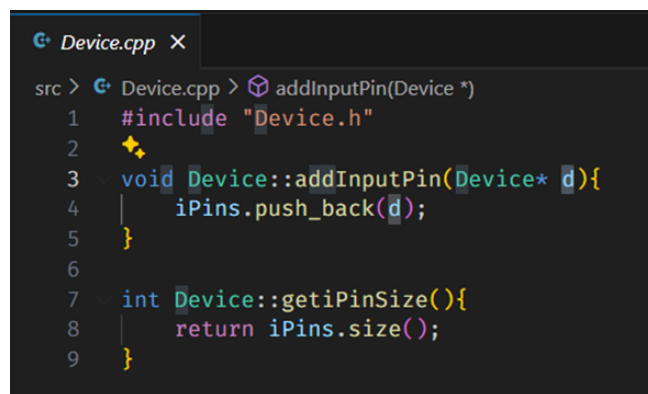


碩0面試作業-葉秉豐

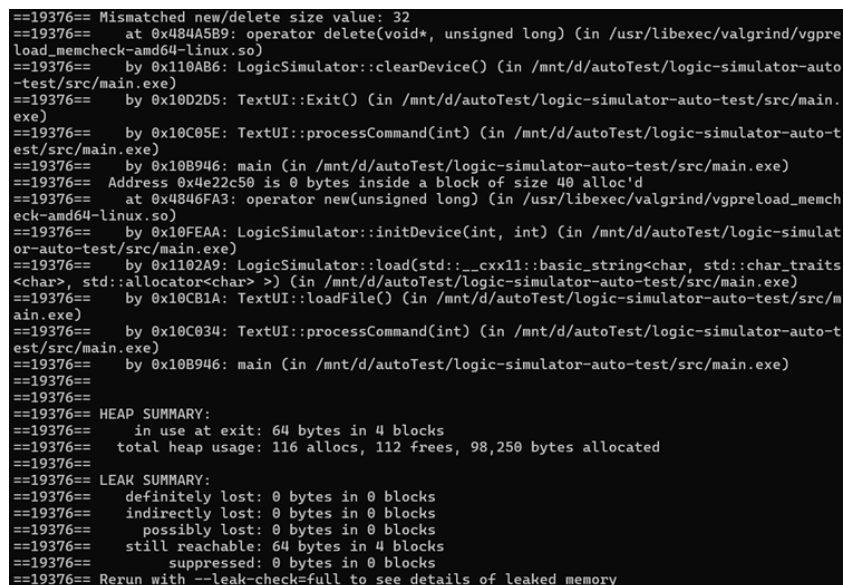
你的作業的功能大致上正確，以下是給你的一些作業建議：

1. 測試部分建議使用C++常用的框架，如Google Test等，可以用來撰寫預期結果來比對你的程式的實際結果。
2. 參數或變數盡量以有意義的名稱命名，提高程式的可讀性，例如下圖中的d。



```
src > Device.cpp X
src > Device.cpp > addInputPin(Device *)
1  #include "Device.h"
2
3  void Device::addInputPin(Device* d){
4      iPins.push_back(d);
5  }
6
7  int Device::getPinSize(){
8      return iPins.size();
9  }
```

3. 選擇Exit時應該會輸出一行Goodbye, thanks for using LS., 你的程式中並未印出。
4. 在你的程式中，我們有檢測到memory leak，這會令電腦可用的記憶體變少而導致電腦的效能變差。你可以利用相關工具比如Valgrind對這個部分研究跟改進。



```
==19376== Mismatched new/delete size value: 32
==19376== at 0x484A5B9: operator delete(void*, unsigned long) (in /usr/libexec/valgrind/vgpreload_memcheck-amd64-linux.so)
==19376== by 0x110AB6: LogicSimulator::clearDevice() (in /mnt/d/autoTest/logic-simulator-auto-test/src/main.exe)
==19376== by 0x10D2D5: TextUI::Exit() (in /mnt/d/autoTest/logic-simulator-auto-test/src/main.exe)
==19376== by 0x10C05E: TextUI::processCommand(int) (in /mnt/d/autoTest/logic-simulator-auto-test/src/main.exe)
==19376== by 0x10B946: main (in /mnt/d/autoTest/logic-simulator-auto-test/src/main.exe)
==19376== Address 0x4e22c50 is 0 bytes inside a block of size 40 alloc'd
==19376== at 0x4846FA3: operator new(unsigned long) (in /usr/libexec/valgrind/vgpreload_memcheck-amd64-linux.so)
==19376== by 0x10FEAA: LogicSimulator::initDevice(int, int) (in /mnt/d/autoTest/logic-simulator-auto-test/src/main.exe)
==19376== by 0x1102A9: LogicSimulator::load(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >) (in /mnt/d/autoTest/logic-simulator-auto-test/src/main.exe)
==19376== by 0x10CB1A: TextUI::loadFile() (in /mnt/d/autoTest/logic-simulator-auto-test/src/main.exe)
==19376== by 0x10C034: TextUI::processCommand(int) (in /mnt/d/autoTest/logic-simulator-auto-test/src/main.exe)
==19376== by 0x10B946: main (in /mnt/d/autoTest/logic-simulator-auto-test/src/main.exe)
==19376==
==19376== HEAP SUMMARY:
==19376==   in use at exit: 64 bytes in 4 blocks
==19376== total heap usage: 116 allocs, 112 frees, 98,250 bytes allocated
==19376==
==19376== LEAK SUMMARY:
==19376==   definitely lost: 0 bytes in 0 blocks
==19376==   indirectly lost: 0 bytes in 0 blocks
==19376==   possibly lost: 0 bytes in 0 blocks
==19376==   still reachable: 64 bytes in 4 blocks
==19376==   suppressed: 0 bytes in 0 blocks
==19376== Rerun with --leak-check=full to see details of leaked memory
```