# 民國一○七年新生組程式設計競賽(106學年度入學)

# 2018/09/03

# 競賽須知：

# 試題總共六題，不論難易每題皆100分，總分600分。新生組競賽採人工評分，未完成的程式仍有部份分數。各題均須準備好可執行的執行檔並且保留程式碼原始檔備查，評分時評審必要時得要求重新編譯產生執行檔。

# 競賽時間五小時(9:30-14:30; 評審委員得視情況調整時間)，競賽開始50分鐘後不得進場。每一組所有組員共用一部電腦，未經監考人員同意不得自行更換電腦。

# 對試題有疑義者請於競賽開始後一小時內舉手提問，評審委員將視情況回答或不予回答或對全體競賽者宣布修正與說明試題。競賽開始一小時之後即不得再對試題提問。

# 評審委員將於14:30開始評分(評分順序由評審委員決定)，未經評審評完分前不得離場，否則視同放棄。提早做完隊伍可在13:30至14:15間舉手要求提前評分，其他時間不得要求提前評分。每組均配發有編號的拇指碟，各組必須自行隨時備份於該有編號的拇指碟，並於競賽結束之前將所有完成的程式原始碼與執行檔存入拇指碟內備查。競賽結束後監考人員將立刻收回拇指碟(備查用)，所有參賽同學請留在原地靜待評審評分，各組請確定六題都被評過分數簽名後才可離場。評分順序由評審委員視情況決定，考生不得異議。

# 競賽期間各組組員討論時請勿過於大聲影響他組，若需上洗手間請舉手由監考人員陪同，進出考場均須簽名紀錄。

# 中午12:15開始提供便當，可自行決定是否在競賽結束前輪流用餐，請舉手告知監考人員後到指定區用餐，進出考場均須簽名紀錄。

# 評分結束離場的同學請繳回名牌換取參賽紀念品。

# 競賽結果評審委員討論議決後將交由資訊技術服務中心統一公告於本校資訊技術服務中心網頁。

# 第 1 題 潛入敵營 (Against the Terrorist)

You are a spy working for the government. Recently you got a tip-off that a terrorist group has bought amounts of weapons, and the shipment is about to arrive tonight.

The shipment is stored in large cubical crates(木板箱), all the crates have the same volume and dimensions. The crates are stacked in piles, forming a three-dimensional grid. And there is a security system which is monitoring the piles using three cameras: a front camera, a side camera, and a top camera. The image from the front camera shows the height of the tallest pile in each column, the image from the side camera shows the height of the tallest pile in each row, and the image from the top camera shows whether each pile is empty. Once the security system detects a change in these images, the alarm will sound. Since each crate has been locked, you cannot steal the content in the crate only.

Now, your mission is to steal as many crates as possible without being detected.

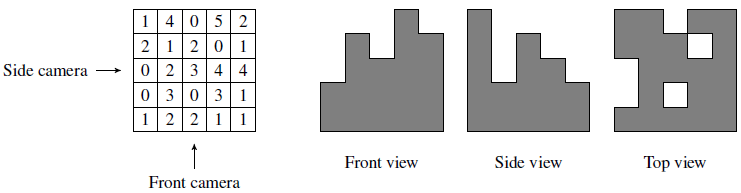


Figure 1. Heights of each grid and the corresponding camera views.

**Input**

You should read the input from a file. The first line of the input file will contain two integers, *m* (1 ≤ *m* ≤ 100) and *n* (1 ≤ *n* ≤ 100), representing the number of rows and columns in the grid, respectively. Following are *m* lines, each line contains *n* integers, which represents the height of crates in the corresponding grid. All height are between 0 and .

**Output**

The output is the maximum number of the crates that can steal without being detected.

**Sample Input**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5 | 5 |  |  |  |
| 1 | 4 | 0 | 5 | 2 |
| 2 | 1 | 2 | 0 | 1 |
| 0 | 2 | 3 | 4 | 4 |
| 0 | 3 | 0 | 3 | 1 |
| 1 | 2 | 2 | 1 | 1 |

**Sample Output**

9

**注意: 本題評分時的執行時間每次不得超過10秒。**

# 輸入可以從標準輸入裝置(鍵盤) ，或是從檔案讀取，檔案名稱請用 data1.txt為輸入檔案名稱；輸出則須寫到標準輸出裝置(螢幕)。

# 第 2 題 數字融合

給一串*n*個大於零的正整數，**X** = {*x*0, *x*1, ..., *xn*-1}，其中任一整數*xi*可以在{*xi*+1, *xi*+2, ..., *xn*-1}之中找到相同的數字*xj* （從*i*之後找相同數字，不包含*i*以前的數字），即可進行融合。融合的方法為：

1. 若*xi*為偶數，*xi* = *xi* + 1；若*xi*為奇數，*xi* = *xi* + 3
2. 將*xj*從**X**中移除

如此反覆進行融合，直到**X**中沒有任何數字可以再進行融合。舉例說明，若**Ｘ** = {8, 4, 5, 4, 1}，則融合過程會是如下：

8 **5** 5 1

8 **8** 1

**9** 1

那麼最後結果就是9與1。

請注意，*xi*必須完成融合或無法在*i*之後找相同數字，才會進行*xi*+1*­*的融合。所以若**Ｘ** = {6, 6, 6}，則融合過程會是如下：

**6 6**  6 🡺 7 6

或

**6** 6 **6 🡺** 7 6

但不會有這樣的結果：

6 **6 6**  🡺 6 7

請注意，融合結果其數字的排列不是唯一，有可能會產生不同的順序。比如**Ｘ** = {6, 6, 5, 4, 5, 7, 8, 6}的融合結果就會有兩種排列順序：

10 6 9 4

與

10 9 4 6

若有兩種以上的融合結果，請按照如下準則來輸出：

若有X與Y兩種排列順序且X在Y之後輸出，則必存在 *xi* > *yi* 且*x*0, *x*1, ..., *xi*-1皆≥ y0, *y*1, ..., *yi*-1

本題所有測試資料、結果以及資料數量皆可用32位元的整數型態來表示。

**Input**

輸入資料第一列為一個大於零的正整數，即*n*。第二行則是*n*個大於零正整數，每個數字之間以空格字元隔開，即為*x*0 *x*1 ... *xn*-1。

**Output**

輸入資料第一列為融合結果的數量*k*。接著輸出*k*列的融合結果，每列代表一組融合結果，每個數字之間以空格字元隔開。請注意，各結果不可重複。

**Sample Input 1**

8

6 6 5 4 5 7 8 6

**Sample Output 1**

2

10 6 9 4

10 9 4 6

**Sample Input 2**

6

5 5 6 6 5 6

**Sample Output 2**

3

8 5 7 6

8 7 5 6

8 7 6 5

**評分**

1. **輸出正確的*k* (10%)**
2. **當*k* > 1時，可正確地輸出*k*個結果(70%)；只能輸出一個融合結果 (30%)**
3. **當*k* > 1時，輸出順序正確(10%)**
4. **當*k* > 1時，輸出不重複(10%)**

**注意: 本題評分時的執行時間每次不得超過10秒。**

**輸入可以從標準輸入裝置(鍵盤) ，或是從檔案讀取，若從檔案讀取則檔案名稱請用** **data2.txt為輸入檔案名稱；輸出則須寫到標準輸出裝置(螢幕)。**

# 第 3 題 逃出洞穴 (Escape The Cave)

泰國在洞穴失蹤的足球隊，全數獲救，舉世譽為偉大的奇蹟。小明有一張類似的洞穴地圖。這地圖錯綜複雜，並且相當難行。這地圖只以點和線段來表示，線段代表洞穴內的路段。這些點大部分是路段的交叉點，只有少數幾個點是出/入口。地圖中的每一線段都有標示長度以及最高的行走速度。

現在要訓練小明走出洞穴的能力。小明只有m天的乾糧和飲水，**請設計程式幫小明找出一條在m天內可以走到任一出/入口的路徑**。**小明每24小時內，必須至少休息8小時，以維持體力**。這8小時的休息時間可以是連續的，也可以任意分配在24小時內休息，並且在任何地方都可以休息。

**註:**

**不必找出最短時間的路徑**，**只要找出一條可以在m天內走到任一出/入口的路徑**，**並將此路徑依下述輸出的格式列印出來即可**。

**輸入格式**

**檔案內有多組測試資料，每一組測試資料最後以0 代表該組測試資料結束。檔案結束代表輸入資料結束。**

每組測試資料

第1列為數字 ，，代表小明只有 天的乾糧和飲水

第2列為數字 ，，代表地圖中共有個點，點的編號從1至

第3列為數字 ，，代表地圖中共有個出口

第4列為數字 , , …, ，代表點 , , …, 為出口點

第5列為 代表小明的位置，其中

和 **代表小明目前在點 和 間的路段上**

**代表小明離點 的距離**

第5列以後，每一列的數字為 ，代表一條路段的資料，其中

和 是點的編號，代表這路段連接點 和

**代表此路段的長度(km，公里)**

**代表目前此路段最高的行走速率(km/h，公里/小時)**

若 ，代表該組測試資料結束。

**輸出格式**

每組測試資料的輸出格式為

第一列為數字y

若 y=0，代表找不到可以在 天內走出洞穴的路徑

若 y=1，代表找到可以在 天內走出洞穴的路徑

若 y=1

第二列為 **, , …, ，為路徑經過的點，其中最後一個點 為出口點**

第三列為 和

為實際行走的總時數；為休息的總時數。

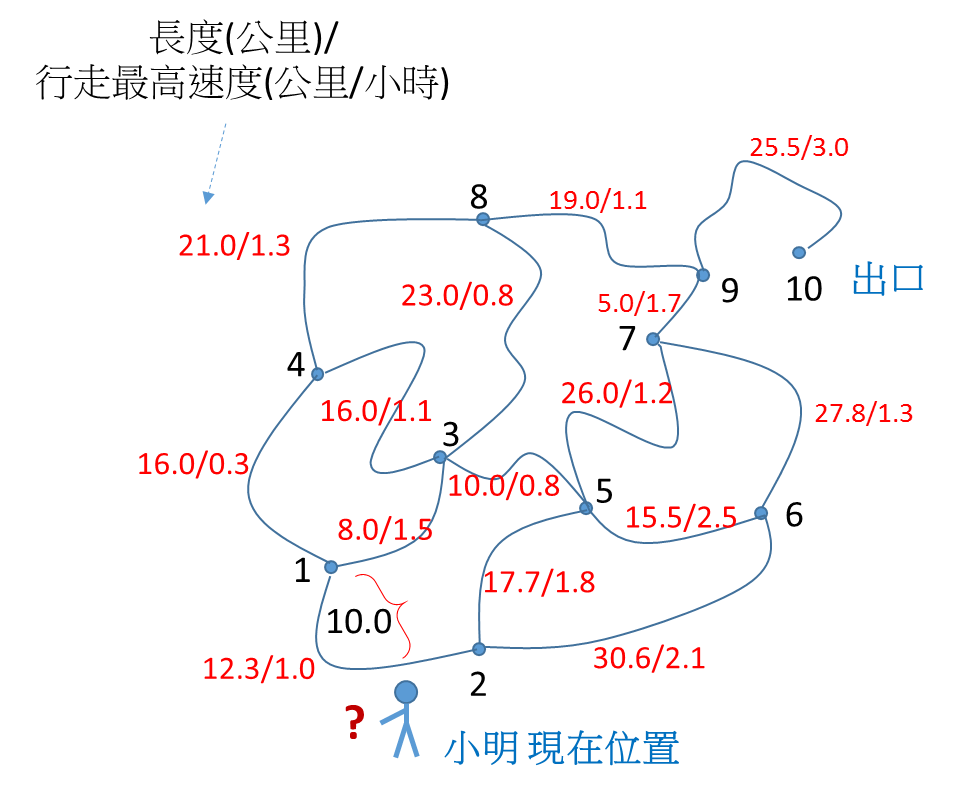
和 必須符合 的限制。

**和 ，請四捨五入至小數點一位**。

請在兩組答案之間列印 “----------------------“，以便區隔

**範例**

以下是一組測試資料的輸入和輸出範例



**輸入範例 (右邊文字是說明)**

3 代表小明只有3天的乾糧和飲水

10 代表地圖中共有10個點，點的編號從1至10

1 代表地圖中只有1個出口

10 代表點 10為出口點

1 2 10.0 代表小明目前在點 **1和點2間的路段上，並且離點1 的距離為10.0公里**

1 2 12.3 1.0 點1和點2間路段長12.3公里，最高行走速率為1.0 km/h

1 3 8.0 1.5

1 4 16.0 0.3

2 5 17.7 1.8

2 6 30.6 2.1

3 4 16.0 1.1

3 5 10.0 0.8

3 8 23.0 0.8

4 8 21.0 1.3

5 6 15.5 2.5

5 7 26.0 1.2

6 7 27.8 1.3

7 9 5.0 1.7

8 9 19.0 1.1

9 10 25.5 3.0

0 代表本組測試資料結束

**輸出範例 (右邊文字是說明不必印)**

1 代表找到可以在3天內走出洞穴的路徑

2 5 7 9 10 為路徑經過的點，其中最後一個點10為出口點

45.2 16 實際行走的總時數為45.2小時；休息的總時數為16小時

---------------------- 代表本組答案結束(以便和下一組答案區隔開)

**注意: 本題評分時的執行時間每次不得超過10秒。**

# 輸入可從標準輸入裝置(鍵盤)，或是從檔案讀取，若從檔案讀取則檔案名稱請用 data3.txt為輸入檔案名稱；輸出則須寫到標準輸出裝置(螢幕)。

# 本頁空白

# 方便雙面列印後每題都從奇數頁開始 (封面另外印)

# 第 4 題 拉曲線大賽 (Find The Best Fitting Curve)

Given an by array of short integers with values from 0 to 255 as image data. We often use the linear least square fitting to obtain a plane (). If the original data value is , we use the plane to normalize the original data as .

Let’s simplify the problem and start to solve it step by step. In the beginning, you need to find the line between two data points. For example, you get an array of data of 100 and 200. You can find out a fitting line of to fit the data of and you obtain . You can normalize your data by using .

As a second step, you need to find the best fitting line to an array of data. In this case, you have more than two points. The best fitting will give you . Here you have two unknown parameters so you need one more equation to solve it. The additional equation is . You see that it is just a simple multiplication of on the first equation. Now you can obtain the two parameters by using the two equations. After you get the parameters, you can normalize your original data by .

As a third step, you can use parabola curve functions to fit the data. The best fitting will give you . To find the solutions of the parameters, you need to have two more equations: and . Please find the three parameters and normalize the data by:

.

**Input:** You need to read an array of unsigned short integers (0-255) from the standard input. The data are unsigned short integers separated by a space character.

**Output:** You need to check if is larger than two or not. The output gives:

1. Print out Linear least square fitting to the data gives and the normalized data are
2. If , print out Parabolic fitting to the data gives and the normalized data are

Please present the parameters in numbers with at least 5 digits. Remember to use variables of the type “double” in your calculation for the best precision and transform to unsigned integers in the range from 0 to 255.

**Sample Input(輸入範例)**

Example1:

100 200

Example2:

20 32 38 45 70 75 79 93 99

**Sample Output(輸出範例)**

Example1:

a = 100.00, b = 100.00, normalized data: 128 128

Example2:

a = 20.4889, b = 10.1833, normalized data: 127 129 125 121 136 131 125 129 125

a = 18.7818, b = 11.6465, c = -0.1829, normalized data: 129 129 124 120 135 130 124 129 126

**注意: 本題評分時的執行時間每次不得超過 5秒。**

# 輸入可以寫成從標準輸入(鍵盤)讀取或從檔案讀取，輸出則須寫到標準輸出裝置(螢幕)。若從檔案輸入請用 data4.txt為輸入檔案名稱。

# 第 5 題 沒有白吃的午餐 (No Free Lunch in a Treasure Land)

There is a two-dimensional treasure land. The land is discretized as a grid consists of a set of square cells. Each square cell contains a valuable (non-negative value) or dangerous (negative value) item. The valuable items can be sold for coins but coins must be consumed to clean the dangerous items. A region of the land can be enclosed by placing an axis-aligned rectangular fence. Inside the region, the valuable items are collected and the dangerous items must be cleaned up. In the game, the goal is to place a rectangular fence to enclose a region of the land such that the total value of the items inside the region is maximized. The constraint is that the perimeter of the fence is not larger than a given length. The fence must be placed at the boundaries of the square cells. The side lengths of the fence along the x-axis and y-axis must be multiples of kx and ky, respectively. The task is to compute the best region with the highest total value and the perimeter of the fence length is the smallest. Another way is to use a circular fence to bound a region. The fence is centered at the center of a square cell and the fence must be entirely inside the land. The diameter of the circular fence must be an odd number, i.e., 1, 3, 5, and so on. A square cell is inside the fence *iff* the center of the square cell is inside the fence.

If there are two or more regions have the same total value, the one with the shortest fence length is chosen. If the regions have the same fence length, the one with the smallest width is chosen.

The items are denoted as ‘0’,’1’, …, ‘9’, and ‘A’, ‘B’, …, ‘Z’. The item value is between [-50, 50].

An example is shown below.

Item values:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item | ‘A’ | ‘B’ | ‘C’ | ‘Z’ | ‘2’ | ‘1’ | ‘0’ | ‘5’ | ‘4’ |
| Value | -2 | -4 | -20 | 1 | 2 | 4 | -5 | 100 | 0 |

For example, ‘A’ represents a dangerous item. The corresponding item must be cleaned by consuming 2 coins. ‘Z’ is a valuable item and it can be sold for 1 coin.

|  |  |
| --- | --- |
| Figure 1. The land and the items. | Figure 2. The best retangular fence and the best circular fence. |

**Input:** The first line contains the width (*W*) and height (*H*) of the map, along the *x*-axis and *y*-axis, respectively. The second line contains the number of different items, *N*. The third line contains the fence maximum length, *L*, kx, and ky. After that there are N lines and each line contains two items: item symbol and its corresponding value. Finally, there are *H* lines and each line contains *W* item symbols. There are no space characters between two consecutive symbols at the same line.

**Output:** For each problem instance, you should read the data from a text file, e.g., data1.txt. You should print out the best rectangular fence at the first line: the value of the best region, and the width and height of the rectangular fence (i.e., side length along the *x*-axis and *y*-axis, respectively). At the second line, print the best circular fence: the value of the best region and its diameter.

|  |  |  |
| --- | --- | --- |
| **Sample Input 1**  3 3  2  6 1 1  5 1  Y -1  YYY  Y5Y  YYY | **Sample Input 2**  5 5  3  12 2 3  5 0  Y -2  H 2  55YHH  HY5YH  YH555  5H5YY  5HHHH | **Sample Input 3**  25 5  3  60 3 5  5 5  Y -2  H -2  55YHHHY5YHYH5555H5YY5HHHH  5H55YHYY5H5H55H5YHYHHY55Y  H5YYYHH55YYHYYH5H5YH555**H**5  H5Y5HYHHY55HYHYH55H5HY555  5HH5HHHY55H555Y5H5HH5YHYH |
| **Sample Output 1**  1 1 1  1 1 | **Sample Output 2**  8 2 3  8 3 | **Sample Output 3**  86 18 5  31 3 |

In Sample 3, the center of the circular fence is marked as bold.

There are five problems. Each one is 20pts.

1. W = 10, H = 10, N = 3, L = 60, kx = 1, ky = 1.
2. W = 50, H = 50, N = 6, L = 80, kx = 1, ky = 2.
3. W = 100, H = 100, N = 10, 100<L <=120, kx = 2, ky = 1.
4. W = 300, H = 300, 10< N <= 26, 300<L <= 400, kx = 6, ky = 7.
5. W = 500, H = 500, 26<N <= 36, 400<L <= 600, 6<= kx <= 8, 40 <= ky <= 50.

**注意: 本題評分時的執行時間每次不得超過 10秒。**

# 輸入可以寫成從標準輸入(鍵盤)讀取或從檔案讀取，輸出則須寫到標準輸出裝置(螢幕)。若從檔案輸入請用 data5.txt為輸入檔案名稱。

# 第 6 題 兩國論 (Calendar in different country)

　　大家都知道中華民國開國元年一月一日是西元1912/01/01,

那請問前一天是西元何年何月何日? 你一定說1911/12/31，答錯了! 大部分人都答錯! 正確答案是1911/12/18 星期日ㄝ!

所以1911年12月只有18天，沒有 1911/12/19 這天,就是說 1911/12/18 隔日就是 1912/01/01 星期一。那ㄟ安捏？後面補充。

請寫一個可考慮兩國的萬年曆程式，可以依使用者輸入查知某日是星期幾，也可以印出某月份的月曆，為了簡單起見，不必做印出一整年的年曆，注意這與 Unix 上 cal 和 ncal 命令不太一樣。

這個萬年曆要能切換指示程式根據美國用的曆法或是根據台灣(和中國一樣)的曆法。程式一開始預設是用美國的曆法，輸入 roc 表示切換到台灣(和中國一樣)的曆法，用 usa 表示切換到美國用的曆法。注意輸入的英文字要不管大小寫，就是說 roc 和 RoC 和ROC 都看作相同。

**輸入:**

每列輸入有四種可能:

1. 代表切換國家的英文字，只有 roc 和 usa 有意義
2. 一個數值，代表西元年，例如 2018 代表 2018年
3. 兩個數值，代表西元月年，例如 8 123 代表123年8月
4. 三個數值，代表西元日月年，例如 3 8 123 代表123/08/03

為了簡單，若第一個不是字母則該列就是數值，各數值之間空一格。

程式要能處理西元1年到9999年，若年不在此範圍則印出該年不在萬年內即可，要可以辨認錯誤的月和日，例如13月則印出沒有13月，又如32日要印出沒有32日，還有，如果該年不是閏年，則29日也是錯誤的，例如輸入 29 2 2018 要印出2018/02/29 不存在。

**輸出:**

針對每列輸入印出對應的答案:(以下對應上述各項輸入)

1. 印出切換曆法訊息並將規則切換到該國家，不認識的國家代碼則不切換且要印出「這是啥國家」表示不認識這代碼
2. 只有輸入年則印出該年一月的月曆，不必印出整年
3. 印出該月的月曆，參考後面範例輸出，但各月名稱可改用中文
4. 印出該日是星期幾(印英文或中文都可以)

詳細的輸出格式請參考後面輸出範例，以下先做簡單說明可能的輸入與輸出之對應。

輸入 2 9 1752 要輸出 USA: 1752/09/02 是 星期三

輸入 3 9 1752 要會輸出 USA: 1752/09/03 不存在

輸入 19 12 1811 會輸出 USA: 1911/12/19 是 星期二

若還在 USA 模式輸入 9 1752 則會輸出如下:

USA September 1752

Su Mo Tu We Th Fr Sa

1 2 14 15 16

17 18 19 20 21 22 23

24 25 26 27 28 29 30

若還在 USA 模式輸入 1 1912 則會輸出如下(只輸入1912也是):

USA January 1912

Su Mo Tu We Th Fr Sa

1 2 3 4 5 6

7 8 9 10 11 12 13

14 15 16 17 18 19 20

21 22 23 24 25 26 27

28 29 30 31

輸入 roc 會輸出 ROC: 切換到台灣/中國月曆，之後輸入數值一律根據台灣(和中國一樣)的曆法，直到輸入 usa 才切換到美國用的曆法。

若輸入 abc 則輸出 ABC: 這是啥國家

**說明:**

注意西元曆法因為以前天文學家搞錯，所以曾經調整過 !

　　羅馬曆法在1582年10月調整，但是各國家調整日期不盡相同，

英美等國是在1752年9月才調整，中華民國(清朝與民國)則是在1911/12/18的隔日才調整，所以中國(清朝/中華民國)的1911/12/18隔日是1912/01/01 星期一。

這題的重點除了要與 Unix 系統 cal 類似的排版印出外, 還要知道閏年的算法，以及參考點的日期是星期幾，還有要知道美國和中國曆法的調整點:

美國是 1752/09/02 星期二隔日為 1752/09/14 星期三

中國則是 1911/12/18 星期日隔日為 1912/01/01 星期一

如前所述, 各國調整曆法的日期並不相同, 這可以參看Unix的 ncal -p 命令:

Linux% ncal -p

AL Albania 1912-11-30 IT Italy 1582-10-04

AT Austria 1583-10-05 JP Japan 1918-12-18

AU Australia 1752-09-02 LI Lithuania 1918-02-01

BE Belgium 1582-12-14 LN Latin 9999-05-31

BG Bulgaria 1916-03-18 LU Luxembourg 1582-12-14

CA Canada 1752-09-02 LV Latvia 1918-02-01

CH Switzerland 1655-02-28 NL Netherlands 1582-12-14

CN China 1911-12-18 NO Norway 1700-02-18

CZ Czech Republic 1584-01-06 PL Poland 1582-10-04

DE Germany 1700-02-18 PT Portugal 1582-10-04

DK Denmark 1700-02-18 RO Romania 1919-03-31

ES Spain 1582-10-04 RU Russia 1918-01-31

FI Finland 1753-02-17 SI Slovenia 1919-03-04

FR France 1582-12-09 SW Sweden 1753-02-17

GB United Kingdom 1752-09-02 TR Turkey 1926-12-18

GR Greece 1924-03-09 \*US United States 1752-09-02

HU Hungary 1587-10-21 YU Yugoslavia 1919-03-04

IS Iceland 1700-11-16

**Sample Input (輸入範例)**

**3 9 1752**

**29 2 2018**

**29 13 2018**

**2 9 1752**

**9 1752**

**roc**

**3 9 1752**

**19 12 1911**

**12 1911**

**1912**

**usa**

**19 12 1911**

**abc**

**RoC**

**19 12 1911**

**Sample Output(輸出範例)**

**USA: 1752/09/03 不存在**

**USA: 2018/02/29不存在**

**USA: 2018/13/29 沒有13月**

**USA: 1752/09/02 是 星期三**

**USA September 1752**

**Su Mo Tu We Th Fr Sa**

**1 2 14 15 16**

**17 18 19 20 21 22 23**

**24 25 26 27 28 29 30**

**ROC: 切換到台灣/中國月曆**

**ROC: 1752/09/03 是 星期四**

**ROC: 1911/12/19 不存在**

**ROC December 1911**

**Su Mo Tu We Th Fr Sa**

**1 2 3**

**4 5 6 7 8 9 10**

**11 12 13 14 15 16 17**

**18**

**ROC January 1912**

**Su Mo Tu We Th Fr Sa**

**1 2 3 4 5 6**

**7 8 9 10 11 12 13**

**14 15 16 17 18 19 20**

**21 22 23 24 25 26 27**

**28 29 30 31**

**USA: 切換到美國/英國月曆**

**USA: 1911/12/19 是 星期二**

**ABC: 這是啥國家**

**ROC: 切換到台灣/中國月曆**

**ROC: 1911/12/19 不存在**

更多提示:

　(1)首先，當然須知道 1/01/01 這天是星期幾? 星期六啦!

(2)然後，要能知道每年是閏年還是平年? 可寫函數isleap(n);

當然這必須考慮到美國與中國調整曆法的日期不同(要有個變數記住目前採用 USA 還是採用 ROC 的曆法) 。

(a)若照英美曆法，1752年(含)之前都用年除以 4 除得盡就是閏年;

(b)若考慮中國曆法則是 1911年之前用年除以 4 除得盡就是閏年 !

(c)調整點之後閏年算法請參考(4)說明。

　　　　都是除以 4 除得盡就是閏年!所以要寫到 isleap(n)內

(3)曆法調整日各國不同:

　　(a)若考慮美國英國曆法，1752年9月要特別處理，

　　　　1752/09/02 星期三，隔天是 1752/09/14 星期四!

　　(b)若考慮中國則在1911年12月18日星期日,

　　　　隔天就是 1912年1月1日星期一 ! 所以1911年12月只有18天!

(4)關於(2)的閏年判斷補充，在**調整日之後的閏年算法**如下:

　 (a) 除以 400 除得盡則是閏年，否則 :

(b) 除以 100 除得盡則不是閏年! 否則 :

(c) 除以 4 除得盡則是閏年，否則是平年 !

(5)如何計算某日是星期幾? 很簡單，針對給定任何一個日期，

先決定在其之前的參考點日期，算出它相對於參考點差了幾天, 然後從差了幾天就可推算出該日是星期幾 (自己想想用手做如何做?)

(6)關於算某日是星期幾的參考點有兩個，一個就是 1/1/1 星期六，另一個就是調整日的隔日，英美是1752/09/14 星期四，

中國則是 1912/01/01 星期一。

================================================

**補充說明:(關於曆法的調整)**

你可以用 Unix (Linux/FreeBSD等) 下這命令 ncal -p 以及 ncal -sCN 12 1911 看看便知道!

為何要調整曆法? 當然是因為弄錯了!

誰弄錯了?

羅馬帝國的凱薩大帝的天文學家弄錯了!

怎會弄錯?

因為地球繞太陽一圈到底多久? 以前弄錯!

　　西元曆法是根據地球繞太陽一圈作一年! 可是一年究竟是幾天?

或者說為何會有閏年?

顯然地，地球繞太陽一圈不是剛好365天, 也不是剛好366天!

　 在公元1582年之前，大家一直以為地球繞太陽一圈是 365.25天，所以才會認定每四年該多一天(閏年)!

　　當然，那是錯的，因為地球繞太陽一圈是365.2422天，不是365.25天!

我們現行的曆法稱為格里曆（Gregorian calendar），就是俗稱的公曆或公元，這是十六世紀的羅馬教皇Gregorian XIII (格里十三世)針對當時使用的儒略曆(Julian calendar)進行修訂後，於1582年10月開始實行的。

　　在1582年10月4日(含)之前所實行的曆法稱為儒略曆(Julian calendar)，從公元前46年(BC 45)羅馬統治者儒略‧凱薩大帝(Julius Caesar)開始實施。

　(Note that 公元前一年是 BC 0, 沒有公元 0 年 )

　　在公元4年3月1日以前的閏年是不規則的，自公元4年後則規定所有4除得盡的年都是潤年，也就是等於規定一年是365.25天，與真正的回歸年365.2422天相差0.0078天，從公元前46年(BC 45)羅馬統治者儒略‧凱薩大帝開始實施後，到格里十三世的一千多年裡，儒略曆累積的誤差已達10天(24節氣都亂掉了)，為了消除這個差額，格里十三世於是將儒略曆1582年10月4日的隔天訂為格里曆的1582年10月15日，把日期去掉十天，就可以讓春分日期又回復到3月21日左右，冬至回到12月22日左右。

所以應該是AD 1 到 AD 1582 年間多算了 12 天，BC 45 到 AD 1 之間少算了2天。

這麼說來應該是1582年的10月只有21天，

那為什麼在 UNIX上用 cal 9 1752 會是1752年的九月憑空消失了11天呢? 而不是1582年10月少掉10天呢?

(就是說用 cal 10 1582看到1582年10月仍是31天 !)

這牽涉到新舊教的問題，更精確的說是因當時的英皇亨利八世與教宗Gregorian關係惡劣，所以英國拒絕調整!

阿眾所週知，美國大都是英國人跑去新大陸阿後來獨立建國的，歷史太短，只好懷念"祖國"囉，既然 UNIX 是美國人寫的阿所以 cal 就採用英國的 Gregoriancalendar! 大不列顛(英國)是在 1752 年9月才改用 Gregorian calendar，當時誤差已達11天，所以該月就憑空消失了11天! (see "man cal")。所以 1752/9/2 是星期三，隔天就是 1752/9/14 星期四。

據說當時英國大遊行的盛況空前熱鬧，因為租房子的人抗議多繳了11天的房租! 要求政府應該用免稅來補償!

聰明的你可能發現問題:

就是英國美國加拿大等是在 1752年九月才調整, 那就是

這些年: 公元100, 200, 300, 500, 600, 700, 900, 1000, 1100,以及 1300, 1400, 1500, 1700 這些被都誤認為閏年!

總共十三個年各多算了一天，共多算了 13天 !

那樣不是該扣回 13 天?

阿怎會1752年9月只有少了 11 天 ? 難道又弄錯?

(大多數天主教國家1582年10月則是扣掉10天，但錯算12個閏年!)其實這沒有弄錯 ! 那是因為在公元前46年到公元開始之間也搞錯, 在該46年間少算了兩個閏年 !

所以兩相抵銷之下，1752 年9月只扣回 11天!(比1582年時多錯算了1700這年 )大家發現萬年曆很有趣吧!? 趕快寫程式囉。

注意輸出某月月曆要依照Linux 上 cal 的格式，不要依照如下 ncal 的格式:

Linux % **ncal -s CN 12 1911**  (CN 代表中國)

**December 1911**

**Mo 5 12**

**Tu 6 13**

**We 7 14**

**Th 1 8 15**

**Fr 2 9 16**

**Sa 3 10 17**

**Su 4 11 18**  (如前說的，中國與台灣 1911年12月只有 18天;**因為清朝沒有調整曆法**，推翻滿清後順便調整)

Linux2%

**注意: 本題評分時的執行時間每次不得超過10秒。**

# 輸入可以寫成從標準輸入(鍵盤)讀取或從檔案讀取，輸出則須寫到標準輸出裝置(螢幕)。若從檔案輸入請用 data6.txt為輸入檔案名稱。

評審委員將於**14:30開始評分**(評分順序由評審委員決定)，**未經評審評完分前不得離場，否則視同放棄。**提早做完隊伍可在13:30至14:15間舉手要求提前評分，其他時間不得要求提前評分。

**~ ~ ~ 考題結束 ~ ~ ~**