

### 第一題

#### 1. find

```
size_t find (const string& str, size_t pos = 0) const noexcept;
```

```
size_t find (const char* s, size_t pos = 0) const;
```

```
size_t find (const char* s, size_t pos, size_type n) const;
```

```
size_t find (char c, size_t pos = 0) const noexcept;
```

括號裡面可以放要被尋找的字串的指標或 **reference** 或是字元，也可再傳入要尋找的起始位置，或是傳入的字串要被尋找的長度。

#### 2. find\_first\_of

```
size_t find_first_of (const string& str, size_t pos = 0) const noexcept;
```

```
size_t find_first_of (const char* s, size_t pos = 0) const;
```

```
size_t find_first_of (const char* s, size_t pos, size_t n) const;
```

```
size_t find_first_of (char c, size_t pos = 0) const noexcept;
```

括號裡面可以放要被尋找的字串的指標或 **reference** 或是字元，也可再傳入要尋找的起始位置。

回傳字串中與被尋找字串中相同的第一個字元。

#### 3. find\_first\_not\_of

```
size_t find_first_not_of (const string& str, size_t pos = 0) const noexcept;
```

```
size_t find_first_not_of (const char* s, size_t pos = 0) const;
```

```
size_t find_first_not_of (const char* s, size_t pos, size_t n) const;
```

```
size_t find_first_not_of (char c, size_t pos = 0) const noexcept;
```

括號裡面可以放不被包含的字串的指標或 **reference** 或是字元，也可再傳入要尋找的起始位置。

回傳字串中與被尋找字串中不包含的第一個字元。

4. 異同:**find** 只能一次找一種東西，**find\_first\_of** 可以一次在字串裡面找很多東西的其中一個相同的，**find\_first\_not\_of** 也可以一次找很多種字串裡面沒有的，他們都是在字串中尋找有或沒有的，都是回傳位置。

### 第二題

**str.length()**回傳的是一個 **size\_t**，而 **i** 是 **int**，兩者的類型不太一樣。

中文編碼不是一個字一個字元編，可能是兩個，所以這樣比較就會出問題，要兩個兩個比較。

### 第三題

(a)

```
size_t getSpaceCount(const size_t from, const size_t to) const {  
    string T = *this;  
    size_t found = T.find_first_of(' ');  
    int count = 0;  
    while(found != string::npos){  
        count ++;  
        T[found] = '*';  
        found = T.find_first_of(' ');  
    }  
    return count;  
}
```

(b)

前兩個 `const` 是起點跟終點的，避免起點、終點被更改。最後一個 `const`，表示這個函數並不會更改這個 `object` 本身。

回傳值不需要加 `const`，因為是回傳幾個空白的結果，增加 `const` 並沒有太大意義。

### 第四題

這三個詞語一起使用是用來避免重複 `include`。

`#ifndef`，if not defined 的意思，如果沒有 `define` 這個標頭執行下列的程序。

`#define`，define 一個東西(常數)，跟 `ifndef` 一起用的時候會用來 `define header`。

`#endif`，跟 `#ifndef` 一組，做為一個段落。

`myMax.h` 需要增加為，

```
#ifndef _Max_H_
```

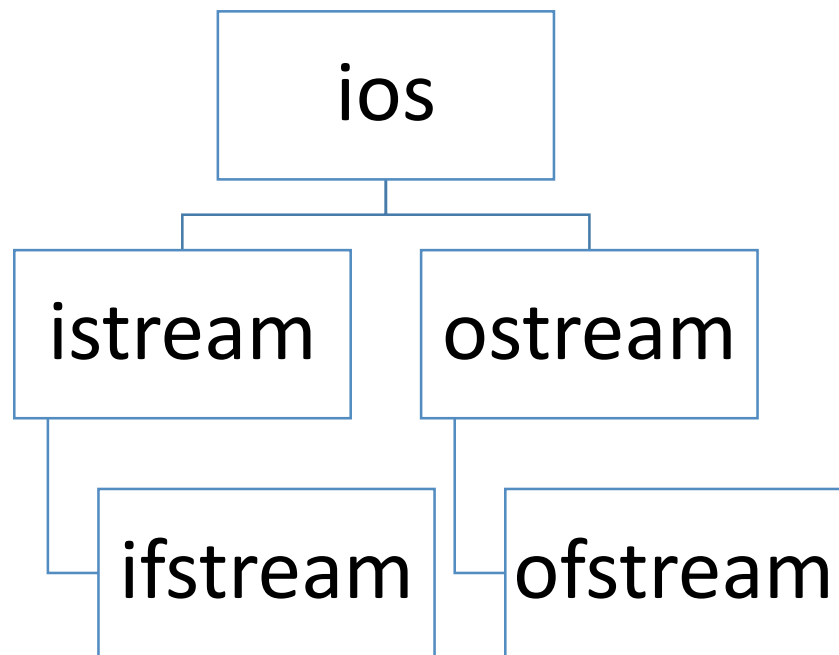
```
#define _Max_H_
```

原本的程式碼

```
#endif
```

其他 `main.cpp`、`myMax.cpp` 不需要變更。

## 第五題



在同意個系列之下的繼承才能使用多形，<<是在 `ostream` 之下，所以 `ofstream` 可以利用多型沿用，但是 `cin` 物件在 `istream` 下，所以沒有辦法使用，`ifstream` 也是在 `istream` 的繼承底下。