

Nota: Para cada paso del taller, se recomienda compilar por medio del botón “build” que se encuentra en la barra inferior de vscode. Luego de compilar, ejecutar los test por medio del botón “Run CTest” de la barra inferior (ejecuta en terminal) o por medio de la extensión “C++ TestMate” (ícono de erlenmeyer en la barra lateral).

Ejercicios de memoria dinámica

Abrir en vscode la carpeta “memoria_dinamica” y chequear que el compilador y cmake esté configurado.

Ejercicio 1) Punteros:

Objetivos: practicar reservar y liberar memoria dinámica. Conocer los errores de acceso a memoria:

- liberar memoria ya liberada.
 - acceder a memoria liberada o no reservada.
 - no liberar memoria reservada.
- a. Implementar en orden los tests del archivo “punteros_test.cpp”, guiándose por los comentarios.

Nota: EXPECT_EQ(a, b) hace pasar al test si a==b, falla en caso contrario. Si todos los EXPECT_EQ pasan, entonces el test ejecuta satisfactoriamente.

Ejercicio 2) Lista enlazada:

- a. Implementar borrarUltimo: dada una lista enlazada con al menos un elemento, esta operación debe borrar el elemento de la última posición. La memoria reservada para el elemento debe liberarse correspondientemente. Debe pasar hasta el test 06.
- b. Implementar borrarIesimo: dada una lista enlazada con al menos “posicion” elementos, debe borrar el elemento de la posición pasada por parámetro. Debe pasar hasta el test 09.

Sugerencia: reimplementar borrarUltimo a partir de borrarIesimo.

- c. Implementar actual: apunta a un elemento de la lista enlazada y se indefine si la lista no tiene elementos. Al ppio. debe apuntar al primer elemento que se agregue y si se borra el actual, queda apuntando al elemento siguiente. Debe pasar hasta el test 11.
- d. Implementar avanzar: cambia el elemento al que apunta actual a su siguiente. Nota; si actual apunta al último, se vuelve al primero. Debe pasar hasta el test 13.

- e. Convertir la lista de simple a doblemente enlazada. Debe seguir pasando hasta el test 13.
- f. Implementar retroceder: cambia el elemento al que apunta actual a su anterior. Nota; si actual apunta al primero, retroceder lo cambia al último. Deben pasar todos los tests.
- g. Implementar retroceder en $O(1)$.

Tip: Convertir la lista en circular

Ejercicio de templates

El ejercicio se encuentra en la carpeta “templates”.

Enunciado: dada la implementación de una lista simplemente enlazada de elementos `int`, transformarla para ser una lista de elementos genéricos `T`. Deben pasar todos los tests.