

# 資料結構與進階程式設計 (106-2)

## 作業一

作業設計：孔令傑

國立臺灣大學資訊管理學系

繳交作業時，請以英文或中文於紙本作答。不接受遲交。請用電腦打字並印出；**不接受手寫作答**。這份作業的截止時間是 **2018 年 3 月 12 日下午 2:40**。你也可以選擇在當天 **下午 2:10** 前把作業放進老師在管二一樓的信箱。在你開始前，請閱讀課本的第 9-11 章<sup>1</sup>。

由於本課程修課同學眾多，而助教人力有限，為了避免過度壓榨助教，我們可能會進行隨機批改，亦即助教會隨機選擇一份作業的若干題進行批改和給分。助教會對全班同學都批改同樣的題目。若有批改之題目的總分為  $x$  而一位同學得到  $y$  分，則其在該份作業會得到  $100\frac{y}{x}$  分。公佈的參考解答中會包含所有題目（包含沒被批改的）。

## 第一題

(20 分) 針對上課教的 `MyVector`，請幫 indexing operator `[]` 做以下版本的 operator overloading：

```
double MyVector::operator [] (char* c) const;
```

傳入的參數是一個 C string  $c$ ，格式為一個整數  $i$ 、一個冒號，再一個整數  $j$ 。如果  $i$  或  $j$  小於 0 或大於等於  $n$ ，就結束程式；如果  $j$  沒有比  $i$  大，也結束程式。如果  $0 \leq i < j \leq n-1$ ，則回傳  $m[i]$ 、 $m[i+1]$  直到  $m[j]$  的和。舉例來說，以下的程式

```
double d[5] = {1.1, 2.2, 3.3, 4.4, 5.5};
MyVector v(5, d);
char c[] = "1:3";
cout << v[c];
```

應該要印出 9.9。請用 C++ 完整地實作；如果有幫助，請加上適當的註解。

## 第二題

(15 分) 承上題，針對同一個功能，你的朋友不想要傳入 C string，而是想要傳入兩個整數，像這樣：

```
double MyVector::operator [] (int i, int j) const;
```

並且使得

```
double d[5] = {1.1, 2.2, 3.3, 4.4, 5.5};
MyVector v(5, d);
cout << v[1, 3];
```

會印出 9.9。如果這做得到，請幫他用 C++ 完整地實作；如果這不可能，請解釋為什麼。

<sup>1</sup>課本是 Deitel and Deitel 著的 *C++ How to Program: Late Objects Version* 第七版。

## 第三題

(15 分) 承上題，針對同一個功能，你的朋友不想要傳入 C string，而是想要傳入一個 structure `TwoInt`：

```
struct TwoInt
{
    int i;
    int j;
};
```

並且使得

```
double d[5] = {1.1, 2.2, 3.3, 4.4, 5.5};
MyVector v(5, d);
TwoInt ti = {1, 3};
cout << v[ti];
```

會印出 9.9。如果這做得到，請幫他用 C++ 完整地實作；如果這不可能，請解釋為什麼。

## 第四題

(25 分) 假設有兩個向量  $x$  和  $y$ ，維度各是  $n_x$  和  $n_y$ ， $n_x < n_y$ ，則他們「循環相加」後會得到一個維度為  $n_y$  的向量<sup>2</sup>，其前  $n_x$  項即為  $x$  和  $y$  的前  $n_x$  項相加、第  $n_{x+1}$  項為  $x$  的第 1 項和  $y$  的第  $n_{x+1}$  項相加、第  $n_{x+2}$  項為  $x$  的第 2 項和  $y$  的第  $n_{x+2}$  項相加、依此類推。 $n_y$  不必然是  $n_x$  的整數倍，所以  $x$  裡面每一項被用到的次數未必相同；如有必要， $x$  可能會被循環不只一次。 $x + y$  和  $y + x$  要得到一樣的結果。舉例來說：

- $(1, 2, 3) + (4, 5, 6, 7, 8) = (5, 7, 9, 8, 10)$ 。
- $(1) + (4, 5, 6) = (5, 6, 7)$ 。
- $(10, 9, 8, 7, 6) + (1, 2) = (11, 11, 9, 9, 7)$ 。

針對上課教的 `MyVector`，請幫 + 做以下版本的 operator overloading：

```
const MyVector MyVector::operator+(const MyVector& v) const;
```

傳入的參數是一個 `MyVector v`。如果  $v$  的維度和 invoking object 一樣，就回傳一個 `MyVector`，其中的第  $i$  項是 invoking object 的第  $i$  項和  $v$  的第  $i$  項的和；如果不一樣，就做「循環相加」。舉例來說，以下的程式

---

<sup>2</sup>如果你寫過 R，R 裡面兩個向量相加就是循環相加。

```
double d[5] = {1.1, 2.2, 3.3, 4.4, 5.5};
MyVector u(5, d);
double f[2] = {1.1, 2.2};
MyVector v(2, f);
MyVector w = u + v;
w.print();
```

應該要印出 (2.2, 4.4, 4.4, 6.6, 6.6)。請用 C++ 完整地寫下此 overloaded operator 的實作；如果有幫助，請加上適當的註解。

## 第五題

(25 分) 針對上課教的 MyVector，請 overload 運算子 >>：

```
istream& operator>>(istream& in, MyVector& v);
```

以使使用者可以用 `cin >> v` 這樣的運算，去把資料從鍵盤輸入寫入到 `MyVector v`。寫入的規則是使用者會先輸入一個左角括號，接著每輸出一個數字，就在後面接一個逗點，但最後一個數字是例外，其最後是一個右角括號。從頭到尾都不會出現空白字元。不論使用者輸入的向量維度是否和 `v` 原本的維度相同，都要把 `v` 徹底變成使用者輸入的向量。舉例來說，以下的程式

```
double d[5] = {1.1, 2.2, 3.3, 4.4, 5.5};
MyVector u(5, d);
cin >> u;
u.print();
```

如果使用者想要讓 `u` 變成 (3,14,5)，則他會輸入

```
<3,14,5>
```

然後此程式應該要印出

```
(3, 14, 5)
```