**Vector**

**可用的建構元:**

Vector<型別> 變數名;

Vector<型別> 變數名2(已經存在的變數名1);// copy constructor

Vector<型別> 變數名2(變數名1.begin(), 變數名1.end());

Vector<型別> 變數名1(n); // 創建n個元素的vector

Vector<型別> 變數名1(n, val); // 預設為n個val

**常用的method:**

Begin(): 回傳第一個元素的iterator

End(): 回傳最後一個的再後面一個的iterator

Size(): 回傳整個vector元素的個數

Empty(): 回傳vector是否為空

Front(): 回傳第一個元素

Back(): 回傳最後一個元素

Push\_back(x): push x，x會為最後一個元素

Pop\_back(): pop 最後一個元素

Capacity(): 回傳當前使用的(已分配)vector的容量數，大於等於size

max\_size(): 回傳當前vector能夠容納的所有容量數(未被分配的)

rbegin(): 回傳vector reverse後的第一個元素的iterator

rend(): 回傳vector reverse後的最後一個元素的後一個的iterator

resize(n, val): 調整vector的容量為n，如果當前的size大於n則，多出來的銷毀；反之，則利用val當預設值做初始化。

Insert(): 在指定iterator前(或範圍內)插入元素做初始化

Erase(): 移除一個位置或一段範圍內的元素

Clear(): 清除vector內所有的元素，容量歸零

**List**

**可用的建構元:**

list<型別> 變數名;

list <型別> 變數名2(已經存在的變數名1);// copy constructor

list <型別> 變數名2(變數名1.begin(), 變數名1.end());

list <型別> 變數名1(n); // 創建n個元素的list

list <型別> 變數名1(n, val); // 預設為n個val

**常用的method:**

Size(): 回傳整個list的元素的個數

Empty(): 回傳vector是否為空

Push\_back(x): push x，x會為最後一個元素

Push\_front(x): 在list的頭增加一個 x

Pop\_back(): pop 最後一個元素

Pop\_front(): pop 第一個元素

Remove(x): 刪除list中所有元素質為x的元素

Clear(): 清除list內所有的元素，容量歸零

insert(iterator it, const T& x ) 在it的位址前插入x，並回傳x的it

insert(iterator it,size\_type n,const T& x): 迭代器指針it前插入n個相同元素x

insert(iterator it,const\_iterator first, const\_iteratorlast): 把(first,last)間的元素插入迭代器指針it前

Begin(): 回傳第一個元素的iterator

End(): 回傳最後一個的再後面一個的iterator

Front(): 回傳第一個元素

Back(): 回傳最後一個元素

rbegin(): 回傳vector reverse後的第一個元素的iterator

rend(): 回傳vector reverse後的最後一個元素的後一個的iterator

erase(iterator it): 刪除迭代器指針it對應的元素，回傳iterator

erase(iterator first,iterator last): 刪除迭代器指針(first,last)間的元素，回傳iterator

sort(): list內所有元素排序，預設是升序(1,2,3…)

unique(): list內相鄰元素若有重複的，則僅保留一個

splice(iterator it,list& li): list的合併函數，list li插入迭代指針it前，li變成空list

splice(iterator it, list& li ,iterator first): list li中把(first,end)間的元素插入迭代指針it前

splice(iterator it, list& li, iterator first, iterator last): last中移走(first,last)間元素插入迭代器指針it前

reverse(): 反轉list中元素順序

**Map**

**可用的建構元:**

Map <key的元素, value的元素>變數名稱;

Map <key的元素, value的元素>變數名稱2(變數名稱1);

或

map<int, string> studentMap = {

{1, "Tom"},

{2, "Jack"},

{3, "John"}

};

**常用的method:**

插入 insert()範例:

mapStudent.insert(pair<string, string>("r000", "student\_zero"));

or

mapStudent["r123"] = "student\_first";

mapStudent["r456"] = "student\_second";

尋找 find():  
出現時，它返回資料所在位置，如果沒有，返回 iter 與 end() 函數的返回值相同。

clear() : 清空 map 中的資料

empty() : 回傳是否有為空

erase()範例:

//迭代器刪除

iter = mapStudent.find(**"r123"**);

mapStudent.erase(iter);

//用關鍵字刪除

**int** n = mapStudent.erase(**"r123"**);//如果刪除了會返回1，否則返回0

//用迭代器範圍刪除 : 把整個map清空

mapStudent.erase(mapStudent.begin(), mapStudent.end());

//等同於mapStudent.clear()

**SET**

**可用的建構元:**

set<型別> 變數名稱;

set<型別> 變數名稱2(變數名稱1);

set<string> myset{"123", "456", "789"};

**常用的method:**

myset.insert (25); //插入25

myset.insert (it,25); //在it前插入25

begin() : 返回指向第一個元素的迭代器

clear() : 清除所有元素

count(): 返回某個值元素的個數

empty(): 如果集合為空，返回true

end(): 返回指向最後一個元素之後的迭代器，不是最後一個元素

erase(): 刪除集合中的元素

find() : 返回一個指向被查找到元素的迭代器

insert(): 在集合中插入元素

max\_size(): 返回集合能容納的元素的最大限值

rbegin(): 返回指向集合中最後一個元素的反向迭代器

rend(): 返回指向集合中第一個元素的反向迭代器

size(): 集合中元素的數目

swap(): 交換兩個集合的元素

**Stack**

**可用的建構元:**

Stack<型別> 變數名稱;

**常用的method:**

Top(): 回傳最後push的元素

POP(): pop 最後進入stack的元素

Push(x): push x 進入stack

size(): stack中元素的數目

empty(): 如果stack為空，返回true

swap(): 交換兩個stack的變數

**Queue**

**可用的建構元:**

queue<型別> 變數名稱;

**常用的method:**

back (): 回傳最後進來的元素

front(): 回傳第一個進來的元素

POP(): pop 最後進入queue的元素

Push(x): push x 進入queue

size(): stack中元素的數目

emplace(): 用於將新元素插入queue，並將新元素添加到queue的末尾

empty(): 如果queue為空，返回true

swap(): 交換兩個queue的變數