程式設計（105-2）
作業八

作業設計：孔令傑
國立臺灣大學資訊管理學系

繳交作業時，請至 PDOGS（`http://pdogs.ntu.im/judge/`）為第一、二題上傳一個 PDF 檔，再為第三題至第五題各上傳一份 C++ 原始碼（以複製貼上原始碼的方式上傳）。第四題跟第五題都是 bonus 加分題。每位學生都要上傳自己寫的解答。不接受紙本繳交；不接受遲交。請以英文或中文作答。

這份作業的截止時間是 ***2017 年 4 月 24 日下午兩點***。在你開始前，請閱讀課本的第 22.1–22.6 節[1]。為這份作業設計測試資料並且提供解答的助教是林敬傑（Jack Lin）。

第一題

（20 分）請在 PDOGS 上批改你被隨機分配到的兩份作業的程式碼：

(a)（20 分）作業六第三題。

說明：這一題的分數算在這份作業。

(b)（0 分）作業七第三題。

說明：這一題的分數算在作業七。

第二題

（20 分）在第七份投影片（講 C string 的）的第 47 頁有如下的程式：

```
char a[100] = "this is a book";
char* p = strstr(a, "is");
while(p != nullptr)
{
  strcpy(p, "IS");
  p = strstr(p, "is");
}
cout << a;
```

請說明這段程式碼「有沒有」成功地把 `a` 裡面的小寫 is 都改成大寫 IS，如果有，為什麼嘗試印出 `a` 時不會看到兩個大寫 IS，而如果沒有又是為什麼。請特別注意第 43 頁的程式碼是可以的。

[1]課本是 Deitel and Deitel 著的 *C++ How to Program: Late Objects Version* 第七版。

第三題

（40 分）在這份作業的第三至五題，我們將討論亂數產生器（random number generator），特別是如何衡量一個亂數產生器的品質，也就是「產生的亂數是否夠亂」。我們將在這三題中分三個步驟來討論這個議題。

給定 n 個整數 $x_1 \le x_2 \le \cdots \le x_n$，我們定義其中位數「median」為

$$M = \begin{cases} x_{\frac{n+1}{2}} & \text{如果 } n \text{ 是奇數} \\ \dfrac{x_{\frac{n}{2}} + x_{\frac{n}{2}+1}}{2} & \text{如果 } n \text{ 是偶數} \end{cases}。$$

翻成白話文的話，就是如果有奇數個數字，就取中間那一個；如果有偶數個，就取中間那兩個的平均。例如五個數字 1、3、5、7 和 9 的中位數是 5，而六個數字 1、2、3、5、7 和 9 的中位數是 $\frac{5+7}{2} = 6$。顯然這表示一堆整數的中位數可能是小數（不過小數部份只可能是 0.5）。而如果給定的一堆數字是未被排序的，我們就把這堆數字排序之後再按照上面的定義去找他們的中位數。

在本題中，你將會被給定 n 個未排序的整數，而你的任務就是印出他們的中位數。

說明：大家當然可以先把這 n 個整數排序，然後再按照上面的定義把中位數找出來。不過既然我們只是要找中間的那一或兩個數字，當然是不需要排序的。大家可以試著想一個好的演算法、去網路上找找看，或者就練習一下寫排序也可以。事實上，在網路上很容易就可以查到用 C++ 實作的找中位數程式，不過我們也不禁止大家去找，畢竟練習在網路上找可用的資源，對程式設計師來說也是很重要的技能。相反地，我們要請大家練習一下字串處理（當然，字串處理的程式碼在網路上也有一大堆）。

輸入輸出格式

系統會提供許多筆測試資料，每筆測試資料裝在一個檔案裡。在每個檔案中，第一列存放兩個整數 n 跟 m，第二列起則每列存放 m 個整數 x_1、x_2 直到 x_n，直到 n 個整數都被列出為止。在第二列開始的每一列中，任兩個數字之間都用一個逗點隔開，而一列的結尾沒有逗點。已知 $1 \le n \le 1000$、$1 \le m \le 100$、$0 \le x_i \le 1000000$。

給定一筆測試資料，請輸出 x_1 到 x_n 這 n 個數字的中位數，如果中位數是整數則輸出一個整數，是小數則印出那個小數到小數點第一位。舉例來說，如果輸入是

```
10 4
6,7,4,1
5,1,7,1
10,9
```

則輸出應該是

```
5.5
```

你上傳的原始碼裡應該包含什麼

你的.cpp 原始碼檔案裡面應該包含讀取測試資料、做運算，以及輸出答案的 C++ 程式碼。當然，你應該寫適當的註解。針對這個題目，你***不可以***使用上課沒有教過的方法。

評分原則

這一題的其中 40 分會根據程式運算的正確性給分。PDOGS 會編譯並執行你的程式、輸入測試資料，並檢查輸出的答案的正確性。前 30 分由 15 筆測試資料判定分數，一筆測試資料佔 2 分；後 10 分由 5「組」測試資料判定分數，每一組裡面有若干筆測試資料，全對的話才能得到 2 分。

第四題

（40 分）我們知道所謂的「亂數」，在電腦裡其實都是用函數產生的，而且下一個亂數幾乎總是用前一個亂數算出來的，所以如果那個函數寫得不好，這批「亂數」事實上就會有某種規律，那就不夠亂了。舉例來說，假設我設定公式為

$$x_{i+1} = -x_i + 1，$$

則如果亂數種子 $x_0 = 2$，我們就會依序得到 -1、2、-1、2 一直循環下去，那這個「亂數產生器」當然就很爛了。

假設有一個亂數產生器根據亂數種子 x_0 產生了一批「亂數」x_1、x_n 一直到 x_n，我們怎麼知道這批數字到底「夠不夠亂」? 這當然有很多方法，在本題中我們將介紹一個（可能是）最簡單的方法：Wald–Wolfowitz run test（以下簡稱 run test）[2]。方法簡述如下。首先，給定依序產生的 x_1、x_n 一直到 x_n，我們先求出這些數字的中位數 M。接著我們定義

$$y_i = \begin{cases} 1 & \text{如果 } x_i > M \\ 0 & \text{如果 } x_i = M \\ -1 & \text{如果 } x_i < M \end{cases}$$

以產生 y_1、y_2 一直到 y_n 等一系列的值。接著我們把其值為零的 y_i 刪掉，得到數列 z_1、z_2 一直到 z_p（當然 $p \le n$），使得 $z_p \in \{-1, 1\}$。根據 $\{z_i\}_{i=1,\dots,p}$ 的值，我們把這 n 個值切成 k 份，稱為 k 個「run」，其中每個 run 都包含盡量多個同樣為負或同樣為正的連續數字。舉例來說：

- 如果 $x = (6, 7, 4, 1, 5, 1, 7, 1, 10, 9)$，則中位數為 5.5，因此 $y = (1, 1, -1, -1, -1, -1, 1, -1, 1, 1)$，$z$ 則跟 y 一模一樣，因此我們有 5 個 run（前兩個數字、第三到六個數字、第七個數字、第八個數字、最後兩個數字）。
- 如果 $x = (1, 1, 1, 2, 2, 2, 3, 3, 3, 4)$，則中位數為 2，因此 $y = (-1, -1, -1, 0, 0, 0, 1, 1, 1, 1)$，因此 $z = (-1, -1, -1, 1, 1, 1, 1)$，因此我們有 2 個 run。
- 如果 $x = (1, 0, -1, 0, 1, 0, -1, 0, 1, 0, -1)$，則中位數為 0，此時 y 跟 x 一模一樣，因此 $z = (1, -1, 1, -1, 1, -1)$，因此我們有 6 個 run。

很直觀地，如果 n 個數字「很亂」，那麼 run 數應該不太多也不太少，但如果這串數字不亂，則可能會有很少個或很多個 run。統計學家針對各種不同的情況（主要取決於 $\{z_i\}_{i=1,\dots,p}$ 中 1 跟 -1 個別的個數）計算了可接受的 run 數範圍[3]，因此給定一個亂數產生器，我們只要用它產生一串數字，然後對這

[2]有興趣的同學可以看一下 Wikipedia：`https://en.wikipedia.org/wiki/Wald\%E2\%80\%93Wolfowitz_runs_test`。為了定義題目方便，我們所描述的方法可能和某些網頁的說明略有差異。

[3]大家可以參考例如這個網頁：`http://www.real-statistics.com/statistics-tables/runs-test-table/`，去看看這個「範圍」。

串數字計算其 run 數，並看看我們得到的 run 數是否落在可接受的範圍內，就可以決定我們的亂數產生器是否夠亂了（當然不放心的話，可以多產生幾串）。

在本題中，我們將給你一串 n 個數字 x_1、x_2 一直到 x_n，請依序印出 $\{z_i\}_{i=1,\ldots,p}$ 中 1 的個數、-1 的個數，以及 run 數。

輸入輸出格式

本題的輸入方式跟第三題一樣。給定一筆測試資料，請依序輸出 $\{z_i\}_{i=1,\ldots,p}$ 中 1 的個數、-1 的個數，以及 run 數。兩個數字之間用一個空白鍵隔開。舉例來說，如果輸入是

```
10 4
6,7,4,1
5,1,7,1
10,9
```

則輸出應該是

```
5 5 5
```

如果輸入是

```
10 4
1,1,1,2
2,2,3,3
3,4
```

則輸出應該是

```
4 3 2
```

你上傳的原始碼裡應該包含什麼

你的.cpp 原始碼檔案裡面應該包含讀取測試資料、做運算，以及輸出答案的 C++ 程式碼。當然，你應該寫適當的註解。針對這個題目，你***不可以***使用上課沒有教過的方法。

評分原則

- 這一題的其中 20 分會根據程式運算的正確性給分。PDOGS 會編譯並執行你的程式、輸入測試資料，並檢查輸出的答案的正確性。前 15 分由 15 筆測試資料判定分數，一筆測試資料佔 1 分；後 5 分由 5「組」測試資料判定分數，每一組裡面有若干筆測試資料，全對的話才能得到 1 分。
- 這一題的其中 20 分會在作業九中被評定。在作業九中，我們會讓同學們互相檢視彼此的本題程式碼，並且就可讀性、易維護性、模組化程度、排版等面向寫評語和給評分（當然一切都是匿名的）。該任務在本題中會佔 20 分，其中 10 分取決於檢視你的程式碼的同學給你的分數（必要時助教會出來主持公道，請不用緊張），另外 10 分取決於你對同學的程式碼的評語和評分的合理性和

建設性。若你在本次作業中沒有寫這一題，那作業九自然沒有人能檢視你的程式碼，你也就得要損失這 10 分了。

第五題（bonus）

（20 分）現在給定一串數列，我們已經有能力判斷這串數列夠不夠亂了（如果我們去翻統計課本查表或代公式）。因此，給定一個亂數產生器，我們也可以判斷它夠不夠亂了。在本題中，我們將會讓你用下列公式

$$x_{i+1} = \mathrm{mod}(a + bx_i, c)$$

來產生一串亂數，其中 $\mathrm{mod}(u, v)$ 是 u 除以 v 的餘數。換言之，基於這個公式，我們可以稱呼一組參數 (a, b, c) 為一個亂數產生器。給定一個亂數產生器 (a, b, c)、一個亂數種子 x_0，以及一個整數 n，請將 x_0 代入公式以產生 x_1、x_2 一直到 x_n 的 n 個整數，然後按照第四題的定義，根據 $\{x_i\}_{i=1,\ldots,n}$ 產生 $\{y_i\}_{i=1,\ldots,n}$ 和 $\{z_i\}_{i=1,\ldots,p}$，接著印出 $\{z_i\}_{i=1,\ldots,p}$ 中 1 的個數、-1 的個數，以及 run 數。

本題的輸入格式如下。系統會提供許多筆測試資料，每筆測試資料裝在一個檔案裡。在每個檔案中，第一列存放五個整數 a、b、c、x_0 和 n，兩個整數之間用一個空白隔開。已知 $0 \le a \le 10000$、$1 \le b \le 10000$、$0 \le c \le 10000$、$0 \le x_0 \le c - 1$，以及 $1 \le n \le 1000$。輸出格式則跟第四題一樣。

針對這個題目，你***可以***使用任何方法。這一題的 20 分都根據程式運算的正確性給分。前 15 分由 15 筆測試資料判定分數，一筆測試資料佔 1 分；後 5 分由 5「組」測試資料判定分數，每一組裡面有若干筆測試資料，全對的話才能得到 1 分。