



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA
ESTRUCTURAS DE DATOS Y ALGORITMOS I
M. en I. Fco. Javier Rodríguez García



Grupo

Práctica 4: Memoria dinámica	
Alumno:	
Número de cuenta:	
Fecha:	
Calificación:	

1 Objetivos

Que el alumno ponga en práctica lo estudiado en la teoría sobre la memoria dinámica construyendo y destruyendo objetos en la *free-store*. Al finalizar el alumno será capaz de:

- Crear y destruir arreglos dinámicos para tipos básicos del lenguaje.
- Crear y destruir arreglos dinámicos para TDAs.
- Crear y destruir objetos dinámicos.

2 Pre-laboratorio

Cuestionario

1. ¿Qué es la compilación multi-archivo?
2. ¿Qué es y qué hace el pre-procesador de C?
3. ¿Qué es la memoria dinámica?
4. ¿Qué funciones de la biblioteca estándar de C se usan comúnmente para usar la memoria dinámica? Haga una lista y dé una breve explicación de cada una.
5. ¿Qué es el área de memoria *stack* (pila) y qué tipo de variables se almacenan ahí?
¿Porqué a estas variables se les llama también “variables automáticas”?
6. ¿Qué es el *heap* (o *free-store*) y porqué las variables que se almacenan ahí no se “destruyen” automáticamente?
7. ¿Qué hace el operador `sizeof`?
8. ¿Qué es el casting (o moldeado o promoción, y porqué se necesita en las funciones de memoria dinámica?
9. ¿Qué son las estructuras de datos?
10. ¿Qué es una pila y cuál es su principal característica?
11. ¿Qué es una cola?
12. ¿Qué es una cola circular?

Ejercicios de Warm-up (en su casa o antes de entrar al laboratorio)

1. Termine el módulo **Reloj** utilizando las plantillas que está recibiendo junto con esta práctica (**Bool.h**, **Reloj.c** y **Reloj.h**). Compile y pruebe con el driver program que recibió en la práctica anterior. Antes de compilar asegúrese de eliminar de éste todo el código relacionado con el ADT Reloj, como lo hicimos en el salón.
2. Capture el código que se encuentra en el archivo **arreglo_dinamico.c.pdf** en un **archivo arreglo_dinamico.c**. Compile y pruebe.
3. Capture el código que se encuentra en el archivo **objetos_dinamicos.c.pdf** en un **archivo objetos_dinamicos.c**. Compile y pruebe.
4. Capture el código del punto 3 del documento **Matrices_dinámicas.pdf**. Compile y pruebe.

3 Actividades

3.1 Operaciones New y Delete

Agregue las operaciones **New()** y **Delete()** a su ADT **Reloj** del punto 3 de la sección “Ejercicios de *warm-up*”. Esto es, las declaraciones de las funciones en el archivo **Reloj.h**, y las definiciones en el archivo **Reloj.c**.

Pruébuelas en su driver program.

3.2 Problema del estacionamiento

Utilizando su ADT **Reloj** resuelva dicho problema. Para esto tendrá que agregar una operación **Reloj_Diff()** que devuelva la diferencia (resta) entre dos horas. La firma *podría* ser:

```
int Reloj_Diff (Reloj* this, Reloj* other);
```

Ud. decida si es conveniente usar **int** como valor devuelto –y qué significaría–, pero también podría usar cualquier otro tipo si Ud. cree que es lo mejor. Esta es una decisión de diseño.

En su programa deberá utilizar objetos dinámicos para la hora de entrada y de salida. Y recuerde que su programa deberá ser interactivo (*menu driven*) para probar lo que ha codificado y simular algunas acciones, como leer las horas de entrada y salida.

3.3 Matrices dinámicas

Modifique el punto 3 del documento *Matrices dinámicas.pdf* para que incluya una función **Crea_Matriz()** con el número de filas y columnas que el cliente decida. Por supuesto deberá agregar la contra-operación **Destruye_Matriz()** que devolverá la memoria pedida por la primer función. Escriba un driver program para probar su matriz dinámica. Las firmas de las funciones deberán ser:

```
int** Crea_Matriz (int filas, int cols);
```

```
void Destruye_Matriz (int** m);
```

El uso podría ser:

```
int** matriz = Crea_Matriz (n, m);  
// use la matriz como cualquier otra ...
```

```
Destruye_Matriz (matriz);
```

4 Post-laboratorio

Termine las actividades que hubieran quedado pendientes.

5 Entregables

Actividades 3.2 y 3.3, como el profesor se lo indicará más adelante.

6 Referencias y bibliografía recomendada

- [JOYANES05]
- [DEITELxx]
Deitel, H. M., Deitel, P. J. **Cómo programar en C/C++**. 2da. ed.
ESPAÑA: McGraw-Hill, 20xx.
- [JOYANES04]
- Referencia del language C
<http://www.cplusplus.com/reference/>