

程式設計第九次作業

Due: 2020/6/18 23:00

※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。

本次作業只有一題為程式撰寫。

● 作業題目：

類別 MyVector 表示三維向量，類別內包含：

1. 一個私有成員 `double* vector`，以動態陣列的方式存放向量在 x, y, z 軸的分量
2. 預設建構子、三個參數的建構子、拷貝建構子以及除構子
3. 計算向量長度的函式
4. 請重載以下函式
 - A. `operator=()`，設定運算子
 - B. `operator+()`，使兩向量進行相加
 - C. `operator-()`，使兩向量進行相減
 - D. `operator*()`，使可進行向量內積、向量乘上實數、實數乘上向量
 - E. `operator[]()`，使可單獨取出 x, y, z 軸的分量值
 - F. `operator>()`，比較兩向量的長度、向量長度與實數比大小、實數與向量長度比大小
 - G. `operator/()`，使向量除以一個實數，須注意分母不可為 0
 - H. `operator>>()`，輸入導向運算子
 - I. `operator<<()`，輸出導向運算子

附錄為此次作業之主程式，請直接複製至你的程式中並勿對其進行任何修改。

請在主程式程式碼後撰寫註解，若為該行為呼叫建構子，請說明呼叫何種建構子；若該行為呼叫類別成員函式，請說明分別呼叫那些函式。

向量類別宣告於 `MyVector.h`，定義於 `MyVector.cpp`，主程式撰寫於 `HW10.cpp` 中。

※請勿使用標準樣板函式庫(Standard Template Library)或與題目無關之巨集指令※

附錄：

```

int _tmain(int argc, _TCHAR* argv[])
{
    MyVector vec_A(1, 1, 1);
    MyVector vec_B(vec_A);
    MyVector vec_C = vec_A;
    MyVector vec_D;
    MyVector vec_E(0, 4, 2);
    MyVector vec_F;

    cout << "vec_A 為 " << vec_A << endl;
    cout << "vec_C 為 " << vec_C << endl;
    cout << "vec_C 與 vec_A 相加 ";
    vec_D = vec_C + vec_A;
    cout << vec_D << endl;
    cout << "vec_C 與 vec_A 相減 ";
    vec_B = vec_C - vec_A;
    cout << vec_B << endl;

    cout << endl;
    cout << "vec_C 為 " << vec_C << endl;
    cout << "vec_E 為 " << vec_E << endl;

    cin >> vec_F;
    cout << "vec_F 為 " << vec_F << endl;

    if (10 > vec_C)
        cout << "vec_C 長度小於 10" << endl;
    else
        cout << "vec_C 長度大於 10" << endl;
    if (vec_C > vec_E)
        cout << "vec_C 長度較長" << endl;
    else
        cout << "vec_E 長度較長" << endl;
    if (vec_C > 5)
        cout << "vec_C 長度大於 5" << endl;
    else
        cout << "vec_C 長度小於 5" << endl;

    cout << endl;
    cout << "vec_A 為 " << vec_A << endl;
    cout << "vec_E 為 " << vec_E << endl;

    cout << "vec_A 與 vec_E 內積 " << vec_A * vec_E << endl << endl;
    cout << "5 * vec_A = " << (5 * vec_A) << endl;
    cout << "vec_E / 2 + vec_A * 4 = " << (vec_E / 2 + vec_A * 4) << endl;

    cout << endl;
    cout << "向量 D (" << vec_D[0] << ", " << vec_D[1] << ", " << vec_D[2] << ")";

    return 0;
}

```