

台北市 102 學年度高級中學資訊學科能力競賽
程式設計試題

准考證號碼： _____
姓名： _____
USB 號碼： _____
評分主機帳號： _____

下表供自行記錄評分結果

題目 評分	間隔數 number	主機排程 schedule	直式撞球 fling	惡意軟體 malware	超級細菌 bacteria	任意門 transporter
一						
二						
三						
四						
五						
六						
七						
最高分						

總分 _____ / 600

注意事項

- 一、本競賽採電腦線上自動評分，程式必須依規定上傳至評分主機。請嚴格遵守每一題目所規定之原始程式、輸入檔、輸出檔等檔名及格式。若未遵守，該題將以 0 分計算。有關評測主機的詳細操作說明，請參考評分主機說明文件。
- 二、本次競賽採取全面回饋機制，程式上傳至評分主機後，將自動編譯並進行測試。視等待評分題數多寡，該題測試結果及該題得分數將可於短時間內得知。程式可重複上傳及評分，但同一題兩次上傳之間必須間隔二分鐘以上。每題最終分數以該題所有單次評分結果之最高分計算。
- 三、程式執行時，針對每組測試資料執行時間以 4 秒為限（以評分主機執行時間為準）。執行時間超過 4 秒者視同未完成，該組測試資料得分將以 0 分計算。每題可使用記憶體空間，除非題目另有規定，以 512MB 為限。
- 四、本次競賽第一題 (number) 不需要讀取輸入檔，該題測試資料已於題目中宣告。請先將答案寫入至一個純文字檔(檔名為“output_0.txt”)，再將該純文字檔上傳至評分主機進行評審。兩次上傳之間必須間隔二分鐘以上。
- 五、本次競賽第二題至第六題，學生需上傳原始程式碼 (.c, .cpp, .pas)，輸入資料則自“input.txt”檔讀取，計算結果需輸出至“output.txt”檔。注意：所有檔案的讀寫都在執行檔的工作目錄下進行，請勿自行增修輸出入檔的檔名或路徑，若因此造成評分程式無法評分，該次評分結果將以 0 分計算。
- 六、本次競賽每一題皆有不同難易度的測試資料，詳細配分及限制條件請詳各題題目說明。
- 七、本次競賽題目無需使用超長整數 (long long) 變數，請勿使用 long long 宣告變數。若因為使用該資料型態造成程式編譯或執行錯誤，導致評分程式無法正確評分，該次評分結果將以 0 分計算。
- 八、請用主辦單位分配的隨身碟備份原始程式碼，若因任何原因而需更換電腦時，僅能將隨身碟內程式複製至新電腦，或下載已上傳至評分主機的程式碼。
- 九、若以非 C/C++/Pascal 程式語言撰寫程式者，請事先告知監試人員，領取評分用隨身碟。若要接受評分，需將程式編譯成可執行檔 (.exe)，複製至評分用隨身碟，交給監試人員送交評審評分。評分結果將以書面通知。

間隔數 (number)

輸入檔：本題無輸入檔

輸出檔：output_0.txt

問題描述

在數字系統中，有一類的正整數具有下列特殊的性質，稱為「間隔數」：

- (1) 該數由恰好兩個相異的「單位數字」組成，每個單位數字為 0 到 9 中的一個整數；
- (2) 該數所含的單位數字，必定不會連續出現；
- (3) 該數至少含有一個單位數字，其出現的次數為兩次(含)以上。

例如，989 是一個間隔數，因為他含有兩個相異的單位數字，分別為 9 和 8，且這兩個數字並未連續出現，而 9 在 989 中共出現兩次；同理，6363 也是一個間隔數，因為他含有兩個相異的單位數字，分別為 6 和 3，且這兩個數字並未連續出現，同時在 6363 中各出現兩次。然而，10 並不是個間隔數，因為他的兩個單位數字都只出現過一次；222 也不是間隔數，因為他只有一個單位數字；221 也不是間隔數，因為他的單位數字 2 在 221 中連續出現。

依照上述的規則，我們可以得知前五個間隔數依次為 101, 121, 131, 141, 151。現在，請您設計一個程式，在第一行輸出第 10 個間隔數，在第二行輸出第 100 個間隔數，在第三行輸出第 10,000 個間隔數，在第四行輸出第 1,000,000 個間隔數，在第五行輸出第 100,000,000 個間隔數。

註 1：本題沒有資料需要自輸入檔讀入。測試資料已於題目中宣告，但仍須產生輸出檔。

輸入檔格式

本題無輸入檔。

輸出檔格式 (output_0.txt)

共五行，請依序在每一行中分別輸出第 10、100、10,000、1,000,000、100,000,000 個間隔數。

上傳檔案說明

本題不需要上傳程式原始檔，只需要上傳檔名為 output_0.txt 的輸出檔。

評分說明

本題共有五筆輸出資料，每筆正確解答可獲得 20 分。

主機排程 (schedule)

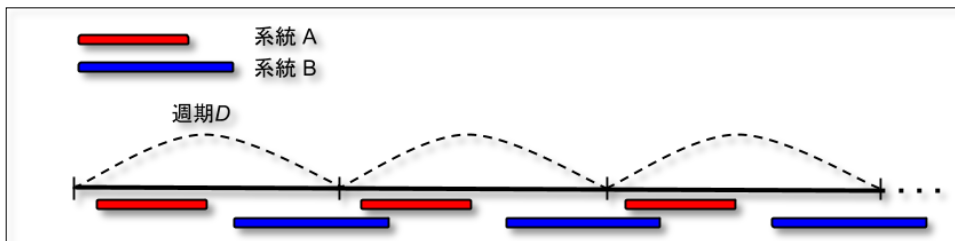
輸入檔: `input.txt`

輸出檔: `output.txt`

問題描述

云云科技顧問公司是一家幫忙企業客戶規劃實體主機轉成虛擬主機的諮詢服務公司，而使用雲端虛擬主機平台主要的好處，就是讓硬體資源可以集中共享，並透過平台有效率及彈性地分派硬體資源給實際需要的系統，不但可以因為減少主機的數量達到節能減碳，還可以為公司省下許多經費開銷。專案經理小楊，他的其中一項工作就是幫忙客戶整理各個系統運行實際需要的硬體資源，並依照系統執行的忙碌時段計算出實際需要的各項硬體資源總數，提供給企業客戶作為購置硬體之參考。

假定所有系統一旦開始執行後，就會固定以 D 天為一個週期重複執行（如下圖），且除了系統執行之忙碌時段外，其他都是閒置時間。當處於忙碌時段時，資源會被佔住，如果在相同時段有很多系統同時在忙碌時，就必須提供更多的硬體資源，才能確保這些系統能正常執行；反之，當系統閒置時，其所擁有的硬體資源會被雲端主機平台回收，並轉給其他需要硬體資源的系統。需要統計的硬體資源包含系統執行所需的核數（core）數、記憶體數（按每 GB 計）以及網路流量（按每 MB 計），而系統忙碌的時間皆以整點小時計，你的任務就是要幫小楊寫一個程式，能自動統計每個企業客戶在固定時間週期 D 內，任一時間下可能需要準備的各種硬體資源總數。



輸入檔格式 (`input.txt`)

第一列有兩個整數 n 與 D ，中間以一個空白區隔，分別代表此資料檔有 n 群的資料要分別統計， $1 \leq n \leq 10$ ， D 為一個定值，表其所有系統忙碌時間將以 D 天為一個週期循環。資料群由第二列開始，每群資料開頭的第一列是一個整數 s ，代表該群資料需處理的筆數，隨後每一列代表一筆系統運行時的資源需求資料，皆包含六個整數 $(T_d, T_s, T_h, R_{core}, R_{ram}, R_{bw})$ ，數字間以一個空白隔開，前三個整數為執行的時間，分別表示系統於第 T_d 天的整點 T_s 時開始執行，並一共持續執行了 T_h 小時 ($1 \leq T_d \leq D, 0 \leq T_s \leq 23, T_h \geq 0$)，但不會持續超過一個週期的總時數；後三個整數分別代表執行時所需要硬體資源，包含核心數 R_{core} ， $1 \leq R_{core} \leq 10$ ；記憶體數 R_{ram} GB， $1 \leq R_{ram} \leq 20$ ；以及網路頻寬 R_{bw} MB， $0 \leq R_{bw} \leq 30$ ；例如整數序列：3 10 15 2 4 3，則代表此系統在第三天上午 10 時起，連續執行 15 小時，至第四天凌晨 1 時止皆為該系統的忙碌時間，執行時需要 2 個核心、4GB 的記憶體以及 3MB 的網路流量才能順利運行。每群資料的最後一列，以一個數字「0」，代表該群資料的結束。請注意，這些硬體資源需求資料並未事先排序。

輸出檔格式 (output.txt)

每一列為統計每群資料會同時用到的最大資源數，應包含三個整數（每一個整數皆小於 30,000），分別代表同時間可能用到的最大核心數、最大記憶體數，以及最大網路頻寬數值。整數間以一個空白作為區隔。

輸入檔範例一

```
2 3
3
2 5 7 2 3 1
1 10 18 3 10 2
3 17 3 1 1 1
0
2
2 9 21 4 4 5
1 16 8 2 10 1
0
```

輸出格式範例一

```
3 10 2
4 10 5
```

(說明)

兩群資料中的忙碌時間都未重疊，故取各資源的最大值即為結果。請注意，第二群中的第一筆資料，執行時間已跨越至隔日。

輸入檔範例二

```
1 5
8
5 8 12 2 4 2
1 10 19 1 2 4
2 12 5 2 1 10
2 9 13 3 2 3
2 4 20 2 4 10
3 8 12 4 5 8
3 12 21 3 10 6
4 13 8 2 5 1
0
```

輸出格式範例二

```
7 15 23
```

(說明)

本測試資料有多組執行時段有重疊，如第 2 至 5 筆，以及第 6 及 7 兩筆，執行時段重疊時，資源必須累加，才能得知應準備的資源總數量。

評分說明

本題共有 5 組測試資料，每組 20 分。

第 1, 2 組測試資料， $D \leq 10, s \leq 20$ ；

第 3, 4, 5 組測試資料， $D \leq 1,000, s \leq 1,000$ 。

直式撞球 (fling)

輸入檔: `input.txt`

輸出檔: `output.txt`

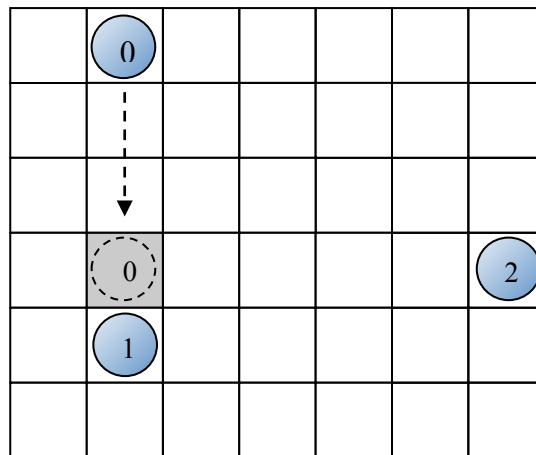
問題描述

直式撞球 (Fling) 和一般撞球不同之處，在於球只能沿橫向或直向移動，兩球撞擊時，原先移動中的球將停止在被撞擊球的前一格，如範例中 0 號球撞擊 1 號球後將停在灰底的格子中(請注意，這不是 1 號球原本的位置)，被撞擊的球則沿著原移動方向繼續移動，並可能再撞擊其它球，如果不再發生撞擊，則最後被撞擊的球將脫離球檯。

遊戲一開始，可以啟動任一顆球，沿橫向或直向去撞擊另一顆球 (包括相鄰的球)，但是請注意，如果一開始移動就無法撞擊到其它球，則為無效的移動，不可藉此將該球移出球檯。當移動中的球脫離球檯後，則可再啟動任一顆球去撞擊另一顆球，直到無法再產生撞擊為止。

不同的啟動順序，可能導致最後留在球檯上的球數不同。例如範例中，如果先啟動 1 號球去撞擊 0 號球，則最後會留下兩顆球。但若先啟動 0 號球去撞擊 1 號球，接著 2 號球去撞擊 0 號球 (或是 0 號球再去撞擊 2 號球)，則最後只會留下一顆球，

請找出一系列球的啟動順序，讓最後留在球檯上球的個數最少。



輸入檔格式 (`input.txt`)

第一行有一個整數 N ，代表開始時球檯上有 N 顆球。

第二行有兩個整數 S_x 和 S_y ，以空白分隔，代表球檯 X 方向和 Y 方向的檯面大小。

接下來有 N 行，每行有兩個整數 X_i 和 Y_i ，並以空白分隔。代表編號 i 的球的 X 座標和 Y 座標。球的編號 i 由 0 開始，最大編號為 $N-1$ 。

輸出檔格式 (`output.txt`)

每筆測試資料輸出下列資料:

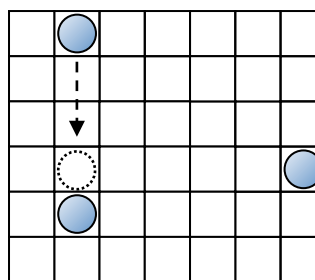
第一行為一整數 K ，代表啟動撞擊的次數 (請注意，被撞擊後再次發生撞擊的次數，亦即間接撞擊，不需計入)。接下來有 K 行，每行有兩個整數 A 和 B ，代表啟動編號 A 的球去直接撞擊編號 B 的球。

輸入檔範例 1

3
7 6
1 0
1 4
6 3

輸出檔範例 1

2
0 1
2 0

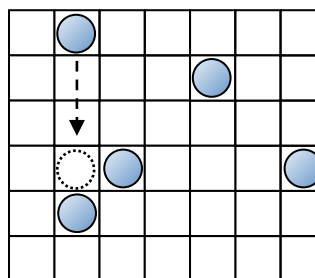


輸入檔範例 2

5
7 6
1 0
1 4
6 3
2 3
4 1

輸出檔範例 2

4
0 1
0 3
0 3
0 4



評分說明

本題共有六組測資。每組測資所輸出撞擊方式必須全部合法方能得分。

第一組測試資料， $N \leq 3, S_x \leq 6, S_y \leq 6$ ，最佳解得 20 分，否則 0 分；

第二組測試資料， $N \leq 6, S_x \leq 10, S_y \leq 10$ ，最佳解得 20 分，否則 0 分；

第三組至第六組測試資料， $N \leq 30, S_x \leq 1,000, S_y \leq 1,000$ ，每組得分計算方式為 $15 \times (N - R) / (N - R_{min})$ ， R 為檯面最後剩餘球數， R_{min} 為檯面最少可能剩餘球數，最多 15 分，四組共 60 分。

惡意軟體 (malware)

輸入檔: `input.txt`

輸出檔: `output.txt`

問題描述

某資訊安全公司推出雲端版防毒軟體，除了支援最新病毒碼雲端更新功能外，並藉此了解各類病毒感染的相關性，以便提供使用者更全面的保護。所以，此防毒軟體在使用者電腦偵測到任何一支病毒時，就會回報一筆資料給資訊安全公司的雲端資料處理平台，標明使用者電腦的 ID、檔案名稱及病毒種類等資訊。藉由這些資料分析，進一步找出那些病毒種類常常一起出現，例如使用者中了病毒 A 之後很容易會再中病毒 B 和病毒 C，也就是網路罪犯者常常會使用那些病毒組合來做攻擊。從每天收到的回報資料就可找出每天較常出現的病毒種類的組合為何，在雲端的防毒系統就可以把這些病毒解藥的組合同時給使用者，以便提供更完整的保護。你的目標就是要從每天收到的回報資料中，找出那些病毒常常一起出現，且出現的次數超過一定的門檻。

輸入檔格式 (`input.txt`)

第一行有三個整數， n 、 m 和 k ，並以空白分隔， n 代表有 n 個（行）使用者的中毒資料；而 m 代表最多有 m 種電腦病毒，但所有病毒不一定都會出現；而 k 代表系統預設病毒的出現次數門檻， $(1 \leq k \leq n)$ 。接下來是 n 行，第一個數字 i 為這個使用者今天中了幾種電腦病毒 $(1 \leq i \leq m)$ ，接下來由小到大依序列出這 i 種電腦病毒的代號 j ，並以空白分隔 $(1 \leq j \leq m)$ ，所以每一行會有 $(i+1)$ 個整數數字。

輸出檔格式 (`output.txt`)

請輸出兩個數字，並以空白隔開。第一個數字為計算出至少有 k 個使用者有一起出現不同病毒種類集合的個數。每個集合內病毒種類必須至少 2 種以上才符合題目規定的一起出現的病毒種類。第二個數字為最大的集合中病毒種類的個數。

輸入檔範例

```
4 5 3
2 1 3
3 1 2 3
3 1 2 3
5 1 2 3 4 5
```

輸出檔範例

```
4 3
```

說明

至少出現 3 次的集合有 (1,2)、(1,3)、(2,3) 和 (1,2,3)，總共找到 4 個集合，最大的集合內元素個數為 3，所以答案為 4 3

輸入檔範例

```
5 100 4
2 1 98
4 1 2 7 98
1 4
1 98
10 1 2 3 4 5 6 7 98 99 100
```

輸出檔範例

```
0 0
```

說明

因為最常出現的集合 (1, 98) 也只有 3 次，所以答案為 0 0

評分說明

本題共有 5 組測試資料，每組 20 分：

第一組為小型測試資料集， $1 \leq n \leq 100$ ， $1 \leq m \leq 10$ ，符合條件的不同病毒種類集合個數小於 20。

第二組為中型測試資料集， $1 \leq n \leq 1,000$ ， $1 \leq m \leq 24$ ，符合條件的不同病毒種類集合個數小於 150。

其它三組為大型測試資料集， $1 \leq n \leq 10,000$ ， $1 \leq m \leq 1,000$ ，但符合條件的不同病毒種類集合個數小於 4,000。

超級細菌 (bacteria)

輸入檔: `input.txt`

輸出檔: `output.txt`

問題描述

超級細菌為一種帶有可破壞抗生素基因的細菌, 近日科學家發現某種超級細菌具有快速變型的能力, 為了更了解這種超級細菌, 科學們家著手研究超級細菌各種變型與數量繁衍的關係, 以利追蹤並有效的控制細菌的傳播。經過實驗發現, 這種超級細菌變型及繁衍的速度非常快, 且每次新變型的超級細菌都有不同的繁衍速度, 例如一個新型的新超級細菌a經過 X 分鐘就能發展成為一個成熟的超級細菌a, 而每個成熟的超級細菌a每 Y 分鐘能繁衍出 Z 個新生的超級細菌a; 值得注意的是新生的超級細菌在長成為成熟的超級細菌後不會立即繁衍新生的超級細菌, 因此一個新生的超級細菌a必須經過 $X+Y$ 分鐘後才能第一次繁衍新生的超級細菌a, 因為經過 X 分鐘長成為成熟超級細菌a後, 必須再經過 Y 分鐘後才能第一次開始繁衍新生的超級細菌a。例如有一種新型的新超級細菌「奧林匹克一號」經過 2 分鐘就能發展為一個成熟的奧林匹克一號, 而每個成熟的超級細菌每經過 2 分鐘可以繁衍出 2 個新生的奧林匹克一號; 如下表所示, 若實驗一開始有一個成熟的奧林匹克一號及一個新生的奧林匹克一號, 經過一分鐘後, 新生的奧林匹克一號變成非成熟也非新生的奧林匹克一號, 但是成熟的奧林匹克一號也還沒有進行第一次的繁衍; 經過兩分鐘後, 成熟的奧林匹克一號進行第一次的繁衍, 且一開始新生的奧林匹克一號也變為成熟的奧林匹克一號, 因此在兩分鐘後奧林匹克一號的總數變為 4, 且成熟與新生的奧林匹克一號數量各為 2, 依此類推。

時間(分鐘後)	0(初始)	1分 鐘後	2分 鐘後	3分 鐘後	4分 鐘後	5分 鐘後	6分 鐘後
成熟的奧林匹克一號數量	1	1	2	2	4	4	8
新生的奧林匹克一號數量	1	0	2	0	4	0	8
所有奧林匹克一號總數	2	2	4	4	8	8	16

假設超級細菌在實驗過程中不會死亡, 為了預測超級細菌數量與感染致病的關係, 請您為科學家建立一個超級細菌數量與時間關係的估計系統, 以利進一步建立分析該超級細菌的感染致病模型; 值得注意的是, 此系統假設實驗一開始有 M 個成熟的超級細菌及 N 個新生的超級細菌, 並依此假設預測 W 分鐘後所有超級細菌的總數。

輸入檔格式 (input.txt)

輸入檔有 1 行, 共有 6 個整數, 分別為 M 、 N 、 W 、 X 、 Y 、 Z , 每個整數以一個空白隔開。第一個數字 M 代表初始時間 0 分鐘時所有成熟的超級細菌的個數, 第二個數字 N 代表初始時間 0 分鐘時所有新生的超級細菌的個數, 第三個數字 W 代表要估計實驗進行到 W 分鐘後的超級細菌總數, 第四個數字 X 代表一個新生的超級細菌長成為成熟的超級細菌所需要的時間, 第五個數字 Y 代表一個成熟的超級細菌要經過多少時間才能繁衍一次。第六個數字 Z 代表一個成熟的超級細菌每次可以繁衍多少個新生的超級細菌。

輸出檔格式 (output.txt)

請根據輸入檔的資料，輸出在 W 分鐘後所有的超級細菌總數。

輸入檔範例 1

1 1 6 2 2 2

輸出檔範例 1

16

輸入檔範例 2

1 0 6 2 1 2

輸出檔範例 2

37

評分說明

本題共有 3 組測試資料。

第一組測試資料 $0 \leq M, N \leq 6$ 、 $1 \leq W, X, Y, Z \leq 6$ ，且細菌總數 ≤ 100 ，共 20 分；

第二組測試資料 $0 \leq M, N \leq 40$ 、 $1 \leq W, X, Y, Z \leq 40$ ，且細菌總數 $\leq 10^9$ ，共 40 分；

第三組測試資料 $0 \leq M, N \leq 100$ 、 $1 \leq W, X, Y, Z \leq 100$ ，且細菌總數 $\leq 2 \times 10^9$ ，共 40 分。

任意門 (Transporter)

輸入檔: `input.txt`

輸出檔: `output.txt`

問題描述

給定一個 $M \times N$ 迷宮，其中部分方格設有障礙物，部分則設有陷阱。其他剩餘的位置可用以設置「空間任意門」，用以將遊戲者轉換至迷宮內另一位置。但是任意兩個「空間任意門」不能在同一排或同一列出現，除非兩者之間另有障礙物存在。請寫一個程式計算迷宮內最多可以設置幾個任意門。

輸入檔格式 (`input.txt`)

輸入檔第一行，有兩個整數， M, N ，以一個空白隔開。接下來有 M 列，每列 N 個 0, 1, 或 2 的數字，兩個數字之間以一個空白隔開，用以代表該迷宮每個位置目前配置狀態。其中 0 代表該位置目前可用以設置空間任意門，1 代表該位置設有陷阱，2 代表該位置設有障礙物。

輸出檔格式 (`output.txt`)

給定輸入檔的迷宮，輸出可設置「空間任意門」的最大數量。

輸入檔範例 1

```
3 4
2 0 0 0
2 2 2 1
0 1 0 2
```

輸出檔範例 1

```
2
```

輸入檔範例 2

```
1 3
2 1 2
```

輸出檔範例 2

```
0
```

評分說明

本題共有四組測試資料。

第一組測試資料 $1 \leq M, N \leq 10$ ，共 20 分；

第二組測試資料 $1 \leq M, N \leq 50$ ，共 30 分；

第三組測試資料 $1 \leq M, N \leq 100$ ，共 30 分；

第四組測試資料 $1 \leq M, N \leq 200$ ，共 20 分。