

// Due: 5月16日

// 沒依照助教安排的時間完成demo: **作業零分**

/\*

/////////////////////////////////////////////////////////

繳交方式

 將作業的原始程式檔案以自己學號命名 上傳至 tronclass

 如有任何問題請與助教聯絡。

/////////////////////////////////////////////////////////

參考資料: Memory (C++標準庫) 的智慧型指標 shared\_ptr

https://zh.wikipedia.org/wiki/Memory\_(C%2B%2B%E6%A0%87%E5%87%86%E5%BA%93)

/////////////////////////////////////////////////////////

作業評分要求 :

1. 勿抄襲

2. 完成程式功能

 程式除了做規定的計算之外，必須在stdout顯示註解所列的訊息

3. 要徹底清除資料使用的記憶體

4. 詳細清楚的註解解釋程式:

 1) 助教老師能看懂

 2)

 Good Comment (程式碼的目的)

 if( record\_count > limit) // have all the records been processed?

 Bad Comment (程式碼的重覆敘述)

 if( record\_count > limit) // check if record\_count greater than limit

\*/



#include <iostream>

using namespace std;

class counter\_ptr

{

private:

 char \* \_name;

 int\_array\_cell \* \_cell\_ptr;

};

//template <class T>

class int\_array\_cell

{

private:

 int \_counter;

 int \* \_memory;

 //T\* \_memory; //template

};

void main()

{

// PART 一 (35分)

 counter\_ptr b("b", new int[10]); //需要撰寫 counter\_ptr constructor(建構子) 接受兩個參數

 // stdout 輸出: int\_array\_cell is allocated

 // stdout 輸出: counter\_ptr b is assigned to an int\_array\_cell: counter 1

 {

 b = new int[100]; //需要撰寫 counter\_ptr operator=

 // stdout 輸出: int\_array\_cell counter 0: deleted

 // stdout 輸出: int\_array\_cell is allocated

 // stdout 輸出: counter\_ptr b is assigned to an int\_array\_cell: counter 1

 counter\_ptr a("a"); //需要撰寫 counter\_ptr constructor(建構子) 接受一個參數

 // stdout 輸出: counter\_ptr a is not assigned to an int\_array\_cell

 a = b; //需要撰寫 counter\_ptr operator=

 // stdout 輸出: int\_array\_cell counter is increased: counter 2

 // stdout 輸出: counter\_ptr a is assigned to an int\_array\_cell: counter 2

 }

 // a leaves it scope: //需要撰寫 counter\_ptr destructor

 // stdout 輸出: counter\_ptr a is deleted

 // stdout 輸出: int\_array\_cell counter is decreased: counter 1

// PART 二 (35分)

 for(int i=0; i<10; i++)

 b[i] = i; //需要撰寫 counter\_ptr operator[] 寫值(回傳資料的參考 return type: int &)

 for(int i=0; i<10; i++)

 cout << b[i] << ' '; //需要撰寫 counter\_ptr operator[] 讀值(回傳資料)

 cout << endl;

 // stdout 輸出: 0 1 2 3 4 5 6 7 8 9

 counter\_ptr c("c");

 // stdout 輸出: counter\_ptr c is not assigned to an int\_array\_cell

 c = b;

 // stdout 輸出: int\_array\_cell counter is increased: counter 2

 // stdout 輸出: counter\_ptr c is assigned to an int\_array\_cell: counter 2

 b.release(); //需要撰寫 counter\_ptr release function: 不使用記憶體

 // stdout 輸出: int\_array\_cell counter is decreased: counter 1

 // stdout 輸出: counter\_ptr b is not assigned to an int\_array\_cell

}

 // c leaves it scope: //需要撰寫 counter\_ptr destructor

 // stdout 輸出: counter\_ptr c is deleted

 // stdout 輸出: int\_array\_cell counter 0: deleted

 // stdout 輸出: counter\_ptr b is deleted

/\*

PART 三 (30分)

 1) 用 template 改寫 class counter\_ptr int\_array\_cell

 2) 重覆 PART 一 二 的測試

\*/