

- Python 課程課本：大澤文孝, <Python入門教室>, 臉譜

- C/C++ 課本：

- (1) 基礎：蔡志敏, <Hello! C++ 程式設計 (融合「大學程式設計先修檢測APCS」)>, 碁峰 (建議初學者購買可作學習參照)

- (2) 進階：黃建庭, <C++ 程式設計解題入門 融入程式設計競賽與 APCS 實作題>, 碁峰

- (3) 進階：鄧文淵, <APCS 大學程式設計先修檢測最強考衝特訓班：C/C++ 解題攻略>, 碁峰

- 課程進度非依照課本內容，在學習期間參考課本內容提供概念穩固形成的支持。

- 建議同學學習順序

- (1) 課程提供簡單的觀念核心架構

- (2) 課本或自學網站有多一點細節的系統性陳述

- (3) 筆記部落格和技術論壇提供更多元應用資訊。

- 其他參考書：(無須購買，可以去借來看)

- 謝宗翔, <什麼啦！這也是資訊科學>, 聚光文創

- 秋葉拓哉, 岩田陽一, 北川宜稔, <培養與鍛鍊程式設計的邏輯腦>, 博碩

- 陳小玉, <趣學演算法：50 種必學演算法的完美圖解與應用實作>, 碁峰

- 石田保輝, 宮崎修一, <演算法圖鑑：26種演算法 + 7種資料結構，人工智慧、數據分析、邏輯思考的原理和應用全圖解>, 臉譜

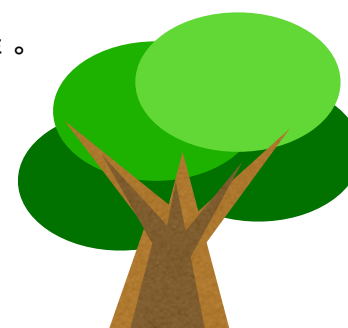
- 授課老師：黃老師

- 留美電腦工程碩士，APCS 專門授課

- 經歷包括美國矽谷、貝爾實驗室、ARM 等頂尖科技公司與機構擔任軟體開發與應用工程等工作

- 曾任教於公立學校、立案電腦專業培訓機構授課

- 擁有演算法 (演算法) 專利發明



- 初學常見錯誤

類型	正確	錯誤	說明	警告或錯誤訊息
運算符號	<code>if (a == b) ...</code>	<code>if (a = b) ...</code>	(或迴圈判斷式處) 兩個等號是比較，一個是指定 注意：只有警告	warning: suggested parentheses around assignment used as truth value 警告：建議將用作真假值的指定加上括弧
輸入串流運算符號	<code>int m, n; cin >> m >> n;</code>	<code>int m, n; cin >> m, n;</code>	輸入多個變數用 >> 分開 注意：只有警告	warning: right operand of comma operator has no effect 警告：逗號運算子的右運算元無作用
輸出串流運算符號	<code>cout << "x:" << x << endl;</code>	<code>cout << "x:" x << endl;</code>	輸出項目用 << 分開	error: expected ";" before '...' 錯誤：在 '...' 前期望有 ";"
輸出入串流運算符號	<code>cin >> x; cout << x;</code>	<code>cin << x; cout >> x;</code>	想像訊息流動的方向	error: no match for operator ... 錯誤：無法匹配運算子 ...
忘記換行	<code>cout << x << endl;</code>	<code>cout << x;</code>	如果題目要求換行，要輸出換行	(無) 注意：沒有任何訊息
忘記分號	<code>cin >> x;</code>	<code>cin >> x</code>	每個指令，說完一句話後要分號。 if (...) 只說了半句	error: expected ";" before '...' 錯誤：在 '...' 前期望有 ";"
忘記大括弧	<code>for (auto i: a) { cout << i << " "; cout << endl; }</code>	<code>for (auto i: a) cout << i << " "; cout << endl;</code>	超過一個指令，要做的事不只一件， 要用大括弧包成一個區塊	(無) 注意：沒有任何訊息
懸掛的 else	<code>if (...) { if (...) ... } else ...</code>	<code>if (...) if (...) ... else ...</code>	else 會找最接近的 if 配對。 根據算法邏輯加上大括號或是每個 if 都配上 else 注意：只有警告	warning: suggest explicit braces to avoid ambiguous 'else' 警告：建議加上括弧避免模稜兩可的 else
使用保留字作變數名稱	<code>int dot;</code>	<code>int do;</code>	C/C++ 語言保留字不得作為變數或函式名稱	error: expected unqualified-id before '...' 錯誤：在 '...' 前期望有 unqualified-id

- while/for 迴圈語法中的條件是迴圈**持續**的條件，**不是中止**的條件 (比較迴圈中可以有的 if/break；和自然語言的習慣一致)。
- while/for 迴圈中，while/for 那列陳述包含迴圈每次迭代所要檢查的條件 (以及在 for 迴圈時每次條件的改變)，所以條件的初始設定在迴圈開始**前**，不是在迴圈內。
- for 迴圈注意迴圈的持續條件，變數遞增的情況較遞減的常用一些，特別注意遞減時的情況。
- 輸出結果如果出現亂七八糟奇怪的數字，或是 Code::Blocks 開始卡住或不正常工作，可能的原因之一是你的程式碼陣列宣告得不夠大，以至於執行時錯誤地使用了沒有分配給你的記憶體區域。適時使用 build clean。
- 到目前為止，實作題最容易的 P1，其次 P2，P3、P4 比較難；P4 至少有三次沒有比 P3 難，只是敘述很長，試圖把複雜的觀念講清楚，但程式要怎麼寫題目也說明了，所以只要轉換成程式碼就可以了，不用特別去思考設計運算法。觀念題最後幾題也傾向放較難或是敘述較複雜的題組，但每題的配分都一樣，所以挑容易、簡單、閱讀理解和處理所需的時間較少的題目先做。
- 觀念題和實作思考時變數的地方用常數代入可以比較快想清楚。
- **注意考試時要根據題號命名 P1.cpp、P2.cpp、P3.cpp、P4.cpp，並在提交後複製一份程式碼到 backup 檔案夾中。**
檔名大小寫注意聽監考老師宣布和詳讀規定，109 年 10 月：主檔名大寫，副檔名小寫。
- 輸出列尾該換行就要換行，特別注意最後一列。
處理輸入連續測資。注意題目允不允許輸出列尾的空格，可以在答題系統驗證看看。
沒有解題系統驗證大測資，所以解題時就儘量做好效能的考量
包括加入

```
ios::sync_with_stdio (false);  
cin.tie (0);  
cout.tie (0);
```


但第一題通常不會太複雜 (第一題先求有再求好，測資有可能不會有太嚴厲的要求，所以先做出來，有時間再去想說怎麼樣可以更好、更快；但是注意滿分測資的要求，盡量從滿足滿分的測資想解，例如 <秘密差> 那一題。)
- 大陣列宣告成全域變數。
- 考試**可以上廁所**，跟監考老師說，他會帶你過去，再帶你回來。109 年 10 月：不可以帶橡皮擦，紙不夠用跟監考老師再要。
- Code::Blocks 管理視窗不見了: View → Manager, Logs, Toolbars → ...
Code::Blocks 編輯視窗內按滑鼠右鍵 → Split view 可以分割編輯視窗，程式長長後要參照前後很好用。
- Code::Blocks 用 `int a [200] = { 0 };` 之後寫入超過 `a [10]` 之後的元素在 debugger 中都會看到寫到 `a [1]` (但實際上可能正常)。
- Code::Blocks
Settings->Debugger->Default->Debugger initialization commands
`set max-value-size unlimited`
- 學過 Python 的同學注意在 Python 不需要大括弧、小括弧的地方，C/C++ 需要。
- GreenJudge 版本的 C++ 不接受可變大小的陣列宣告，也沒有萬能標頭 `bits/stdc++.h` 可用。

- 常用數值與技巧：¹

- 1 累加到 $10 = 55$
- 2 的次方倍：1, 2, 4, 8, 16 (2^4), 32, 64, 128, 256 (2^8), 512, 1024 (2^{10}), 2048, 4096
- 最大 int: $(2^{31} - 1) = 2,147,483,647 \div 2 \times 10^9$ ：10 進位時為 10 位數 / unsigned 數值多一倍，位數仍為 10 位
最大 long long: $9,223,372,036,854,775,807 \div 9 \times 10^{18}$ ：10 進位時為 19 位數 / unsigned 數值多一倍，位數增加一位為 20 位
- 二元搜尋時間複雜度: $\log_2 n$
- 判斷奇數：(n % 2) == 1 或 (n & 1) == 1
- 個位數數字：n % 10

- 邏輯運算真(假)值表 truth table:

	真	假
!	假	真

&&	真	假
真	真	假
假	假	假

	真	假
真	真	真
假	真	假

- 笛摩根定理： $!(p \&\& q) == (!p) || (!q)$
- 運算優先順序不確定時，保守一點多加一些括弧
- 在 xterm d 上：滑鼠中鍵 (滾輪往下按)，在其他視窗複製貼上: Ctrl-C, Ctrl-V / 滑鼠右鍵 / 視窗命令列「編輯」下拉
- 先用紙筆操作範例可以幫助並確定對問題的正確了解，並且從中構思解法
- 實作數字的問題有時必須 (106-3-p1 第 3 子題組) 用字串處理會比較容易
- 費氏數列、輾轉相除常考
- 用 Leafpad 整理測資方便每次測試的複製貼上。
- Ubuntu Linux 計算工具：LXTerminal → bc -l。活用編譯指令 #define/#if/#else/#endif

¹// 寫程式一定要寫註解。幫助日後看懂程式外，隨著註解複習自己的程式加深所使用編程技巧的印象

• 好程式標準：最重要不要錯，盡量不要有 bug，易讀好改 (容易維護)，其次在記憶體不爆掉的情況下跑得快，再其次精簡

```

• #include <bits/stdc++.h>
  using namespace std;

  int main () {
    ios::sync_with_stdio (false); // C++ 的原生 iostream 很吃效能，加上這三行能避免這個問題
    cin.tie (0);
    cout.tie (0);
    . . .

    int m = INT_MAX, n = INT_MIN; // 最大整數 (2,147,483,647) 與最小整數 (-2,147,483,648)
    . . .

    y = abs(x); // y 被指定為 x 的絕對值
    . . .

    int x, y, m, n;
    . . .
    m = max (x, y); // m 被指定為 x 和 y 之中較大的值
    n = min (x, y); // m 被指定為 x 和 y 之中較小的值
    . . .

    int n;
    . . .
    cin >> n;
    int a [n];
    int mn = INT_MAX;
    for (int i = 0; i < n; i++)
        mn = min (a[i], mn);

    // 至此，mn 會是 a [n] 中的最小值
    . . .

```

如果用 if 寫

```

// if (mn > a[i])
//     mn = a[i];

```

```

int x, y;
x = 2;
y = 3;
swap (x, y); // x 和 y 的值交換 // int z = x; x = y; y = z;
cout << x << " " << y << endl; // 會輸出: 3 2

```

```

int n;
. . .
cin >> n;
int a [n];
. . .
sort (a , a + n ); // a [0] ~ a [n - 1] (n 個) 由小到大排序

```

```

// 或者
sort (a + i, a + j + 1); // a [i] ~ a [j] 由小到大排序
. . .

```

```

double x;
int m;

m = round (x); // 指定 m 為最接近 x 的整數
m = floor (x); // 指定 m 為不大於 x 的整數
m = ceil (x);  // 指定 m 為不小於 x 的整數
. . .

```

```

}

```

```

int main () {
    int a [] = { 2, 4, 3, 1 };
    sort (a, a + 4);
    // 用迴圈逐一輸出 a []
    . . .
    return 0;
}

```

```

int main () {
    char a [] = "hello";
    sort (a, a + 5);
    cout << a << endl;
    return 0;
}

```

目 檢測內容

科目	說明
程式設計觀念題	兩份測驗題本共計40個試題，分兩節次施測 (平均一題 3 分鐘 → 在出題時就決定好哪 25 題計分，一題 4 分，剩下 15 題是用來做試題分析、評測試題的鑑別度的參考)
程式設計實作題	一份測驗題本共計4個題組 (共 150 分鐘)

① 重要時程

觀念題與實作題	觀念題	實作題
---------	-----	-----

科目	節次	入場時間	檢測起始	提前交卷 起始時間	提前交卷 截止時間	檢測結束
觀念題	第一節	09:30	09:40	10:10	10:30	10:40
	第二節	11:00	11:10	11:40	12:00	12:10
實作題	第三節	13:30	13:40	14:10	16:00	16:10

檢測成績級別說明

2021/1/9 APCS官方成績查詢中給的成績統計數據

程式設計觀念題 檢測人數2998人			程式設計實作題 檢測人數2962人		
級別	原始總分範圍	百分比*	級別	原始總分範圍	百分比*
五	90~100	2.1	五	350~400	0.9
四	70~89	13.5	四	250~349	2.2
三	50~69	26.8	三	150~249	20.5
二	30~49	36.1	二	50~149	35.1
一	0~29	21.5	一	0~49	41.3

* 該次檢測人數百分比 (四捨五入取概數到小數第一位)

檢測成績級別說明

2020/10/17 APCS官方成績查詢中給的成績統計數據

程式設計觀念題 檢測人數2334人			程式設計實作題 檢測人數2265人		
級別	原始總分範圍	百分比*	級別	原始總分範圍	百分比*
五	90~100	7.8	五	350~400	0.9
四	70~89	33.1	四	250~349	2.8
三	50~69	34.1	三	150~249	16.5
二	30~49	18.6	二	50~149	46.3
一	0~29	6.5	一	0~49	33.6

* 該次檢測人數百分比 (四捨五入取概數到小數第一位)

檢測成績級別說明

2020/07/04 APCS官方成績查詢中給的成績統計數據

程式設計觀念題 檢測人數1583人			程式設計實作題 檢測人數1585人		
級別	原始總分範圍	百分比*	級別	原始總分範圍	百分比*
五	90~100	3.1	五	350~400	0.8
四	70~89	17.8	四	250~349	1.7
三	50~69	34.1	三	150~249	19.0
二	30~49	30.6	二	50~149	29.7
一	0~29	14.4	一	0~49	48.8

* 該次檢測人數百分比 (四捨五入取概數到小數第一位)

檢測成績級別說明

2020/01/05 APCS官方成績查詢中給的成績統計數據

程式設計觀念題 檢測人數2141人			程式設計實作題 檢測人數1931人		
級別	原始總分範圍	百分比*	級別	原始總分範圍	百分比*
五	90~100	2.1	五	350~400	0.8
四	70~89	12.6	四	250~349	3.6
三	50~69	27.4	三	150~249	11.0
二	30~49	34.7	二	50~149	26.0
一	0~29	23.3	一	0~49	58.6

* 該次檢測人數百分比 (四捨五入取概數到小數第一位)

```
#include <bits/stdc++.h>
using namespace std;

int main () {
    char A [] = "1234567";    // char A [] = { '1', '2', '3', '4', '5', '6', '7', 0 };
    cout << strlen (A) << endl; // 回傳 A 內字元串之長度
    if (isdigit (A [2]))      // 若 A [2] 為數字，回傳真
        cout << "A [2] is a digit." << endl;
    if (isalpha (A [2]))      // 若 A [2] 為英文字母，回傳真
        cout << "A [2] is an alphabet." << endl;
    int x = atoi (A);        // 將字元陣列 A 裡的數字順序視為一個十進位數整數，回傳對應的整數
    cout << x << endl;
    return 0;
}
```

若萬能標頭的 bits/stdc++.h 找不到：

```
#include <cstring>    ... strlen()
#include <climits>    ... INT_MAX, INT_MIN
#include <cstdlib>    ... abs()
#include <algorithm> ... swap(), max(), min(), sort()
```


Python	C++ / STL	Java / the Java Collections
(script, interpretive)	(compiled)	(bytecodes)
<code>print (3//2)</code>	<code>cout << 3/2 << endl;</code>	<code>System.out.println (3/2);</code>
<code>x, y = map (int, input().split())</code>	<code>int x, y; cin >> x >> y;</code>	<code>import java.util.Scanner; ... Scanner input = new Scanner (System.in); int x = input.nextInt (), y = input.nextInt ();</code>
<code>if ...: ... elif ...: ... else: ...</code>	<code>if (...) { ...; } else if (...) { ...; } else { ...; }</code>	(same as C++)
<code>... if cond else ...</code>	<code>cond ? ... : ...;</code>	(same as C++)
<code>and, or</code>	<code>&&, </code>	(same as C++)
<code>while ...: ...</code>	<code>while (...) { ...; }</code>	(same as C++)
<code>for ... in ...: ...</code>	<code>for (auto ...: ...) { ...; }</code>	-
<code>for i in range (n): ...</code>	<code>for (i = 0; i < n; i++) { ...; }</code>	(same as C++)
<code>for i in range (s, t): ...</code>	<code>for (i = s; i < t; i++) { ...; }</code>	(same as C++)
<code>for i in range (s, t, inc): ...</code>	<code>for (i = s; i < t; i += inc) { ...; }</code>	(same as C++)
-	<code>int a [8] = { 0 };</code>	<code>int[] a = new int [8]; // 語言規範保證全 0</code>
-	<code>int a [] = { 1, 2, 3, 4 };</code>	<code>int[] a = { 1, 2, 3, 4 }; // int[] a = new int []{1, 2, 3, 4};</code>
<code>x = lambda a : a + 10</code>	<code>auto x = [](int a) { return a + 10; };</code>	-

Python	C++ / STL	Java / the Java Collections
<pre>for c in s: ...</pre>	<pre>for (auto c: s) { ...; }</pre>	<pre>for (int i = 0; i < s.length (); i++) { char c = s.charAt (i); ...; }</pre>
list	<u>vector</u>	ArrayList: List
x = [1, 2, 3] # 中括弧, 方括弧	vector <int> x = { 1, 2, 3 };	ArrayList<int> x = newArrayList<int>(); x.add (1); x.add (2); x.add (3);
m = x [1]	auto m = x [1];	int m = x.get(1);
n = len (x)	auto n = x.size();	int n = x.size();
x.append (4)	x.push_back (4);	x.add (4);
x.pop (1)	x.erase (1);	x.remove (1);
x.remove (3) # removes the 1st occurrence of 3	(use find)	int i = x.indexOf (3); if (i != -1) x.remove ();
x.clear()	x.clear(); // x.resize(0);	x.clear();
y = x.copy()	auto y = x; // auto y (x);	ArrayList<int> y = newArrayList<int>(x);
z = x + [4, 5, 6]	vector <int> w = { 4, 5, 6 }; auto z (x); // auto z = x; z.insert (z.end(), w.begin(), w.end());	ArrayList<int> z = newArrayList<int>(x); ... z.addAll (w);
c = x.count (2)	int c = count (x.begin(), x.end(), 2); // or auto is2 = [] (int p) { return p == 2; }; int c = count_if (x.begin(), x.end(), is2);	Collections.frequency (x, 2);
i = x.index (2)	int i = find (x.begin(), x.end(), 2) - x.begin(); // or find_if	int i = x.indexOf (2);
x.insert (1, 5)	x.insert (x.begin() + 1, 5); // returns x.begin() + 1	x.add (1, 5);

Python	C++ / STL	Java / the Java Collections
<code>x.reverse()</code>	<code>reverse (x.begin(), x.end());</code>	<code>Collections.reverse (x);</code>
<code>x.sort()</code>	<code>sort (x.begin(), x.end());</code>	<code>Collections.sort (x);</code>
<code>mx = max (x)</code>	<code>int mx = *max_element (x.begin(), x.end())</code>	<code>int mx = Collections.max (x);</code>
<code>mn = min (x)</code>	<code>int mx = *min_element (x.begin(), x.end())</code>	<code>int mn = Collections.min (x);</code>
<code>sm = sum (x)</code>	<code>int sm = accumulate (x.begin(), x.end(), 0)</code>	<code>int sm = MathUtils.sum (x);</code>
<code>x = (1, 2, 3)</code> # 小括弧，圓括弧	<code>tuple, pair</code>	
<code>dict</code>	<code>map</code>	<code>Map</code>
<pre> x = { "a": 23, "b": 56, "c": 89 } # key: value # or x = dict (a=23, b=56, c=89) # or ks = ('a', 'b', 'c') x = dict.fromkeys (ks) x ["a"] = 23 x ["b"] = 56 x ["c"] = 89 </pre>	<pre> map <string, int> x = { { "a", 23 }, { "b", 56 }, { "c", 89 } }; </pre>	<pre> Map <String, int> x = new HashMap<>(); x.put("a", 23); x.put("b", 56); x.put("c", 89); </pre>
<code>v = x ["b"]</code> # <code>k = x.get ("b")</code>	<code>int v = x.find ("b")->second;</code>	<code>int v = x.get ("b");</code>
<code>x ["d"] = 100</code>	<pre> x ["d"] = 100; // x.insert (make_pair ("c", 100)); </pre>	<code>x.put ("d", 100);</code>

Python	C++ / STL	Java / the Java Collections
for k in x: # for k in x.keys () ...	for (auto it = x.begin(); it != x.end(); it++) { string k = it->first; int v = it->second; ...; }	for (String k : x.keySet ()) { ...; }
for v in x.values(): ... # or for k in x: v = x [k] ...		for (int v : x.values ()) { ...; }
for k, v in x.items(): # each pair ...		for (Map.Entry<String, int> e: x.entrySet()) { String k = e.getKey (); int v = e.getValue (); ...; } // or x.forEach (k, v) -> { ... };
n = len (x)		int n = x.size();
x ["c"] = 111213 # x.update ({ "d": 1112113 })	x ["c"] = 111213;	x ["c"] = 111213;
x.pop ("b") # del x [1]	x.erase (x.find ("b"));	x.remove ("b");
x.popitem()		
x.clear() # <u>vs.</u> del x	x.clear();	x.clear();
y = x.copy() # y = dict (x)	auto y = x; // auto y (x);	Map <String, int> y = newHashMap<>(x);
v = x.setdefault ("b", 10)		
z = { "e": { "f": 3, "g": 5 }, \ "h": { "i": 8, "j": 10 } } # or p = { "f": 3, "g": 5 } q = { "i": 8, "j": 10 } z = { "e": p, "h": q }	map <string, map <string, int>> z =

Python	C++ / STL	Java / the Java Collections
set	set (also set_... methods on vectors, arrays)	Set
x = { 1, 2, 3 } # x = set ((1, 2, 3))	set <int> x = { 1, 2, 3 };	Set<int> x = new HashSet<>(); x.add(1); x.add(2); x.add(3);
if (3 in x): ...	if (x.find (3) != x.end()) { ...; }	if (x.contains (3)) { ...; }
x.add (4) # x.update ([4, 2, 6])	x.insert (4); set <int> w = { 4, 2, 6 };	x.add (4);
x.update ({ 4, 2, 6 })	// int w [] = { 4, 2, 6 }; x.insert (w.begin(), w.end());	... x.addAll (w);
y = x.union ({ 4, 2, 6 })	set <int> y; y.reserve (x.size() + w.size()); auto it = set_union (x.begin(), x.end(), w.begin(), w.end(), y.begin()); y.resize (it - y.begin());	Set<int> y = new HashSet<>(x); y.addAll (w);
n = len (x)	int n = x.size();	int n = x.size();
x.remove (3) # <u>vs.</u> x.discard (3)	x.erase (x.find (3));	x.remove (3);
x.pop()		
x.clear() # <u>vs.</u> del x	x.clear();	x.clear();
y = x.copy() # y = set (x)	auto y = x; // auto y (x);	Set<int> y = new HashSet<>(x);
z = x.difference (y)	set <int> z; z.reserve (x.size()); auto it = \ set_difference (x.begin(), x.end(), y.begin(), y.end(), z.begin()); z.resize (it - z.begin());	Set<int> z = new HashSet<>(x); z.removeAll (y);
x.difference_update (y)		x.removeAll (y);
z = x.intersection (y)	set <int> z; z.reserve (x.size() + w.size()); auto it = set_intersection (x.begin(), x.end(), y.begin(), y.end(), z.begin()); z.resize (it - z.begin());	Set<int> z = new HashSet<>(x); z.retainAll (y);
x.intersection_update (y)		x.retainAll (y);

Python	C++ / STL	Java / the Java Collections
<code>z = x.symmetric_difference (y)</code>	<code>set <int> z; z.reserve (x.size() + w.size()); auto it = set_symmetric_difference \ (x.begin(), x.end(), y.begin(), y.end(), z.begin()); z.resize (it - z.begin());</code>	(union - intersection)
<code>x.symmetric_difference_update (y)</code>		(union - intersection)
<code>if (x.isdisjoint (y)) ...</code>		
<code>if (x.issubset (y)) ...</code>	<code>if (includes (y.begin(), y.end(), x.begin(), x.end())) { ...; }</code>	<code>if (x.contains (y) { ; }</code>
<code>if (x.issuperset (y)) ...</code>	<code>if (includes (x.begin(), x.end(), y.begin(), y.end())) { ...; }</code>	<code>if (y.contains (x) { ; }</code>
	<u>deque</u>	Deque
	list	
	priroity_queue (top() at right)	
	stack	Stack: List
	queue	Queue

Python:

```
df = pandas.DataFrame ()  
df [c]: column as pandas.Series (c as some column id in quotes)  
df.loc [r]: row as pandas.Series (r as some row id in quotes)  
df.at [r, c]: element  
df [i:j]: rows equivalent to df.iloc [i:j] but df [i] illegal (i, j as int numbers)  
  
dict.items ()  
df.DataFrame.iteritems ()  
df.DataFrame.iterrows ()
```

Java:

primitive data type: byte, short, int, long, float, double, boolean and char