

A3261576 林郁翔 測試結果、說明:

總共有三題

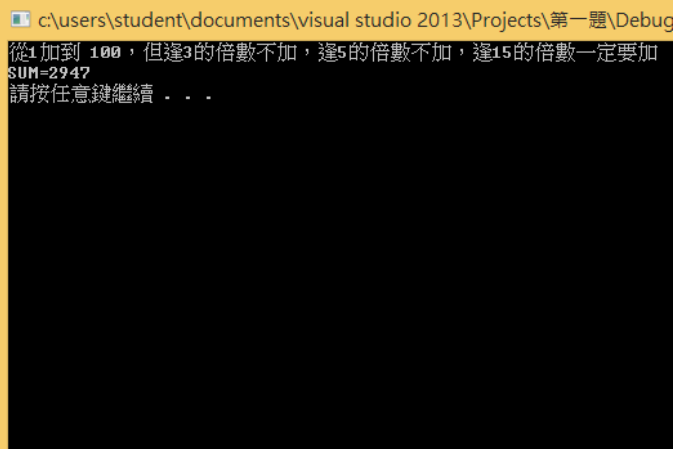
第一題:

```
using namespace std;

int main()

/*1. (20 points)請寫一個迴圈(while or for)，從1加到 100，但逢3的倍數不加，逢5的倍數不加，逢15的倍數一定要加，並印出最後的SUM。*/
int sum = 0;
for (int i = 1; i <= 100; i++)
{
    if (i % 15 == 0)
    {
        sum += i;
        continue;
    }
    if (i % 3 == 0 || i % 5 == 0)
    {
        continue;
    }
    sum += i;
}

cout << "從1加到 100，但逢3的倍數不加，逢5的倍數不加，逢15的倍數一定要加" << endl;
cout << "SUM=" << sum << endl;
system("pause");
```



說明:我先讓他判斷假若是 15 的倍數便直接增加，並使用

continue，換到下一次執行。

若非 15 的倍數，再來判斷 3 的倍數或 5 的倍數 假如是，直接

continue，不做 SUM 總和的增加

第二題:

```
nt main()

/*2. (30 points) 假設treacle為15個double的陣列
，treacle[i] = i2 + 3.7*i - 2.5; i = 1, ..., 15，
請宣告一個指標指向treacle的每個元素
並用指標顯示(印出)陣列的第j個元素(第j個
第j個(j由使用者輸入，3~14)
double treacle[15];
for (int i = 0; i < 15; i++)
{
    treacle[i] = (i+1)*(i+1)
}

double *Ptr = treacle;
int j;
//先印出第一個、第二個
cout << "第1個元素位置" << (Ptr) << endl;
cout << "第2個元素位置" << (Ptr+1) << endl;

cout << "請輸入j(於3~14之間): ";
cin >> j;

cout << "第" << j << "個元素位置" << (Ptr+j-1) << endl;
cout << "最後一個元素位置" << (Ptr+14) << endl;

return 0;
}

c:\users\student\documents\visual studio 2013\Projects\第2題
第1個元素位置0019FB60第1個元素值2.2
第2個元素位置0019FB68第2個元素值8.9
請輸入j(於3~14之間):
5
第5個元素位置0019FB80第5個元素值41
最後一個元素位置0019FBD0最後一個元素值278
請按任意鍵繼續 . . .
```

由於題目  $i$  是從 1~15 開始，為了不避免浪費 8bytes 的空間，直接用

```
for (int i = 0; i < 15; i++)
{
    treacle[i] = (i+1)*(i+1) + (3.7*(i+1)) - 2.5;
}
```

進行初始化，再來是請使用者輸入  $j$  並印出  $j$  的元素，再印出最後一個元素值。

第三題:

```

int main()
{
    /*3. (50 points)請寫出C++程式表示 複數 (complex) , 你必須
    複數 : a + bi; a, b是實數(請宣告為double);
    特徵值(attribute)必須宣告為 private
    要建至少一個建構子, Complex(double x, double y)
    addComplex: 將兩個複數相加 Complex addComplex(Complex x, Complex y)
    mulComplex: 將兩個複數相乘 Complex mulComplex(Complex x, Complex y)
    showComplex: 印出一個複數; (a, bi)
    測試你的程式*/

    Complex a(2, 5), b(4, 3), c(5, 7); // 宣告abc 三個物件
    cout << "目前兩複數:" << endl;
    a.showComplex();
    cout << '\n';
    b.showComplex();
    cout << '\n';
    cout << "請選擇執行方法" << endl;
    cout << "相加:請輸入0"<<endl;
    cout<<"相乘 : 輸入0之外的整數數字為相乘" << endl;
    short boo;
    cin >> boo;
    c = (boo != 0) ? c.mulComplex(a, b) : c.addComplex(a, b);
    c.showComplex();
}

```

```

c:\users\student\documents\visual studio 2013\Projects\Complex\Complex.cpp
目前兩複數:
<2,5i>
<4,3i>
請選擇執行方法
相加:請輸入0
相乘 : 輸入0之外的整數數字為相乘
0
<6,8i>請按任意鍵繼續 . . .

```

使用一個 **C** 當作運算結果的儲存物件。

比較特別的是我是用 **\*this** 做回傳，因此 **set** 可以串接，賦予值

比較方便(使用在 **Complex** 類別裡面)。

而使用者可以

相加:輸入 0

相乘:輸入 0 以外整數

已預先設定兩個複數(2,5i)及(4,3i) 以此兩值做相加、相乘

其中 虛部跟實部我所定的變數為:

實部 **double numR** 用來暫存實部的值

虛部 **double numZ** 用來暫存虛部的值