

物件導向程式設計及應用第一次作業

Due: 2019/9/28 13:00

※注意事項：請依照課程網站內所公告之“作業檔案命名規則與規定”進行作業檔案命名以及繳交作業，未依照規定將斟酌扣分。

● 第一題：(30%)

利用樣版函式撰寫尋找陣列最大值的函式，其引數為傳入一維陣列的指標，以及一個陣列大小的整數，回傳計算結果。

在主程式的部份，產生一維動態陣列，利用亂數產生陣列的內容，並利用函數尋找陣列最大值。須利用樣板函式完成 6 個 short 值的陣列和 8 個 double 值的陣列。

另外撰寫樣板特定化，其引數為 char 指標的陣列，第二個引數為指標個數，回傳最長字串的地址。若有兩個最長字串，則回傳第一個最長字串的地址。請用 6 個 char 指標的陣列測試此特定化。

● 第二題：(70%)

請依照說明完成程式，請定義一個名稱空間 MyGeometry，其中包含一個結構(Structure)以及數個函式，宣告如下：

```
namespace MyGeometry
{
    struct Point ///#1
    {
        float x;
        float y;
        double Length(); ///#2
        string CoutPoint(); ///#3
    };
    Point* LoadPoint(const char* filename, unsigned int& nPoint); ///#4
    ..... MinDistance(.....); ///#5
    ..... SortbyLength(.....); ///#6
    void Report(const char* filename, .....); ///#7
}
```

1. 結構 point 至少包含兩個浮點數 x 與 y 表示二維空間中的點座標。
2. 結構內有成員函式 Length 以計算點到原點的距離。
3. 結構內有成員函式 CoutPoint 以輸出 string 格式，如 X 座標 3, Y 座標 2，則 CoutPoint () 會回傳 string 格式的字串(3, 2)。
4. 撰寫一個函數，在函數內開啟一個文字檔〔附件〕，文字檔內每行內的兩個浮點數代表每個點的 x 與 y 值，逐行讀入每個點，並回傳共幾個點。

5. 撰寫一個函數，將點陣列傳入找出最接近 $x+y=1$ 的點。(引數與回傳值請自行設計)
6. 撰寫一個函數，將點陣列依各點與原點距離由大到小排序。(引數與回傳值請自行設計)
7. 撰寫一個函數，在函數內寫入文字檔案。(引數請自行設計)

※寫入格式如下：

物件導向程式設計第一次作業

XXXXXXXX(學號) XXX(姓名)

點數目：(輸出點數目)

最接近 $x+y=1$ 的點：(x,y)

與原點距離最近的點：(x,y)、距離為：(輸出點距離)

- 請將上述宣告撰寫於 HW01_02.h 標頭檔案內。
- 請將上述宣告的定義撰寫於 HW01_02.cpp 程式碼檔案內。
- 請撰寫一個主程式於 useHW01_02.cpp，以測試上述功能。

※請勿使用標準樣板函式庫(Standard Template Library)或與巨集指令※