: False.

function BuscarBinArchivo(var archivo: TipoArchivo; valor: Tipo): Integer
Busca valor en archivo, mediante el algoritmo de búsqueda binaria. Retorna lo encontró? Posición: -1.

procedure BuscarBinArchivo(var archivo: TipoArchivo; clave: TipoInfo; var Registro: TipoRegistro)
Retorna el registro que tiene la clave buscada, la que se supone existe.
```

Operaciones sobre Arreglos

```
procedure OrdenarArregloPorCampo(var arreglo: TipoArreglo; n: Integer)
Ordena por campo los primeros n elementos de arreglo.

procedure CargarSinRepetir(
    var arreglo: TipoArreglo; var pos, n; var inserto: Boolean; clave: TipoInfo)
Carga una clave sin repetición en un arreglo, retorna en Pos, el índice donde lo encontró o lo inserto y en Inserto True, en caso de haberlo insertado

function BuscarBinArreglo(var arreglo: TipoArreglo; N: Integer; clave: TipoInfo): Integer

procedure BuscarBinArreglo(
    var arreglo: TipoArreglo; N: Integer; clave: TipoInfo; var Pos: Integer)
Similar a la búsqueda binaria en archivo, modificando la estructura de dato y con el agregado de N que representa el tamaño lógico del arreglo
```

Estructuras Enlazadas

Operaciones sobre Pilas

```
procedure Push(var pila: TipoPuntero; valor: TipoInfo)
Agrega valor a la cima de pila.

procedure Pop(var pila: TipoPuntero; var valor: TipoInfo)
Saca la cima de pila y la almacena en valor.

function Estavacia(pila: TipoPuntero): Boolean
Retorna True si la pila está vacía, si no, False.
```

Operaciones sobre Colas

```
procedure Agregar(var frente, fin: TipoPuntero; valor: TipoInfo)
Agrega valor al final de una cola.

procedure Suprimir(var frente, fin: TipoPuntero; var valor: TipoInfo)
Saca de el primer elemento una cola y lo almacena en valor.

function Estavacia(frente, fin: TipoPuntero): Boolean
Retorna True si la cola está vacía, si no, False.
```

Operaciones sobre Listas

```
procedure InsertarNodo(var lista: TipoPuntero; valor: TipoInfo)
procedure InsertarNodoDec(var lista: TipoPuntero; valor: TipoInfo)
Inserta en una lista ordenada en forma creciente (decreciente) por el primer campo valor.

procedure SuprimirNodo(var lista: TipoPuntero; valor: TipoInfo)
Busca un nodo con las características de valor, si lo encuentra lo elimina

procedure BuscarOInsertar(var lista, ptr: TipoPuntero; valor: TipoInfo)
Busca un nodo con los datos de valor, si no lo encuentra, lo inserta y retorna en ptr la dirección de memoria creada. Si estaba retorna en ptr esa dirección.

procedure InsertarPrimero(var lista: TipoPuntero; valor: TipoInfo)
procedure InsertarAntesDe(var lista: TipoPuntero; valor: TipoInfo)
procedure InsertarDespuesDe(var lista: TipoPuntero; valor: TipoInfo)
procedure InsertarAlFinal(var lista: TipoPuntero; valor: TipoInfo)
Cada procedimiento crea un nodo con valor y lo enlazan relativo a lista en la posición indicada por su nombre.

function BuscarPorCampo(lista: TipoPuntero; valor: Tipo): TipoPuntero
Busca el primer nodo con campo igual a valor. Retorna lo encontró? Puntero al nodo: Nil
```

Apellido y nombre: _____ Legajo: _____ Cursó con Prof: _____

Cantidad de hojas entregadas: _____ Nota: _____ Evaluó Prof: _____

1. Temperatura promedio de un área del mundo

Problema:

Se dispone del archivo con las temperaturas del mundo del año pasado. Cada medida con cinco componentes:

- El instante en el que se tomó la medida, representado por un segundo en el año. El primer segundo del año está en la hora cero, minuto cero del primero de enero, y tiene valor cero.
- latitud, entero en el rango $[-90, 90]$. La latitud va de norte a Sur, en grados. El Sur está en el rango $[-90, 0]$ y el Norte en $(0, 90]$; el ecuador es el cero.
- longitud, entero en el rango $[-180, 180]$.
- temperatura, un número real.

Se requiere guardar el promedio de las temperatura por latitud y longitud en una matriz de pares, con la cantidad de medidas y el promedio de temperaturas de cada latitud y longitud. La matriz resultante tiene tantas filas y columnas, como latitudes y longitudes tiene el mundo. Cada valor de la matriz es un par con la cantidad de medidas y el promedio de temperaturas de esa latitud y longitud.

1a. Escriba en Pascal (puede ser C o C++, si así lo decidiera) la definición del tipo de dato de la matriz y del par.

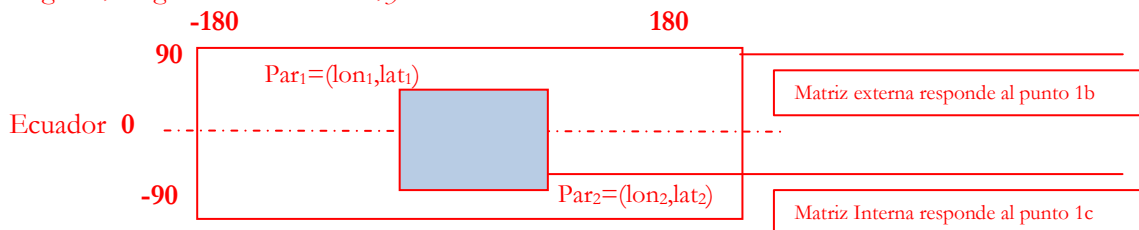
Temas evaluados: Registros, arreglos de dos dimensiones, abstracción de datos y lenguaje de programación.

1b. Diseñe el algoritmo de una función o procedimiento que dado el nombre de un archivo con medidas genere la matriz.

Temas evaluados: Registros, archivos, arreglos de dos dimensiones, abstracción de datos, y estructuras de control.

1c. Diseñe el algoritmo de una función o procedimiento que, a partir de la matriz generada en 1b y de un rectángulo representado por dos pares de longitud y latitud, determine la temperatura promedio del área en ese rectángulo. Asuma que el primer par longitud latitud es la esquina superior izquierda del rectángulo, y que el segundo par es la esquina inferior derecha. Recuerde procesar correctamente los puntos sin medidas.

Temas evaluados: Registros, arreglos de dos dimensiones, y estructuras de control.



2. Diferencia Simétrica

2a. Escriba en Pascal, (puede ser C o C++, si así lo decidiera) el encabezado de una función o procedimiento que reciba dos listas de enlazadas de enteros, a y b, y genere una tercera, c, con la *diferencia simétrica*¹ de a y b. Asuma que las listas a y b están ordenadas. La lista c debe generarse ordenada.

Temas evaluados: Listas enlazadas, abstracción procedural, y lenguaje de programación.

2b. Escriba en Pascal (puede ser C o C++, si así lo decidiera) la definición de los tipos de datos utilizados en 2a.

Temas evaluados: Listas enlazadas, abstracción de datos y lenguaje de programación.

2c. Diseñe un algoritmo para la función 2a. Utilice las operaciones sobre listas enlazadas que figuran al dorso.

Temas evaluados: Listas enlazadas, abstracción procedural, abstracción de datos, y estructuras de control.

3. Indique si la siguiente afirmación es verdadera o falsa; independentemente del valor de verdad, justifique claramente basándose en la lista al dorso: *Las operaciones InsertarNodoDec y SuprimirNodo son aplicables a colas.*

¹ La diferencia simétrica es el conjunto formado por elementos que pertenecen a A y a B, pero no a ambos.



Apellido y nombre: _____ Legajo: _____ Curso con Prof: _____

Cantidad de hojas entregadas: _____ Nota: _____ Evaluó Prof: _____

Para la resolución del examen, los siguientes procedimientos y funciones *pueden* ser invocados sin ser implementados. Por cada procedimiento o función que *decida* invocar **debe reescribir su prototipo** reemplazando correctamente las palabras *subrayadas y en cursiva* (llamadas variables o no terminales); la **invocación debe ser precisa y concordar con el prototipo**.

Biblioteca genérica de plantillas

Operaciones sobre Archivos

```
function LeerEspecial(var archivo: TipoArchivo; var registro: TipoRegistro): Boolean
Lee un registro desde archivo y almacena en registro. Retorna pudo leer? True: False.

function BuscarBinArchivo(var archivo: TipoArchivo; valor: Tipo): Integer
Busca valor en archivo, mediante el algoritmo de búsqueda binaria. Retorna lo encontró? Posición: -1.

procedure BuscarBinArchivo(var archivo: TipoArchivo; clave: TipoInfo; var Registro: TipoRegistro)
Retorna el registro que tiene la clave buscada, la que se supone existe.
```

Operaciones sobre Arreglos

```
procedure OrdenarArregloPorCampo(var arreglo: TipoArreglo; n: Integer)
Ordena por campo los primeros n elementos de arreglo.

procedure CargarSinRepetir(
    var arreglo: TipoArreglo; var pos, n; var inserto: Boolean; clave: TipoInfo)
Carga una clave sin repetición en un arreglo, retorna en Pos, el índice donde lo encontró o lo inserto y en Inserto True, en caso de haberlo insertado

function BuscarBinArreglo(var arreglo: TipoArreglo; N: Integer; clave: TipoInfo): Integer
procedure BuscarBinArreglo(
    var arreglo: TipoArreglo; N: Integer; clave: TipoInfo; var Pos: Integer)
Similar a la búsqueda binaria en archivo, modificando la estructura de dato y con el agregado de N que representa el tamaño lógico del arreglo
```

Estructuras Enlazadas

Operaciones sobre Pilas

```
procedure Push(var pila: TipoPuntero; valor: TipoInfo)
Agrega valor a la cima de pila.

procedure Pop(var pila: TipoPuntero; var valor: TipoInfo)
Saca la cima de pila y la almacena en valor.

function EstaVacía(pila: TipoPuntero): Boolean
Retorna True si la pila está vacía, si no, False.
```

Operaciones sobre Colas

```
procedure Agregar(var frente, fin: TipoPuntero; valor: TipoInfo)
Agrega valor al final de una cola.

procedure Suprimir(var frente, fin: TipoPuntero; var valor: TipoInfo)
Saca de el primer elemento una cola y lo almacena en valor.

function EstaVacía(frente, fin: TipoPuntero): Boolean
Retorna True si la cola está vacía, si no, False.
```

Operaciones sobre Listas

```
procedure InsertarNodo(var lista: TipoPuntero; valor: TipoInfo)
procedure InsertarNodoDec(var lista: TipoPuntero; valor: TipoInfo)
Inserta en una lista ordenada en forma creciente (decreciente) por el primer campo valor.

procedure SuprimirNodo(var lista: TipoPuntero; valor: TipoInfo)
Busca un nodo con las características de valor, si lo encuentra lo elimina

procedure BuscarOInsertar(var lista, ptr: TipoPuntero; valor: TipoInfo)
Busca un nodo con los datos de valor, si no lo encuentra, lo inserta y retorna en ptr la dirección de memoria creada. Si estaba retorna en ptr esa dirección.

procedure InsertarPrimero(var lista: TipoPuntero; valor: TipoInfo)
procedure InsertarAntesDe(var lista: TipoPuntero; valor: TipoInfo)
procedure InsertarDespuesDe(var lista: TipoPuntero; valor: TipoInfo)
procedure InsertarAlFinal(var lista: TipoPuntero; valor: TipoInfo)
Cada procedimiento crea un nodo con valor y lo enlazan relativo a lista en la posición indicada por su nombre.

function BuscarPorCampo(lista: TipoPuntero; valor: Tipo): TipoPuntero
Busca el primer nodo con campo igual a valor. Retorna lo encontró? Puntero al nodo: Nil
```