

程式設計（107-1）

作業三

作業設計：楊其恆、陳維漢、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/>) 為三題各上傳一份 C++ 原始碼（以複製貼上原始碼的方式上傳）；第三題的其中 20 分是加分題。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 **10 月 9 日凌晨一點**。在你開始前，請閱讀課本的第 3.6、3.10–3.12 和 4.1–4.11 節¹。為這份作業設計測試資料並且提供解答的助教是莊日陞。

第一題

（20 分；每題 4 分）針對以下五題是非題，我們會使用 PDOGS 自動批改，因此請寫一個 C++ 程式，內容就是先讀入一個整數，若讀入的數字為 i ，則印出第 i 小題的答案，若為是則印出 1、若為否則印出 0。舉例來說，如果題目只有四題，且你認為答案依序是是、否、是、是，則你上傳的程式碼應該是

```
#include <iostream>
using namespace std;

int main()
{
    int problem = 0;
    cin >> problem;
    if(problem == 1)
        cout << 1;
    else if(problem == 2)
        cout << 0;
    else if(problem == 3)
        cout << 1;
    else
        cout << 1;

    return 0;
}
```

PDOGS 會餵給你的程式的，一定是 1、2 直到 10 這十個整數。有別於作業中一般的程式題，本題在你上傳程式碼時，測試資料是還沒有放上 PDOGS 的，助教會等作業截止後才上傳測試資料（和答案）到 PDOGS 並重新批改此題。換言之，你上傳程式碼時是不會顯示你得幾分的，更不會顯示你對或錯哪些筆測試資料。你會看到你得 0 分，但此數字在助教重新批改之後就會被更新成正確的分數了。

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

以下題目如果沒有特別指名，請用 C++ 為基準作答。若你看到一段程式碼，請假設他們是被寫在一個有良好 include 敘述、using namespace 敘述的程式的結構正確的 main function 裡面。

(a) 如果你在一般的電腦上執行

```
float a = 0.1;
float b = 0.2;
if(a + b == 0.3)
    cout << "!";
else
    cout << "?";
```

結果看到問號，那就是那臺電腦壞了。

(b) 如果你在一般的電腦上執行

```
char c = 0;
cin >> c;
char d = c + 32;
cout << d;
```

則不論輸入哪個英文大寫字母，都會看到其相對應的小寫字母被印出來。

(c) 當你執行

```
int array[10] = {0};
cout << array[0] << "\n";
cout << array[20];
```

是有可能會遇到 run-time error 的。

(d) 當你執行

```
int array[10] = {0};
cout << array << "\n";
cout << array + 20;
```

是有可能會遇到 run-time error 的。


(e) 當你執行

```
int array[10][2] = {0};
for(int i = 0; i < 20000; i++)
    cout << array[i] << "\n";
```

是有可能會遇到 run-time error 的。

小提醒：在 PDOGS 上面讓大家繳交此題的地方，會有兩組「與上面正式要計分的題目完全無關的」範例輸入輸出，純粹是用來讓大家確認自己那個被批改的 if-else 程式是可以被正確執行的。請確

第二題



普通車
停靠車站

直達車及普通車皆停靠車站

機場捷運

票價表

A1 臺北 車站	A2 三 重 站	A3 新北 產業 園區 站	A4 新莊 副 都 心 站	A5 泰 山 站	A6 泰山 貴 和 站	A7 體育 大學 站	A8 長庚 醫院 站	A9 林口 站	A10 山鼻 站	A11 坑口 站	A12 機場第一 航廈 站	A13 機場第二 航廈 站	A14a 機場 旅館 站	A15 大園 站	A16 楊山 站	A17 華航 站	A18 高鐵 桃園 站	A19 桃園 機場 站	A20 興南 站	A21 環北 站
30																				
35	30																			
35	30	30																		
35	30	30	30																	
50	40	30	30	30																
80	60	35	35	35	30															
80	60	35	35	35	30	30														
80	60	35	35	35	30	30	30													
120	100	75	75	75	60	40	40	40												
130	110	85	85	85	70	50	50	50	30											
160	140	115	115	115	80	60	60	60	30	30										
160	140	115	115	115	80	60	60	60	30	30	30									
160	140	115	115	115	80	60	60	60	30	30	30	30								
160	150	125	125	125	90	70	70	70	35	30	30	30	30							
160	160	135	135	135	100	80	80	80	45	35	30	30	30	30						
160	160	140	140	140	110	90	90	90	50	40	30	30	30	30	30					
160	160	150	150	150	115	95	95	95	60	50	35	35	35	30	30	30				
160	160	155	155	155	125	105	105	105	65	55	40	40	40	35	30	30	30			
160	160	160	160	160	140	120	120	120	80	70	55	55	55	45	35	30	30	30		
160	160	160	160	160	160	125	125	125	85	75	65	65	65	55	45	35	30	30	30	

公告 Notice

2018.10.1起,全線區間票價優惠10元。
Special fare scheme starts from 1st Oct., 2018.

備註:不用票種與車種,同享優惠。系統票價圖顯示為原價,優惠票價以售票系統顯示為主。既有優惠維持不變。*Remarks: Special fare scheme applies to all kinds of tickets and passenger trains. Special fares will be displayed on our ticketing system instead of this fare map. Existing special offers still stand.

在這題當中，我們希望在給定總共 n 個車站的票價梯形圖與起訖車站後，找出藉由多次進出站，最小化總票價的搭乘方法。然而，中途下車多次會產生相當大的時間成本，因此我們限制中途下車次數 m 最多為 3 次。例如欲從 A1 至 A12，最便宜的搭乘方法可以藉由中途下車 2 次來達成，先由 A1 至 A3 (25 元)，再從 A3 至 A7 (20 元)，再從 A7 至 A12 (50 元)，總計需要 95 元，較直接搭乘至 A12 便宜了 55 元。

3

輸入輸出格式

系統會提供一共 20 組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中，總共有 n 行。第一行有 4 個整數，分別為 n 、 m 、 s 、 d ，分別代表總共的車站數、最多中途下車次數、起點站代號、終點站代號。第二行至第 n 行之中，第 $i+1$ 行總共有 i 個整數。整數之間以空白隔開。第 i 行的第 j 個數字 p_{ij} 代表中途不出站從第 j 個站搭到第 i 個站的票價。已知 $5 \leq n \leq 100$ 、 $1 \leq m \leq 3$ 、 $1 \leq s < d \leq n$ 、 $15 \leq p_{ij} \leq 1530$ 。讀入這些數字之後，請依計算出最小化票價的搭乘方法，將中途下車的站代號依序印出，從離起點站最近的中途下車車站印起，直到離起點站最遠的。若存在多個解具有相同的票價，請選擇中途下車次數最少的方法。若中途下車次數與票價皆相同，請選擇第一個下車車站代號最小的，若仍有相同，則選擇第二個車站代號最小的，依此類推。印完中途下車車站編號後，接著印出整趟旅程的票價。若中途不需下車，則請印出一個 0，接著印出整趟旅程的票價。任兩個整數之間以一個空白字元隔開。

舉例來說，如果輸入是

```
21 3 1 13
20
25 20
25 20 20
25 20 20 20
40 30 20 20 20
60 45 20 20 20 20
70 50 25 25 25 20 20
70 50 25 25 25 20 20 20
110 90 65 65 65 50 30 30 30
120 100 75 75 75 60 40 40 40 20
150 130 105 105 105 70 50 50 50 20 20
150 130 105 105 105 70 50 50 50 20 20 20
150 130 105 105 105 70 50 50 50 20 20 20 20
150 140 115 115 115 80 60 60 60 25 20 20 20 20
150 150 125 125 125 90 70 70 70 35 25 20 20 20 20
150 150 130 130 130 100 80 80 80 40 30 20 20 20 20 20
150 150 140 140 140 105 85 85 85 50 40 25 25 25 20 20 20
150 150 145 145 145 115 95 95 95 55 45 30 30 30 25 20 20 20
150 150 150 150 150 130 110 110 110 70 60 45 45 45 35 25 20 20 20
150 150 150 150 150 150 115 115 115 75 65 55 55 55 45 35 25 20 20 20
```

則輸出應該是

```
3 7 95
```

如果輸入是

```
8 3 1 8
20
20 20
20 20 20
25 20 20 20
25 25 20 20 20
30 25 25 20 20 20
30 25 25 25 20 20 20
```

則輸出應該是

```
0 30
```

小提醒：如果這題沒有限制最多中途下車次數，那就必須使用「遞迴」(recursion) 這樣的觀念和技巧才比較容易做出來。為了讓題目簡單一點，我們在本題有設置最多中途下車次數的限制，只要寫巢狀迴圈就能完成了。反過來說，超過此限制的搭車方式當然就會被判定為錯誤答案了。若你覺得這題有些難，也請不要放棄，會有幾筆測試資料的 $m = 1$ ，若你的程式能正確處理這種情況，也會得到一些分數的；你也可以盲目地總是印出 0 和那個直達票價，說不定也可以賺中幾分。

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。一筆測試資料佔 2 分。
- 這一題的其中 20 分會根據你所寫的程式的品質來給分。助教會打開你的程式碼並檢閱你的程式的運算邏輯、可讀性，以及可擴充性（順便檢查你有沒有使用上課沒教過的語法，並且抓抓抄襲）。請寫一個「好」的程式吧！

第三題（加分題）

（40 分）在密碼學之中，有一種安全協定，稱為 *Diffie-Hellman key exchange*，可以讓雙方在完全沒有對方任何預先資訊的狀況下透過不安全通道建立起一個金鑰，以加密所有要傳送的訊息。圖 ?? 是運作原理的示意圖。

簡單來說，這個協定是如此運作的。假設有二個人 Alice 與 Bob 需要互相傳送訊息，首先二人會約定一個質數 p ，以及一個底數 g ，並公開讓所有人知道。接著，Alice 與 Bob 會分別選定一個數字（有時稱為私鑰）， a 與 b ，其中 $a, b \in \{1, 2, \dots, p-1\}$ ，並各自保留好不讓其他人知道。因此 Alice 不知道 b 是

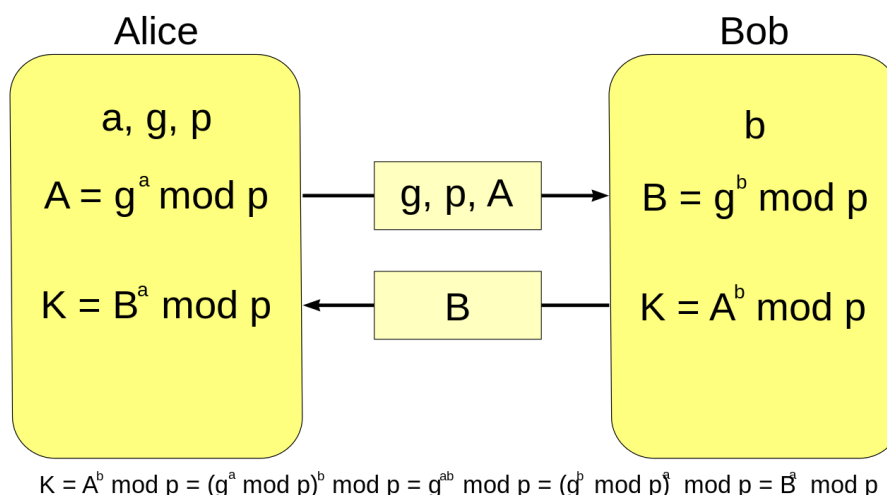


圖 2: Diffie-Hellman key exchange (圖片來源：Wikipedia)

多少，Bob 也不知道 a 是多少。他們會分別利用 a 與 b 計算出各自的公鑰 A 及 B ，計算的方法為

$$A = g^a \bmod p \quad \text{與} \quad B = g^b \bmod p$$

並且將這二個數字公開讓所有人知道。此處的 \bmod 是取餘數的運算。二人取得對方的公鑰後再分別進行以下的計算：

- Alice : $B^a \bmod p = (g^b)^a \bmod p = K$ 。
- Bob : $A^b \bmod p = (g^a)^b \bmod p = K$ 。

如此他們就能算出同一個數字，也就是二人共通的金鑰，最後就能用這個金鑰加密後續所有要傳送的訊息。

知道 K （一個整數）能有什麼用呢？一個很簡單的例子是凱撒加密（Caesar cipher），也就是將要傳送的文字的每個字母都向後移動 K 個單位，作為密文。例如 $K = 2$ 時，A 就會變成 C，B 變成 D，Z 變成 B。若 Alice 今天想要傳送「HELLO」的訊息給 Bob，使用金鑰 $K = 5$ 加密後得到的結果就會得到「MJQQT」的密文，如此一來在不安全的管道中傳送時，即使被其他人拿到這個訊息，他們也不會知道它代表什麼意義，因為他們不知道 K ，然而當 Bob 拿到訊息後，則可以反向操作 Alice 進行的動作，將每個字母都左移 5 個，得到「HELLO」。³

讓我們舉例說明計算的過程。如果二人選定 $p = 11$ 、 $g = 3$ ，他們會經過以下的步驟得到 K ：

1. 各自選擇私鑰 a 與 b ：例如 Alice 選擇 $a = 4$ ，Bob 選擇 $b = 2$ 。
2. 各自計算公鑰 A 與 B ：
 - Alice : $A = 3^4 \bmod 11 = 4$ 。
 - Bob : $B = 3^2 \bmod 11 = 9$ 。
3. 公開公鑰，讓對方知道自己的公鑰是多少。

³當然，經過凱撒加密後的訊息，只要把 26 個文字都嘗試一遍，就能破解，所以只有在古早時代適合使用，但現代很不合適，這邊純粹只是為了說明題目才舉這個當例子。

4. 計算共通的金鑰 K ：

- Alice： $K = 9^4 \bmod 11 = 5$ 。
- Bob： $K = 4^2 \bmod 11 = 5$ 。

得到他們共通的金鑰 $K = 5$

作為一名攻擊者，你會希望破解出 Alice 及 Bob 的私鑰 a 與 b ，如此一來便可計算出二人之間的共同秘密 K ，用以破解他們之間傳送的訊息。也就是說，希望找出 g 的幾次方會變成 A 以及 g 的幾次方會變成 B ，這個問題在密碼學上稱為離散對數問題，實務上 p 會是一個非常大的質數，以致於計算出 a 與 b 需要幾年或是幾乎不可能被算出。但在本次作業中，我們限制 $2 \leq p < 100$ ，好讓計算可以在很短的時間中完成。

重要提醒：在計算時，為了避免 g^a 與 g^b 數字太大而發生溢位 (overflow)，可以在每次做完乘法後先對結果取餘數，再乘下一個，這是因為

$$xy \bmod z = (x \bmod z)(y \bmod z) \bmod z$$

對所有正整數 x 、 y 、 z 都成立（它的正確性可以很容易地被證明，請試試看吧）。

重要提醒 2：在給定同樣的 p 、 g 、 A 、 B 的狀況下可以求得多個 a 與 b ，例如

$$3^2 \bmod 11 = 3^7 \bmod 11 = 9$$

因此為求答案的一致性，在輸出答案的時候請輸出 **最小的** a 與 b

輸入輸出格式

系統會提供一共 20 組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中，第一行會有四個整數，依序為 p 、 g 、 A 與 B 。任兩個整數之間被一個空白字元隔開。已知 $2 \leq p < 100$ 、 $2 \leq g \leq 50$ 、 $2 \leq A \leq p-1$ 、 $B \leq p-1$ 。讀入這些數字之後，請依照題目說明的規則去計算 Alice 與 Bob 的私鑰分別為何，並依序印出二人**最小的**私鑰 a 與 b 以及共通的金鑰 K 。被印出的兩個整數之間用一個空白字元隔開。請注意輸出的最後面應該是一個整數而不是逗點或空白。

舉例來說，如果輸入是

11 3 4 9

則輸出應該是

4 2 5

針對這個題目，你可以使用任何方法。這一題的 40 分會根據程式運算的正確性給分，一筆測試資料佔 2 分。