

Para que un programa C/C++ detecte las lagunas de memoria (memoria dinámica reservada y no liberada), se debe incluir el fichero `MemoryManager.h` al comienzo de cada archivo con extensión `.c/.cpp` que haga reserva dinámica de memoria (`malloc`, `new`, etc.) y añadir el fichero `MemoryManager-vc2012-d.obj` al construir el ejecutable si está utilizando el EDI Microsoft Visual C++, o bien el fichero `MemoryManager_GNU_GCC.o` si está utilizando un compilador GCC (*GNU Compiler Collection*), por ejemplo con el EDI Code::Blocks, o el fichero `MemoryManager-DevC-CB.a` si está utilizando el EDI Code_Blocks o Dev-C++. Estos ficheros pueden obtenerse en la página web de la asignatura y deben copiarse, los que se vayan a utilizar, en la carpeta del proyecto. A continuación, proceda de la forma siguiente:

Desde el EDI Microsoft Visual C++:

Añadir el fichero `.obj` al proyecto que ha generado.

Otra forma sería: Menú proyecto – Propiedades – Propiedades de configuración – Vinculador – Línea de comandos – en la caja Opciones adicionales escribir: `MemoryManager-vc2012-d.obj` – clic en el botón Aceptar.

Desde el EDI Code::Blocks (opción 1):

Menú Project – Build options – Debug/Release – Linker settings – en la caja Other linker options escribir el nombre del archivo: `MemoryManager_GNU_GCC.o` – clic en el botón OK.

Desde el EDI Code::Blocks (opción 2):

Menú Project – Build options – Debug/Release – Linker settings añadir: `MemoryManager-DevC-CB.a` – clic en el botón OK.

Pestaña Search directories - Linker - añadir ruta del fichero anterior.

Desde el EDI Dev-C++:

Menú proyecto – Opciones del proyecto – parámetros – en la caja Linker escribir: `MemoryManager-DevC-CB.a` – clic en el botón Añadir biblioteca u objeto.

Ejemplo de uso con C++ (Programación Avanzada):

```
#include "MemoryManager.h"
// otras directrices del preprocesador
// declaraciones de funciones

int main()
{
    { /* Observe que hay otro bloque dentro del bloque de main */
        // declaraciones de variables
        // cuerpo de la función main
    }
    MemoryManager::dumpMemoryLeaks();
}
```

Ejemplo de uso con C (SSII):

```
#include
#include
#include "MemoryManager.h"

int main()
{
    /* declaraciones de variables */

    /* cuerpo de la función main */

    MemoryManager_DumpMemoryLeaks();

    system("pause");
    return 0;
}
```