

# Obligatorio 3

## Programación 3

### Curso 2011

#### Versión 1.2

## 1 Objetivos

- Manejo de estructuras de datos avanzadas.
- Implementación eficiente de operaciones.
- Manejo de memoria dinámica y punteros.

## 2 Conocimientos previos

- Metodología de programación estructurada
- Estructura de datos avanzadas.
- Conocimientos básicos de C\*
- Análisis de algoritmos

## 3 Descripción del problema

El Comité Olímpico Internacional (COI) es el encargado de supervisar y administrar todo lo concerniente a los Juegos Olímpicos. Este comité, en la realidad aquí planteada, trabajara con la información acerca de los diferentes campeonatos que organizaran las federaciones de cada país el próximo año.

El cometido en este laboratorio es representar la realidad aquí planteada sobre el COI, de manera de satisfacer los órdenes solicitados para el grupo de operaciones de la sección Operaciones.

Los países miembros del COI se identifican por su nombre. Además es de interés conocer para cada país, el continente al que pertenece y el idioma oficial. Se puede asumir que existen  $P$  países máximo en esta realidad.

Las federaciones por su parte, son identificadas dentro del COI por el nombre del país al que pertenecen y por su respectivo nombre. De las federaciones también se conoce la fecha en que fue creada y una descripción. En la realidad planteada aquí, se asume que las federaciones son ingresadas en el mismo orden histórico en que se crean y que los países tienen como máximo  $F$  federaciones.

Las federaciones de cada país son quienes organizan los diferentes campeonatos del COI. De cada campeonato interesa conocer el nombre, la descripción y el nivel de importancia del campeonato dentro del COI. Un campeonato se identifica por su nombre, el nombre de la

federación que lo organiza y el nombre del país de la federación organizadora. Se puede asumir que una federación organiza como máximo  $C$  campeonatos.

El COI identifica a cada deportista mediante un código, este código es asignado cuando el deportista se da de alta en el sistema. Interesa conocer además el nombre, apellido, nacionalidad (País en el que compite) y edad de cada deportista. Un País tiene como máximo  $D$  deportistas.

Se conoce además todos los campeonatos a los que se inscribió cada deportista. Un deportista puede estar federado a distintas federaciones pero todas del mismo país.

### 3.1 Operaciones

Sea  $E$  la estructura de datos utilizada para representar la realidad anterior. Se desea realizar las siguientes operaciones:

**1. *Estructura\* crearCOI()***

Crea la estructura de datos vacía.

**2. *void destruirCOI(Estructura \*E)***

Destruye la estructura, liberando por completo la memoria asignada a ella.

**3. *void altaPais(const char\* nombrePais, Continente continente, const char\* idioma, Estructura \*E)***

Dada la estructura  $E$ , agrega el país de nombre  $nombrePais$ , con el continente  $continente$  e idioma oficial  $idioma$ .

Esta operación debe realizarse en  $O(1)$  promedio.

*Pre-condiciones:*

- El país de nombre  $nombrePais$  no pertenece a la estructura  $E$ .

**4. *void altaFederacion(const char\* nombrePais, const char\* nombreFederacion, const char\* dscFederacion, int diaCreacion, int mesCreacion, int anioCreacion, Estructura \*E)***

Dada la estructura  $E$  y el nombre de un país  $nombrePais$ , agrega a dicho país la federación de nombre  $nombreFederacion$ , con la descripción  $dscFederacion$  y el día  $diaCreacion$ , mes  $mesCreacion$  y año  $anioCreacion$  de creación.

Esta operación debe realizarse en  $O(1)$  promedio.

*Pre-condiciones:*

- El país de nombre *nombrePais* pertenece a la estructura *E*.
- El país de nombre *nombrePais* no tiene una federación de nombre *nombreFederacion* al momento de agregar la nueva federación.

**5. *void altaCampeonatoFederacion (const char\* nombrePais, const char\* nombreFederacion, const char\* nombreCampeonato, const char\* dscCampeonato, int importancia, Estructura \*E)***

Dada la estructura *E*, la federación de nombre *nombreFederacion* del país de nombre *nombrePais*, asigna el campeonato a dicha federación según su importancia.

Esta operación debe realizarse en  $O(P + F + \log(C))$  peor caso.

*Pre-condiciones:*

- El país de nombre *nombrePais* pertenece a la estructura *E*.
- La federación de nombre *nombreFederacion* del país de nombre *nombrePais* pertenece a la estructura *E*.
- La federación de nombre *nombreFederacion* del país de nombre *nombrePais* no tiene un campeonato registrado de nombre *nombreCampeonato* al momento de agregar el nuevo campeonato.

**6. *void altaDeportista (const char\* paisDeportista, int codigoCOI, const char\* nombreDeportista, const char\* apellidoDeportista, int edadDeportista, Estructura \*E)***

Dada la estructura *E*, agrega el deportista con código *codigoCOI*, nombre *nombreDeportista*, apellido *apellidoDeportista*, edad *edadDeportista* y país *paisDeportista*.

Esta operación debe realizarse en  $O(1)$  caso promedio.

*Pre-condiciones:*

- No existe un deportista con código *codigoCOI* en la estructura *E*.
- El país de nombre *paisDeportista* pertenece a la estructura *E*.

**7. *void imprimirFederaciones(const char\* nombrePais, Estructura \*E)***

Dada la estructura *E*, el nombre de un país *nombrePais*, imprime el nombre de las federaciones correspondientes a dicho país ordenados en forma ascendente por fecha de creación.

Esta operación debe realizarse en  $O(F)$  promedio.

*Pre-condición:*

- El país de nombre *nombrePais* pertenece a la estructura *E*.

Para la impresión se debe respetar la siguiente sintaxis:

Para cada federación se debe imprimir:

*nombreFederacion - dscFederacion - dia/mes/año*

Por ejemplo, si tenemos en Uruguay las siguientes dos federaciones:

- Fubb, cuya descripción es: federación uruguaya de basketball y la fecha de creación fue 1/3/1915
- Fudepyh, cuya descripción es federación uruguaya de patín y hockey, y la fecha de creación fue 27/6/1924

Se debe imprimir:

Fubb - federación uruguaya de basketball - 1/3/1915

Fudepyh - federación uruguaya de patín y hockey - 27/6/1924

**8. *void inscribirDeportistaCampeonato(const char\* nombrePais, const char\* nombreFederacion, const char\* nombreCampeonato, int codigoCOI, int mes, int dia, Estructura \*E)***

Dada la estructura *E*, inscribe al deportista de código *codigoCOI* al campeonato de nombre *nombreCampeonato* para el día *dia* del mes *mes*.

Esta operación debe realizarse en  $O(1)$  promedio.

*Pre-condiciones:*

- El país de nombre *nombrePais* pertenece a la estructura *E*.
- La federación de nombre *nombreFederacion* del país de nombre *nombrePais* pertenece a la estructura *E*.
- Existe un campeonato de nombre *nombreCampeonato* en la federación de nombre *nombreFederacion* del país de nombre *nombrePais*.
- El deportista de código *codigoCOI* no está inscripto al campeonato de nombre *nombreCampeonato* de la federación de nombre *nombreFederacion* del país de nombre *nombrePais* para el día *dia* del mes *mes*.

**9. *int obtenerCantidadPaises(Estructura \*E)***

Dada la estructura *E*, retorna la cantidad de países ingresados.

Esta operación debe realizarse en  $O(1)$  peor caso.

**10. *bool existeCampeonatoFederacion(const char\* nombrePais, const char\* nombreFederacion, const char\* nombreCampeonato, Estructura \*E)***

Dada la estructura *E*, devuelve TRUE si existe un campeonato de nombre *nombreCampeonato* asignado a la federación de nombre *nombreFederacion* del país de nombre *nombrePais*.

Esta operación debe realizarse en  $O(1)$  promedio.

*Pre-condiciones:*

- El país de nombre *nombrePais* pertenece a la estructura *E*.

- La federación de nombre *nombreFederacion* del país de nombre *nombrePais* pertenece a la estructura *E*.

**11. void imprimirFederados(const char\* nombrePais, const char\* nombreFederacion, Estructura \*E)**

Dada la estructura *E*, la federación de nombre *nombreFederacion* del país de nombre *nombrePais*, imprime los datos de los deportistas afiliados a dicha federación (No importa el orden).

Esta operación debe realizarse en  $O(D)$  promedio.

*Pre-condiciones:*

- El país de nombre *nombrePais* pertenece a la estructura *E*.
- La federación de nombre *nombreFederacion* del país de nombre *nombrePais* pertenece a la estructura *E*.

Para la impresión se debe respetar la siguiente sintaxis:

Para cada deportista se debe imprimir:

*codigoCOI - nombreDeportista - apellidoDeprtista - edadDeportista*

Por ejemplo, se debe imprimir:

10 – Juan – Pérez – 21

12 – Mariela – Vázquez – 27

17 – Agustina – Almeida - 18

**12. void imprimirCampeonatosDeportista(Estructura \*E, int codigoCOI)**

Dada la estructura *E*, imprime los datos de todos los campeonatos que llevará a cabo el deportista de código *codigoCOI* el próximo año, ordenados por fecha. Si el deportista tiene más de un campeonato el mismo día, no importa el orden de impresión de éstos.

Esta operación debe realizarse en  $O(PD + CF)$  peor caso.

*Pre-condiciones:*

- Existe un deportista con código *codigoCOI* en la estructura *E*.

Para la impresión se debe respetar la siguiente sintaxis:

Para cada deportista se debe imprimir:

*nombreFederacion - nombreCampeonato - descripcionCampeonato - importancia*

Por ejemplo, se debe imprimir:

Auf - Apertura – torneo apertura 2011 – 1

Fuv - Juvenil – campeonato juvenil departamental de volley - 4

## 4 Lenguaje a utilizar

El lenguaje a utilizar en este trabajo será C con las siguientes extensiones:

- Operadores `new` y `delete`.
- Pasaje de parámetros por referencia (uso de `&`).
- Declaración de tipos como en C++ para registros y enumerados.
- Sobrecarga de funciones.
- Uso de `cin` y `cout`.
- Uso del tipo `bool` predefinido en C++.

## 5 Qué se espera

Para cada módulo de cabecera (.h) con los prototipos de las operaciones solicitadas, debe entregarse un módulo (.cpp) con la implementación de dichas operaciones. Debe respetarse estrictamente los prototipos especificados, esto es: nombre de la operación, tipo, orden y forma de pasaje de los parámetros y tipo de retorno. Los **módulos de cabecera** pueden bajarse de la página web del curso ([www.fing.edu.uy/inco/cursos/prog3](http://www.fing.edu.uy/inco/cursos/prog3)). Estos módulos no forman parte de la entrega, y por lo tanto, **no deben ser modificados**.

**Los módulos deben funcionar en el ambiente MinGW instalado en facultad.** Se espera que todos los módulos compilen sin errores (utilizando las flags “-Wall” y “-Werror”), se ejecuten sin colgarse y den los resultados correctos.

## 6 Entregas semanales

Las entregas semanales se realizarán en los respectivos monitoreos.

- **Primer semana**  
Se deberá entregar un análisis del problema. Para esto se deberá diseñar la estructura (dibujo con los comentarios pertinentes) que permita soportar las operaciones mencionadas con los órdenes de ejecución indicados. También se deberá explicar detalladamente porque la estructura elegida cumple los órdenes de ejecución para cada operación.

## 7 Forma de la entrega final

Se deberá entregar únicamente los archivos (respetando las mayúsculas en los nombres):

- Estructura.cpp

No se podrá entregar otros archivos que no sean estos.

La primera línea de cada uno de los archivos debe contener un comentario (`/* ... */`) con la cédula de los autores, sin puntos ni dígito de verificación. Por ejemplo, si las cédulas son 1.234.567-8, 4.254.566-2 y 3.339.717-0, la primera línea de cada archivo deberá ser exactamente:

```
/* 1234567 4254566 3339717 */
```

## 8 Advertencia sobre el manejo de la memoria

Cuando un programa contiene errores en el manejo de la memoria, su comportamiento puede ser inestable. Esto implica que algunas veces funciona correctamente y otras no. En ciertos casos esto puede inducir a creer (erróneamente) que ciertos programas, que en realidad son incorrectos, funcionan correctamente. Este aspecto es influenciado, entre otras cosas, por el *sistema operativo* en el que se ejecuten los programas. Recomendamos tener sumo cuidado con este punto y testear los módulos en sistemas operativos Windows NT, Windows 2000 o Windows XP. CON LA VERSION DE MINGW INSTALADA EN LAS SALAS DE INFORMÁTICA DE LA FACULTAD.

## 9 Sobre la individualidad del trabajo

Para este obligatorio rige el reglamento de No Individualidad publicado en la página web del curso.

## 10 Fecha de entrega

El trabajo debe entregarse el día **domingo 9 de octubre del 2011** antes de las **22:00** horas. La entrega se realizará mediante un formulario que se habilitará oportunamente en la página web del curso.