

104學年度高級中學資訊學科能力競賽決賽

上機程式設計競賽注意事項

- 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之原始程式格式、檔名、輸入方式及輸出方式等規定。若未遵守，該題將可以 **0** 分計算。
- 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、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。程式執行時，針對每組測試資料執行時間**依題目規定為準**。每題可使用記憶體空間**除題目有另外規定外**，以 **128MB** 為限。執行時間與使用記憶體空間以評分主機為準。執行時間或使用記憶體空間超過規定者，視同未完成，該組測試資料得分將以 **0** 分計算。
- 四、本次競賽學生需上傳原始程式碼 (.c, .cpp)。除題目另有規定外，輸入資料則自**標準輸入 (Standard input)** 讀取，計算結果須輸出至**標準輸出 (Standard output)**。注意：**請勿自行讀寫檔案**，若因此造成線上評分系統無法評分，該次評分結果將以 **0** 分計算。若藉此妨礙線上評分系統運行，承辦單位得取消比賽資格。
- 五、線上評分系統使用的編譯器與試場電腦所安裝的編譯器版本不一定一致，**以評分系統為主**。
- 六、請用承辦單位提供之**個人電腦內硬碟或網路儲存空間**備份原始程式碼。若由於任何原因需更換電腦時，僅能將備份之原始程式碼複製至新電腦，或下載已上傳至評分主機的程式碼。
- 七、對考題有任何疑義，請以評分系統之提問功能 (Communication) 提問 (請參考評分系統之使用手冊) 或填寫「提問單」交付監試人員轉送命題委員提出問題。監試人員不負責解答任何有關試題的問題。
- 八、競賽時間結束後，**參賽選手應留在位置上**等待評審前來共同確認評審結果。
- 九、試場守則：
 - 甲、試場內**不得與其他選手討論、上傳與解題無關檔案或惡意干擾線上自動評分系統**運作，承辦單位得依據違規狀況予以警告或取消比賽資格。
 - 乙、考試進行期間，選手可至試場外飲水、用餐、如廁，但**不得與他人研討試題**，違者得取消比賽資格。
 - 丙、離開試場，返回試場時，均須於試場入口填寫「離場休息登記表」。
 - 丁、離開試場時間，除午餐為 **1** 小時外，皆不可超過 **10** 分鐘。
 - 戊、考試時間用餐點、如廁均不可**離開工程三館三樓電梯北側區域**。
 - 己、午餐便當將於 11:30~12:30 三樓電梯門口旁供應。
 - 庚、午餐便當如未能及時食用，請於 14:30 前至三樓電梯門口領取，可使用微波爐加熱服務。
 - 辛、參賽中途如遇緊急事件(如停電等)，因應措施得依據命題及評審委員會決議處理。

(本頁空白)

一、雲端列印

時間限制：1.5 秒

問題敘述：

雲端列印服務公司提出一個新型服務。該公司有 n 台 3D 印表機，其中印表機 P_1, P_2, \dots, P_k 用以優先服務最為重要客戶，印表機 $P_{k+1}, P_{k+2}, \dots, P_n$ 列印速度較慢，用以優先服務一般客戶。每個客戶依該年度所選擇服務等級及所繳交費用可有不同的列印優先權，以 $1, \dots, 10000$ 表示之； 10000 代表最高列印優先權， 1 代表最低列印優先權。

為了不讓低列印優先權的客戶永無止盡的等待，印表機 P_1, P_2, \dots, P_k 一旦有空，等待的工作中優先權最高的工作就會被交付列印；而印表機 $P_{k+1}, P_{k+2}, \dots, P_n$ 一旦有空，等待的工作中優先權最低的工作就會被交付列印。

請寫一個程式列舉交付列印工作的順序。

輸入說明：

輸入只有一行，共有不定數量的整數，整數可為 $\{-2, -1, 0, 1, 2, \dots, 10000\}$ ，兩整數之間以一個空白隔開。 -2 表示印表機 P_1, P_2, \dots, P_k 其中一台有空，可以列印最高優先權的工作； -1 表示印表機 $P_{k+1}, P_{k+2}, \dots, P_n$ 其中一台有空，可以列印最低優先權的工作； $1, 2, \dots, 10000$ 代表新增一個優先權為該數字之工作； 0 則代表輸入結束。若輸入為 -1 或 -2 但無等待列印的工作，則不列印，需等待下一個 -1 或 -2 才再列印新的工作。

輸出說明：

請依被列印工作的順序，輸出該工作的優先權代號，之後緊接著一個空白。尚未交付列印的工作不需輸出。

評分說明：

本題共有五組測試資料。每組可有多個輸入檔案，全部答對該組才得分。

第一組 15 分，最多 11 個工作需要被列印，且只有一次交付列印指令。

第二組 15 分，最多 20 個工作需要被列印，且只有二次交付列印指令。

第三組 20 分，最多 50 個工作需要被列印，且最多有 25 次交付列印指令。

第四組 20 分，最多 15,000 個工作需要被列印或交付列印。

第五組 30 分，最多 500,000 個工作需要被列印或交付列印。

輸入範例 1：

20 15 10 -2 -1 -1 0

輸出範例 1：

20 10 15

輸入範例 2：

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

輸出範例 2：

3 1

注意事項：

使用 C++ 作答的同學，請在程式碼開頭加上 `#include <cstdio>`，並利用 `scanf` 讀入資料。使用 `cin` 讀入資料可能會因為讀入效率太差以致於程式執行時間超過限制。`scanf` 常用的讀入方式如下：

`scanf("%d", &x);` 讀入一個有號整數至 `int` 型態變數 `x`。

`scanf("%lld", &y);` 讀入一個有號整數至 `long long` 型態變數 `y`。

`scanf("%u", &x);` 讀入一個無號整數至 `unsigned int` 型態變數 `x`。

`scanf("%llu", &y);` 讀入一個無號整數至 `unsigned long long` 型態變數 `y`。

二、還原密碼

時間限制：1 秒

問題敘述：

蓋茲是一位著名的寶藏獵人，他花了兩年的時間追尋一批從漢朝流傳下來的寶藏。去年他在西安郊區的寺廟中得到一條線索，包含一段文字「數字加總，反覆為之，止於一位，謂之為根。」以及一個數字 5。今年他又在洛陽的石窟裡找到另一段文字「增添一位，其根相符，不為最大，不為最小。」和一組數字 138。上個月他找到跟這批寶藏有關的寶藏盒，可是寶藏盒還需要兩組四個數字的密碼才能打開。

他思考了很久，總算參透了這兩條線索的意思。

第一條線索所說的，是把一組數字的每個數字加總起來，反覆操作，直到變成一位數字，稱之為根。例如數字 138 會變成 $1+3+8=12$ 再變成 $1+2=3$ ，3 便稱為 138 的根。

第二條線索所說的，是要把 138 加上一位數字，讓此組數字的根為第一條線索說的 5。符合根為 5 的數字組合有四個，分別是 2138，1238，1328 和 1382。不是最大也不是最小的組合是 1328 和 1382，蓋茲嘗試了這兩組密碼，果然就打開了寶藏盒。

現在要請你寫一個程式進行類似上述的密碼還原工作。

輸入說明：

每筆測資有兩行。

第一行有兩個整數值，以一個空白字元隔開。第一個整數 N ($3 \leq N \leq 30$) 代表密碼有幾個數字。第二個數為根 R ($0 \leq R \leq 9$)。

第二行有連續 $(N-1)$ 個數字 $d_1d_2d_3\dots d_{(N-1)}$ ，數字 $d_i \in \{0, 1, 2, \dots, 9\}$ ， $(1 \leq i \leq N-1)$ 。

輸出說明：

依據輸入，由小到大輸出所有可能的密碼，每組密碼輸出於獨立的一行。(可假設至少有一組可能的密碼。)

評分說明：

本題共有三組測試資料。每組可有多個輸入檔案，全部答對該組才得分。

第一組 10 分，所有的測資 $3 \leq N \leq 5$ ，且輸入和輸出都不會有數字 0。

第二組 20 分，所有的測資 $3 \leq N \leq 5$ 。

第三組 70 分，所有的測資 $3 \leq N \leq 30$ 。

輸入範例 1：

3 6
12

輸出範例 1：

132

輸入範例 2：

4 5
138

輸出範例 2：

1328
1382

輸入範例 3：

5 4
0011

輸出範例 3：

00121
00211
02011

三、框架區間

時間限制：10 秒

問題敘述：

基因是含有特定遺傳信息的結構，用來決定生物的種性特徵。生物學家發現，與特定功能相關的一群基因在基因序列上的位置通常十分靠近，因此在基因序列中的連續片段被認為是有意義的。一個包含 n 個基因的序列可以用 $\{1, 2, \dots, n\}$ 所組成的排列 $S = (s_1, s_2, \dots, s_n)$ 來表示。為了預測基因序列 S 上可能有意義的片段，一位生物學家遭遇了下列問題。令 $F(a, b)$ 代表在基因序列 S 上位置落在基因 a 和基因 b 之間的所有整數所構成的集合 (含 a 和 b)。例如，令 $S = (2, 7, 6, 4, 14, 13, 5, 8, 1, 9, 11, 10, 12, 3)$ ，則 $F(6, 8) = F(8, 6) = \{6, 4, 14, 13, 5, 8\}$ 。令 $I(a, b)$ 代表數線上 a 和 b 這兩個整數間所有整數所構成的集合 (含 a 和 b)。例如， $I(6, 8) = I(8, 6) = \{6, 7, 8\}$ 。在基因序列 S 上如果兩個整數 a 和 b ， $1 \leq a < b \leq n$ ，滿足 $F(a, b) = I(a, b)$ 則稱 (a, b) 構成一個「框架區間」(framed interval)。舉例來說，考慮基因序列 $S = (2, 7, 6, 4, 14, 13, 5, 8, 1, 9, 11, 10, 12, 3)$ ，以 $(a, b) = (9, 12)$ 為例，因為 $F(9, 12) = \{9, 11, 10, 12\} = \{9, 10, 11, 12\} = I(9, 12)$ ，所以 $(9, 12)$ 是一個框架區間。相同的 $(6, 7)$ 、 $(10, 11)$ 和 $(13, 14)$ 也是框架區間。

這位生物學家想知道給定一個基因序列 S ，有多少數對 (a, b) ， $1 \leq a < b \leq n$ ，是一個框架區間？例如，在基因序列 $S = (2, 1, 5, 4, 3)$ 上，是框架區間的數對 (a, b) ， $1 \leq a < b \leq 5$ ，有 $(1, 2)$ 、 $(3, 4)$ 、 $(3, 5)$ 和 $(4, 5)$ ，共四個。

輸入說明：

第一行有一整數 T ，代表有 T 組測試資料。接下來每兩行用來描述一組基因序列，第一行有一整數 n ，第二行有 n 個整數 s_1, s_2, \dots, s_n (數字之間以一個空白隔開)，代表基因序列 $S = (s_1, s_2, \dots, s_n)$ ，任兩個數字都不相同且 $1 \leq s_1, s_2, \dots, s_n \leq n$ 。

輸出說明：

針對所輸入的基因序列 S ，輸出一個數字，代表有多少數對 (a, b) ， $1 \leq a < b \leq n$ ，是一個框架區間？

評分說明：

本題共有三組測試資料。每組可有多個輸入檔案，全部答對該組才得分。

第一組 30 分，所有的測資 $T \leq 20$ 、 $1 \leq n \leq 100$ 。

第二組 30 分，所有的測資 $T \leq 6$ 、 $1 \leq n \leq 3000$ 。

第三組 40 分，所有的測資 $T \leq 20$ 、 $1 \leq n \leq 5000$ 。

輸入範例 1：

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

輸出範例 1：

```
0
3
4
```

輸入範例 2：

```
2
14
2 7 6 4 14 13 5 8 1 9 11 10 12 3
11
3 10 4 5 6 8 7 9 11 2 1
```

輸出範例 2：

```
4
9
```


四、彩色紙條

時間限制：1 秒

問題敘述：

雷文想要把一張紙條用油彩塗上顏色。這張紙條上面有 N 個格子，依序編號為 1 到 N ，一開始都是白色(編號為 0)，而雷文不喜歡白色，想要用編號為 1 到 M 的 M 種顏色塗紙條，並把其中第 i 格塗上某個非白色的顏色 c_i (編號大於 0 而不超過 M)。

雷文上色的時候，每次可以一筆畫把連續若干格塗上顏色，並且覆蓋過原先塗在格子上的顏色。舉例來說，如果一個紙條上面有 5 格，顏色依序為(1,1,1,2,2)，那麼雷文可以一筆畫把第 2 格到第 4 格塗上顏色 3，使紙條顏色變成(1,3,3,3,2)。雷文決定好自己想要把紙條塗上那些顏色之後，想要用最少的次數完成塗色。例如雷文想要把有 5 個的紙條塗上(1,2,3,2,1)，最少需要三次：先把第 1 格到第 5 格塗上顏色 1，再把第 2 格到第 4 格塗上顏色 2，最後把第 3 格塗上顏色 3。過程如下：

$$(0,0,0,0,0) \rightarrow (1,1,1,1,1) \rightarrow (1,2,2,2,1) \rightarrow (1,2,3,2,1)$$

這是最少次數的塗法。請幫他撰寫一個程式，依據雷文的喜好，算出最少要塗幾次才能夠完成。

輸入說明：

輸入包含多筆測試案例。整個測試資料的第一行有一個整數 T ，代表總共有 T 個測試案例。每個測試案例有兩行，第一行是紙條上的格子數 N 以及顏色數 M ，由空白隔開。第二行有 N 個正整數 c_1, c_2, \dots, c_N ，由空白隔開。

輸出說明：

針對每個測試案例，以一行輸出最少的塗色次數。

評分說明：

本題共有五組測試資料。每組可有多個輸入檔案，全部答對該組才得分。

第一組 10 分， $T \leq 20$ 、 $N \leq 10$ 、 $M = 2$ 、對所有 $1 \leq i \leq N$ ， c_i 均為 1 或 2。

第二組 10 分， $T \leq 20$ 、 $N \leq 10$ 、 $M \leq 10$ 、對所有 $1 \leq i \leq N$ ， c_i 介於 1 到 M 之間。

第三組 20 分， $T \leq 20$ 、 $N \leq 50$ 、 $M = 2$ 、對所有 $1 \leq i \leq N$ ， c_i 均為 1 或 2。

第四組 20 分， $T \leq 20$ 、 $N \leq 50$ 、 $M \leq 50$ 、對所有 $1 \leq i \leq N$ ， c_i 介於 1 到 M 之間。

第五組 40 分， $T \leq 20$ 、 $N \leq 200$ 、 $M \leq 200$ 、對所有 $1 \leq i \leq N$ ， c_i 介於 1 到 M 之間。

輸入範例 1：(第一組、第三組)

2
5 2
1 2 1 2 1
5 2
1 1 1 2 2

輸出範例 1：

3
2

輸入範例 2：(第二組、第四組、第五組)

2
5 3
1 2 3 2 1
5 3
1 2 3 2 3

輸出範例 2：

3
4

五、網路測試的難題

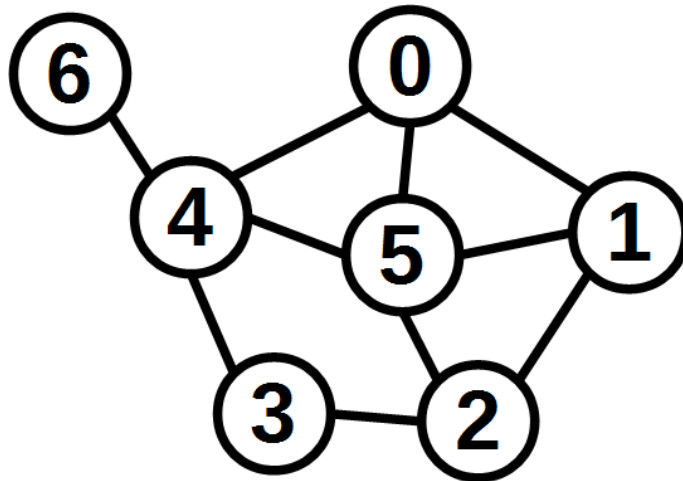
時間限制：8 秒

問題敘述：

骨幹網路是網際網路中重要的基礎設施。骨幹網路由多台伺服器組成，彼此間以高速網路連結；其他電腦與裝置，則透過骨幹網路連線到網際網路上。機關、學校、公司與資料中心也經常透過建置自己的骨幹網路，來提供內部與外部的網路服務。

骨幹網路中的伺服器通常透過不只一條的實體連線相互連結，提供妥善的分流與備援。確保每條連線的品質，就是確保骨幹網路穩定的基本方法，然而有效率的測試所有連線可能是相當艱鉅的任務。假設每條直接連線都是雙向的，考慮以下的簡單方法：「每台伺服器都有自己的唯一 ID 識別碼。每台伺服器查看所有與自己直接連線的伺服器 ID，如對方的 ID 比自己的 ID 大，則 ID 較小的伺服器負責對此連線進行測試。」

如此一來，每條直接連線都會被測試恰好一次。然而 ID 較小又擁有較多直接連線的伺服器可能得負擔較多的測試，會導致測試工作的分配並不是很平均。我們能靠重新分配伺服器的 ID，以減少(最小化)伺服器最大的負擔。例如以下這張網路：



需要的測試如下：

伺服器 0 → {1, 4, 5}

伺服器 1 → {2, 5}

伺服器 2 → {3, 5}

伺服器 3 → {4}

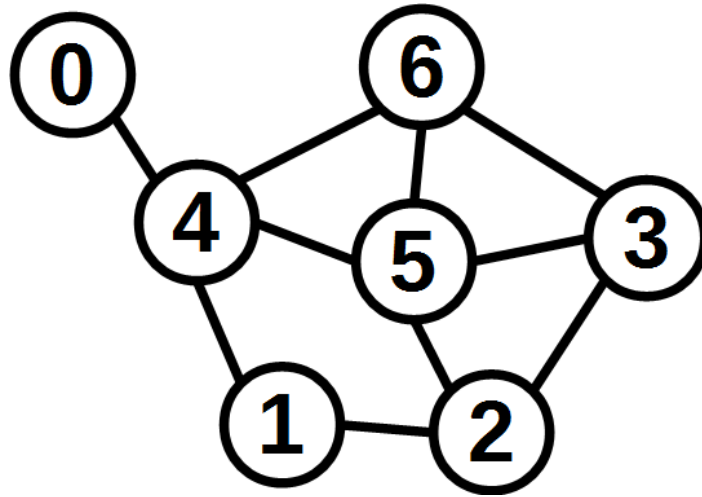
伺服器 4 → {5, 6}

伺服器 5 → {}

伺服器 6 → {}

伺服器 0 最多需要 3 次測試。

若 ID 經過適當地重新分配後：



需要的測試如下：

伺服器 0 → {4}

伺服器 1 → {2, 4}

伺服器 2 → {3, 5}

伺服器 3 → {5, 6}

伺服器 4 → {5, 6}

伺服器 5 → {6}

伺服器 6 → {}

伺服器 1,2,3,4 最多只需要 2 次測試。

可見適當的重新分配 ID，可以使最多的連線測試次數減少。對於給定的網路關係圖，請問重新分配 ID 後，每台伺服器最多連線測試次數可以減至多少？

輸入說明：

輸入第一行為一個整數 T ($T \leq 10$)，以下包含 T 組測資。

每組測資的第一行有以空白分隔的兩個整數 N 與 E ； N 是伺服器的數量， E 表示直接連線的數量。

往後有 E 行。每行包含以空白分隔的兩個整數 u 與 v ($0 \leq u, v < N$)，表示伺服器 u 與 v 之間有直接連線。任一伺服器不會直接連線到自己，任兩伺服器間不會出現重複的連線，但不保證任兩伺服器之間有連通。

輸出說明：

每筆測資有一行輸出，輸出為空白分隔的兩個整數，分別代表重分配 ID 之前與之後的最大的連線測試數量。

評分說明：

本題共有五組測試資料。每組可有多個輸入檔案，全部答對該組才得分。

第一組 15 分， $N \leq 100$ 、 $E \leq 500$ 。

第二組 18 分， $N \leq 5,000$ 、 $E \leq 80,000$ 。

第三組 20 分， $N \leq 50,000$ 、 $E \leq 200,000$ 。

第四組 22 分， $N \leq 200,000$ 、 $E \leq 400,000$ 。

第五組 25 分， $N \leq 500,000$ 、 $E \leq 800,000$ 。

輸入範例：

```
1
7 10
4 6
4 0
4 5
4 3
5 0
5 1
5 2
3 2
1 0
1 2
```

輸出範例：

```
3 2
```

注意事項：

使用 C++ 作答的同學，請在程式碼開頭加上 `#include<cstdio>`，並利用 `scanf` 讀入資料。使用 `cin` 讀入資料可能會因為讀入效率太差以致於程式執行時間超過限制。`scanf` 常用的讀入方式如下：

`scanf("%d",&x);` 讀入一個有號整數至 `int` 型態變數 `x`。

`scanf("%lld",&y);` 讀入一個有號整數至 `long long` 型態變數 `y`。

`scanf("%u",&x);` 讀入一個無號整數至 `unsigned int` 型態變數 `x`。

`scanf("%llu",&y);` 讀入一個無號整數至 `unsigned long long` 型態變數 `y`。

(本頁空白)

六、絕對多數

時間限制：6 秒

記憶體限制：20 MB

問題敘述：

在一群 M 個數字中，如果有某個數字出現的次數超過 $M/2$ ，則稱為此數為這群數的「絕對多數」，一群數字最多有一個絕對多數，但也可能沒有，例如 $(1,2,1,1,3)$ 的絕對多數是 1，而 $(1,2,1,3)$ 沒有絕對多數，本題的任務是計算絕對多數。

每一筆測試資料包含一個 $N \times N$ 的整數方陣 A 與 q 個計算要求，矩陣的元素儲存在 $A[0][0] \sim A[N-1][N-1]$ ，矩陣的橫的一排稱為列(row)，直的一排稱為行(column)。每一個計算要求包含四個整數 r_1 、 r_2 、 c_1 、 c_2 ， $0 \leq r_1 \leq r_2 < N$ ， $0 \leq c_1 \leq c_2 < N$ ，代表一個子矩陣 $A[r_1:r_2][c_1:c_2]$ ，也就是第 r_1 列到第 r_2 列且第 c_1 行到第 c_2 行所組成的子矩陣，我們要計算這個子矩陣中是否有絕對多數。下圖為一個 $N=5$ 的方陣 A ，子矩陣 $A[0:1][0:2]$ 共有六個元素，出現最多的元素是 1，出現了 3 次，並未超過總數的一半，因此沒有絕對多數。而子矩陣 $A[2:4][1:3]$ 共有 9 個元素，3 出現了 5 次，超過總數的一半，因此絕對多數是 3。

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

對於某些子題，你的程式必須很有效率的決定是否存在絕對多數。

輸入說明：

輸入包含多個測試資料，每筆測試資料包含一個輸入矩陣以及若干個子矩陣計算要求。每個測試資料的第一行(line)是矩陣大小的 N ，接下來 N 行(lines)是矩陣內容，以 row-major 方式排列，由第 0 列(row)至第 $N-1$ 列，每列有 N 個非負整數是該列第 0 行(column)至第 $N-1$ 行(column)，數字之間以一個空白間格。接著一行包含一個整數 q 代表計算要求數，其下有 q 行，每行是一個計算要求依序是子矩陣範圍的四個整數 r_1, r_2, c_1, c_2 。一筆測試資料結束後是下一筆測試資料，若 $N=0$ 代表輸入資料結束。

輸出說明：

針對每個測試資料的每個計算要求，以一行輸出絕對多數，如果沒有絕對多數則輸出 -1。

評分說明：

本題共有四組測試資料。每組可有多個輸入檔案，全部答對該組才得分。每組測試資料的輸入檔參數 N 、 D 、 Q 和 S 如下，其中 N 是矩陣大小的上限， D 代表陣列中數字的上限， Q 是計算需求總數即該檔案中 q 值的總和，也就是該子題總共輸出 Q 行， S 則是輸入檔案中，所有矩陣的元素總量，即 N^2 的總和。

第一組 15 分， $N \leq 100$ 、 $D \leq 5000$ 、 $Q \leq 40$ 、 $S \leq 32000$ 。

第二組 27 分， $N \leq 1010$ 、 $D \leq 5000$ 、 $Q \leq 30$ 、 $S \leq 3200000$ 。

第三組 19 分， $N \leq 1010$ 、 $D \leq 2^{31}-1$ 、 $Q \leq 50$ 、 $S \leq 3200000$ 。

第四組 39 分， $N \leq 2000$ 、 $D \leq 2^{31}-1$ 、 $Q \leq 70$ 、 $S \leq 12500000$ 。

輸入範例：

```
5
1 2 3 3 2
2 1 1 2 3
1 2 3 1 1
0 2 3 3 1
0 3 1 3 2
2
0 1 0 2
2 4 1 3
0
```

輸出範例：

```
-1
3
```

注意事項：

使用 C++ 作答的同學，請在程式碼開頭加上 `#include<cstdio>`，並利用 `scanf` 讀入資料。使用 `cin` 讀入資料可能會因為讀入效率太差以致於程式執行時間超過限制。`scanf` 常用的讀入方式如下：

`scanf("%d", &x);` 讀入一個有號整數至 `int` 型態變數 `x`。

`scanf("%lld", &y);` 讀入一個有號整數至 `long long` 型態變數 `y`。

`scanf("%u", &x);` 讀入一個無號整數至 `unsigned int` 型態變數 `x`。

`scanf("%llu", &y);` 讀入一個無號整數至 `unsigned long long` 型態變數 `y`。

七、保全公司的危險加給

時間限制：8 秒

問題敘述：

某保全公司有 N 位保全人員(以 $1, 2, \dots, N$ 表示)。這家公司白天要負責 N 個地區的守衛工作，晚上也要負責另外 N 個地區的守衛工作。保全公司對每個地區都有評估其危險程度值，此數值是一個正整數。白天要守衛的 N 個地區，其危險程度值分別記為 x_1, x_2, \dots, x_N ；晚上要守衛的 N 個地區，其危險程度值分別記為 y_1, y_2, \dots, y_N 。每位保全人員需負責白天時段某個地區和晚上時段某個地區的守衛工作，每一個地區都必需要有保全人員(且剛好一位)來負責守衛。保全公司同意一個保全人員如果從事危險的工作，則會支付他(她)合理的危險加給，但又不願意保全人員為賺取危險加給而選擇太危險工作。所以，這家保全公司制定了以下的規定：

規定一、假設保全人員白天工作地區的危險程度為 x ，而晚上工作地區的危險程度為 y ，則其全天的工作危險程度為 $x+y$ 。如果 $x+y$ 小於 L ，則保全公司不支付任何危險加給；如果 $x+y$ 介於 L 和 U 之間(亦即 $L \leq x+y \leq U$)，則保全公司支付危險加給 $x+y-L$ 。如果 $x+y$ 大於 U ，則保全公司將支付危險加給 $U-L$ 。

另一方面，為了避免保全人員來回奔波，保全公司也制定了另一條規定：

規定二、如果白天工作地區 i (其危險程度為 x_i) 和晚上工作地區 j (其危險程度為 y_j) 距離太遠，則不允許保全人員一天同時在 i 和 j 兩地區工作。我們稱 i 和 j 為禁止配對。

給定 N 個白天工作地區的危險程度值、 N 個晚上工作地區的危險程度值、 L 、 U 及集合 F 包含 K 個禁止配對($0 \leq K \leq N^2$)，請寫一個程式幫保全公司計算所要支付的最少危險加給費用為何？若條件造成無法讓每位保全人員都能順利分配到白天和晚上各一個守衛地區，則請輸出 no。

範例說明：

範例一、

假設 $N=5$ 、

$x_1=4$ 、 $x_2=1$ 、 $x_3=3$ 、 $x_4=2$ 、 $x_5=5$ 、

$y_1=9$ 、 $y_2=7$ 、 $y_3=8$ 、 $y_4=10$ 、 $y_5=6$ 、

$L=2$ 、 $U=100$ 、 $F=\emptyset$ (即 $K=0$)，

則輸出 45。

範例二、

假設 $N=5$ 、

$x_1=4$ 、 $x_2=1$ 、 $x_3=3$ 、 $x_4=2$ 、 $x_5=5$ 、

$y_1=9$ 、 $y_2=7$ 、 $y_3=8$ 、 $y_4=10$ 、 $y_5=6$ 、

$L=2$ 、 $U=100$ 、 $F=\{(1,1),(1,2),(1,3),(1,4),(1,5)\}$ (即 $K=5$)，則輸出 no。

輸入說明：

第一行有一個數字 T 代表測試資料的數目， $T \leq 8$ 。每組測資的第一行有 4 個數字，代表 N 值、 L 值、 U 值與 K 值，任兩個數字以空白隔開。第二行起接下來 K 行代表 K 個禁止配對，每個禁止配對有二個數字(兩個數字間以空白隔開)：第一個數字代表白天的地區；第二個數字代表晚上的地區。每組測資的最後兩行分別代表白天要守衛的 N 個地區的危險程度值及晚上要守衛的 N 個地區的危險程度值，兩行中的第一行有 N 個數字(任兩個數字以空白隔開)代表 x_1, x_2, \dots, x_N ，兩行中的第二行也有 N 個數字(任兩個數字以空白隔開)代表 y_1, y_2, \dots, y_N 。

輸出說明：

針對所輸入的資料，輸出一個正整數代表保全公司所要支付的最少危險加給。若條件造成無法讓每位保全人員都能順利分配到白天和晚上各一個守衛地區，則輸出 no。

評分說明：

本題共有五組測試資料。每組可有多個輸入檔案，全部答對該組才得分。

第一組 11 分：

- (1) $1 \leq N \leq 10$
- (2) $1 \leq x_1, x_2, \dots, x_N, y_1, y_2, \dots, y_N \leq 10$
- (3) $1 \leq L, U \leq 20$
- (4) $0 \leq K \leq 100$

第二組 13 分：

- (1) $1 \leq N \leq 100$
- (2) $1 \leq x_1, x_2, \dots, x_N, y_1, y_2, \dots, y_N \leq 100$
- (3) $1 \leq L \leq 1000$
- (4) $\max_{1 \leq i \leq N} x_i + \max_{1 \leq j \leq N} y_j \leq U$
- (5) $K=0$

第三組 21 分：

- (1) $1 \leq N \leq 1000$
- (2) $x_1 = x_2 = \dots = x_N = y_1 = y_2 = \dots = y_N = 1$
- (3) $1 \leq L \leq 1000$
- (4) $\max_{1 \leq i \leq N} x_i + \max_{1 \leq j \leq N} y_j \leq U$
- (5) $0 \leq K \leq 1000000$

第四組 25 分：

- (1) $1 \leq N \leq 500$
- (2) $1 \leq x_1, x_2, \dots, x_N, y_1, y_2, \dots, y_N \leq 10000$
- (3) $1 \leq L, U \leq 10000$
- (4) $0 \leq K \leq 250000$

第五組 30 分：

- (1) $1 \leq N \leq 500$
- (2) $1 \leq x_1, x_2, \dots, x_N, y_1, y_2, \dots, y_N \leq 10^{18}$
- (3) $1 \leq L, U \leq 9 \times 10^{18}$
- (4) $0 \leq K \leq 250000$

輸入範例 1：

```
2
5 2 100 0
4 1 3 2 5
9 7 8 10 6
5 2 100 5
1 1
1 2
1 3
1 4
1 5
4 1 3 2 5
9 7 8 10 6
```

輸出範例 1：

```
45
no
```

輸入範例 2：

```
3
5 6 100 0
1 7 8 9 10
1 2 3 4 5
5 6 100 1
1 5
1 7 8 9 10
1 2 3 4 5
5 6 100 2
1 4
1 5
1 7 8 9 10
1 2 3 4 5
```

輸出範例 2：

```
20
21
22
```

輸入範例 3：

```
2
5 2 90000000000000000002 0
400000000000000000003 400000000000000000003 4000000000000000002 4000000000000000002 4000000000000000002
500000000000000000003 500000000000000000003 5000000000000000002 5000000000000000002 5000000000000000002
5 20000000000000000 10000000000000000 5
1 1
1 2
1 3
1 4
1 5
40000000000000000 10000000000000000 30000000000000000 20000000000000000 50000000000000000
90000000000000000 70000000000000000 80000000000000000 10000000000000000 60000000000000000
```

輸出範例 3：

```
45000000000000000000
no
```