第二次期中考

第一題

請以分開三個檔案方式設計一類別 Employee
按照題目所需設計適當的 constructor
data fields (**private**) 有 **name:string**、**income:int**、**hours:int**
function 有 **setName**、**setIncome**、**setHours**、**getName**、**getIncome**、**getHours**、**getWage**
getWage 函式為計算員工時薪，即日薪(income)除以每日上班時數(hours)
另外，請以 **this** 方式設計再加入兩個 function，分別為 **swapByReference** 與 **swapByPointer**

函式名稱與形式為：
Employee::swapByReference(Employee &employee2)
Employee::swapByPointer(Employee *employee2)

輸入說明：
請依序輸入兩組 name、income、hours 建立兩個 Employee 物件。

輸出說明：
將兩物件利用 pass-by-reference 函式以及 pass-by-pointer 函式，以 this 方式進行交換，最後依照範例輸出的格式進行輸出。

範例輸入：
Candy 10000 12
Superstar 13000 7

範例輸出(以時薪輸出)：
SwapByReference:Candy-833.33 Superstar-1857.14 to Superstar-1857.14 Candy-833.33
SwapByPointer:Superstar-1857.14 Candy-833.33 to Candy-833.33 Superstar-1857.14

※ 時薪請計算至小數點後兩位

第二題

在 queue.h 中已有完整個 Queue (佇列) 程式
請先將該程式修改為 template 形式接納各種型別
再於主程式利用此類別，完成依序將字元 push 到 Queue 中，並用 pop 取出
運算式為後序運算(Postfix)，也就是**將符號前的兩數字進行該符號之運算**
運算後將結果再 push 回 Queue
Queue 的特性為**先進先出** (FIFO - First In First Out)

佇列類別中函式：
isEmpty() 判斷佇列是否為空值
isFull() 判斷佇列是否填滿
getFront() 取得佇列最前值
getBack() 取得佇列最後值
push(value) 將 value 推入佇列中最後
pop() 將最前值從佇列中移除
getSize() 取得佇列大小
ensureCapacity() 若佇列額滿將會增加佇列大小

輸入說明：
於主程式輸入 7 個字元 push 到 Queue，其中包含數字字元**(皆為個位整數)**與運算符號字元。

輸出說明：
每次用 pop 取出三個字元後，將前兩個數字以後面的符號運算(進行字元處理後運算)，並將結果輸出後再 push 回 Queue。重複此動作直到 Queue 的大小剩一個值為止。
請注意：後序運算符號前必須為兩數字，假設起始值為符號可直接 push 到 Queue 的最後，並再做一次運算。

範例輸入：
33*12+-

範例輸出：
9
3
6

※ 請不需考慮任何例外狀況

第三題

請利用**一維 vector** 建立兩個的**整數**與**浮點數**矩陣

並依序輸入 12 個整數與 12 個浮點數分別依序存入相對應的矩陣中

先將矩陣進行排序後輸出

接著使用 template 形式撰寫一函式接納各種型別

T max(vector<T>)

使用此函式找出 vector 中的最大值

輸入說明：

輸入 12 個整數與 12 個浮點數並分別依序存入相對應的矩陣中。

輸出說明：

先將兩個矩陣進行排序，並分別透過 max 函式找出兩個矩陣的最大值，最後依照範例輸出。

範例輸入：

75 55 4 44 31 81 98 75 81 86 54 40

7.5 5.5 4.0 4.4 3.1 8.1 9.8 7.5 8.1 8.6 5.4 40.0

範例輸出：

Integer vector sort:

4 31 40 44 54 55 75 75 81 81 86 98

Integer vector maximum value: 98

Decimal vector sort:

3.1 4.0 4.4 5.4 5.5 7.5 7.5 8.1 8.1 8.6 9.8 40.0

Decimal vector maximum value: 40.0

※ 矩陣排序不限定排序方式，而輸出時，數字與數字間以一個 space 分隔即可。

第四題

請以分開三個檔案方式設計一類別 Complex

根據需求建立適當的建構子

data fields(private): intNumber:int、complexNumber:int

三個 **function:**

Complex add(Complex) -> (a+bi)+(c+di) = ((a+c)+(b+d)i)

Complex subtract(Complex) -> (a+bi)-(c+di) = ((a-c)+(b-d)i)

String toString() -> 輸出格式 (a+bi)

輸入說明：

主程式輸入 6 整數，建構 3 組物件 comp1, comp2, comp3。

輸出說明：

請以 function 方式計算 comp1-comp2+comp3 並輸出。

(comp1.subtract(comp2)).add(comp3)

範例輸入：

1 0 -3 1 -4 3

範例輸出：

(1+0i)-(-3+1i)+(-4+3i)=(0+2i)

※ Complex 輸出格式中間一律以『+』表示。

第五題

承上題，請以分開三個檔案方式設計一類別 Complex

根據需求建立適當的建構子

data fields(private): intNumber:int、complexNumber:int

三個 function:

Complex add(Complex) -> (a+bi)+(c+di) = ((a+c)+(b+d)i)

Complex subtract(Complex) -> (a+bi)-(c+di) = ((a-c)+(b-d)i)

String toString() -> 輸出格式 (a+bi)

接著完成**+(正)**, **-(負)**, **+(加法)**, **-(減法)**, **<<(輸出)** 的 operator overloading

輸入說明：

主程式輸入 6 整數，建構 3 組物件 comp1, comp2, comp3。

輸出說明：

請以 operator overloading 方式計算 comp1-comp2+comp3 並輸出。

範例輸入：

1 0 -3 1 -4 3

範例輸出：

(1+0i)-(-3+1i)+(-4+3i)=(0+2i)

※ 本題請務必以 operator 方式計算，切勿再使用 (comp1.subtract(comp2)).add(comp3) 計算。