

# Homework 002 v3

更正了第四  
頁的內容

CONTAINER IS GOOD!

2021/4/2 FRI. 12:00 NOON DUE

2021/4/6 TUE. 23.59 為最後補交期限

# Goal – 1 (dynamic array)

```
// 程式執行後首先讓人輸入亂數種子
```

```
Input random seed: 35
```

```
Allocate 5 slots
```

```
0 0 0 0 0
```

```
Push 3
```

```
3 0 0 0 0
```

```
Push 5
```

```
3 5 0 0 0
```

```
Pop
```

```
3 0 0 0 0
```

```
Resize to 3 slots
```

```
3 0 0
```

```
Push 5
```

```
3 5 0
```

```
Push 3
```

```
3 5 3
```

```
Push 2
```

```
3 5 3 (Push Fail!)
```

```
Pop ...
```

程式執行後首先讓人輸入亂數種子。。

程式接著隨機生成一個大小為1~5的容器，然後隨機進行push pop 或是resize

程式執行後不需要額外輸入並隨機輸出。每次輸出皆不一樣，範例輸出格式如左。

出現push / pop fail程式仍可正確執行。



# Operation of outputs

- ▶ Allocate **N** slots – 配置**N**個空間的容器（僅發生在首行）
- ▶ Push **J** – 放入**J**物件, **J > 0** 為整數
  - ▶ 1 3 5 0 0 0 >>> 1 3 5 **J** 0 0
  - ▶ 1 3 5 9 4 3 >>> 1 3 5 9 4 3 (Push Fail!)  
//**J**物件消失，容器維持原樣
- ▶ Pop – 放入**最上面一個**物件
  - ▶ 1 3 **5** 0 0 0 >>> 1 3 0 0 0 0
  - ▶ 0 0 0 0 0 0 >>> 0 0 0 0 0 0 (Pop Fail!) //維持原樣
- ▶ Resize to **M** slots – 容器配置後發生，容器容量變成**M**個
  - ▶ 縮小 – 內容物丟失（Container不需重新配置空間）
  - ▶ 放大 – 內容物補零
    - ▶ 放會需要觸發重新配置空間

如何事後列印出錯誤訊息？  
\*請事先把錯誤訊息存在容器中！

\*另外儲存容器大小

# Goal – 2 (operators)

// 確保下面這段程式碼可以動！

Container A(3), B(5); //容器A,B可為隨機大小

cin >> A >> B; //使用者依序輸入A與B，-1表示後面沒有了

cout << A << B << A + B;

cout << (A = A + 3);

Container C = A + B;

cout << C;

C = A + C + B;

cout << C + 4;

cout << Container::cat(A,B);

完成Goal 1之後  
執行這段程式碼

若 A使用者輸入為3 2 -1 B為1 4 5 -1

3 2 0 //A

1 4 5 0 0 //B

3 2 1 4 5 0 0 0 //A+B

3 2 3 //A=A+3

3 2 3 1 4 5 0 0 //C=A+B

3 2 3 3 2 3 1 4 5 1 4 5 4 0 0 0 //C=A+C+B; C+4;

3 2 3 1 4 5 0 0 //Container::cat(A,B);

更正

更正

# The Container Class (review)

可自行追加函式、變數

```
class Container
{
    ...operator<<(...) //輸出內容物+換行（不含錯誤訊息）
    ...operator>>(...) //讓使用者逐次輸入所有物件 -1表示後面沒有了
public:
    Container( int ); //建立空容器（輸入值為容器大小）
    void push( int );
    int pop(); //移除最後放入的並回傳移出物（沒有則為0）
    void printALL(); //印出所有內容，含錯誤訊息+換行。
    void resize(... int ); //改變容器大小
    static ??? cat(???,???) //合併Container，回傳合併好的
    ??? operator+(... Container) //C = A+B 功能同cat,但A與B不變
    ??? operator+(... int) //C = A + 3功能同push，但A不變
    //你鐵定還需要其他東西（如Copy constructor, operator= ...）
    //各函式的const版又該如何？
private:
    //用動態鎮列該怎麼儲存呢？？？
};
```



# Notes

- ▶ 輸出格式應特別注意、行為亦應符合規範
  - ▶ 為了測試需要，建議練習時亂數種子可以採手動輸入
- ▶ 正確的動態配置、回收資源是必要的
- ▶ 各參數要多用reference作為輸入、回傳則要特別小心
- ▶ Operator的輸入與回傳、應該寫成Global / Member Function、如何對應Const物件應特別注意
- ▶ operator=, Destructor & Copy Constructor都不要忘了自己實做，或許你還會需要operator==
- ▶ 必須要可適應助教額外提供之Driver program
  - ▶ 越周延越多分～

# The Requirements

- ▶ 主程式應區分為main function, 與Container 兩部分，  
總共區分為三個檔案
  - ▶ s1234567\_container.h .cpp
  - ▶ s1234567\_main.h
- ▶ 隨機性
  - ▶ 初始容器的尺寸為1-5個隨機、改變後尺寸1-10隨機
  - ▶ 容器內容皆為1-10之整數
  - ▶ 每次產生的數字皆隨機、行為隨機
  - ▶ 每次隨機15-20個操作
- ▶ 容器採用動態記憶體配置
- ▶ 程式需要「可以編譯執行」助教才會批改





# 請自己寫Driver Program !

在滿足老師的要求後，盡可能多多測試～



# 其他補充 – 前處理器、隨機

```
#pragma once //使得該檔案只會被包含一次前處理器  
//取代 #ifndef + #define + #endif
```

---

rand 隨機產生數值

<https://www.cplusplus.com/reference/cstdlib/rand/>

```
#include <stdlib.h> //srand(), rand()  
#include <time.h>   //time()
```

```
srand (time(NULL)); //用當下時間設定亂數種子  
rand() % 10 + 1;     //產生1到10的數字  
rand() % 30 + 1985;  //產生1985-2014
```

---

你也可以使用<random>

<https://www.cplusplus.com/reference/random/>

# 其他補充 – 動態記憶體配置

**new / delete** 動態記憶體配置

<https://www.cplusplus.com/reference/new/>

```
int* x = new int[10]; //宣告容量為10之動態陣列
x[0] = 3; //使用
x[1] = 4;
x[9] = 5;

delete[] x; //刪除此動態陣列
```

# 動態記憶體使用範例

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      cout << "hello world!" << endl;
7      int* x = new int[10];
8      x[0] = 3;
9      x[1] = 4;
10     x[9] = 5;
11
12     delete[] x;
13     x[3] = 5;
14     return 0;
15 }
```

擲回例外狀況

擲回例外狀況: 寫入存取違規。  
x 為 0x8123。

[複製詳細資料](#)

▲ 例外狀況設定

- ☒ 於擲回這個例外狀況類型時中斷
- 屬於例外的擲回來源:
  - ☐ 1092Hw1ANS.exe

[開啟例外狀況設定](#) | [編輯條件](#)



# Submit your homework

- ▶ 請再次確定你的程式碼可以編譯了
  - ▶ 無法編譯此次將無法批改（0分）
  - ▶ 助教會嘗試通知你，並請你補交
- ▶ Please use s1234567\_Container.h .cpp and s1234567\_Main.cpp as your file names.
  - ▶ Replace s1234567 by your own student ID
  - ▶ Pack all the file into a folder named “s1234567\_hw2”
  - ▶ Zip it and Upload to Portal
  - ▶ Attach a s1234567\_hw2.txt if u want to add some additional information to TA

本次不用  
DEMO喔！

任何格式不符、多上傳檔案或其他造成助教困擾的狀況一律扣10分