2015/05/16 全國大專 ITSA 盃程式設計桂冠挑戰大賽

A. 挑戰組試題

11 題，共 47 頁

題號	題目	Time Limit (secs)
A1	Minority Counts	2
A2	The Robots	1
A3	The Center of a Tree	3
A4	Pan's Labyrinth	3
A5	尋找最經濟的產品升級方式	2
A6	Spy Pass	3
A7	Taxi dispatch	3
A8	Electric Vehicle	2
A9	Lottery	2
A10	Satun’s Test	1
A11	Smart Patrol	6

A1

Minority Counts

(Time Limit: 2 seconds)

Problem Description

English

In an anonymous election, the majority usually wins. In order to avoid the majority force, there is a game called minority counts. Within a certain number of runs, each time the game takes the minority votes to next run. Those who survive to the last are the winners. For example, assume there are 3 runs and 10 players. For the first issue if there are 6 votes for "yes" and 4 votes for "no" then those who voted for "no" get to the next run. If there are 2 votes for "yes" and 2 votes for "no" to the second issue, then no one is eliminated in the second run. If there are 3 votes for "yes" and 1 vote for "no" to the third issue, then the one who vote for "no" is the winner. The crafty game holder wants to increase the suspense and mystery, so they do not announce the result for each run until the game is over. Hence each player will have to vote for every issue. Please help the game holder to write a program to decide the winners of the game.

Chinese

在不計名的投票中，通常是多數者贏。為了避免多數暴力，有一種遊戲叫少數者的價值。在一定的回合數中，每進行一次投票，淘汰多數者而留下少數者得以進入下一回合，支持到最後的就是贏家。例如，假設 10 為參賽者進行 3 回合。對於第一個問題，如果有 6 個投“是”，4 個投“否”，那麼那些誰投票給“否”的參賽者可以進行下一回合。第二個問題如果有 2 票為“是”，2 票為“否”，那麼第二輪沒有淘汰任何人。第三個問題如果有 3 票為“是”，1 票為“否”，那麼那一個誰投“否”的就是贏家。詭計多端的遊戲主持人為了增加懸疑和神秘感，所以他們不在每回合宣布投票結果，直到遊戲結束。因此，每個玩家對每一個問題都會進行投票。請幫助遊戲主持人編寫一個程式來決定比賽的優勝者。

Technical Specification

- **English**
  - The number of players *p* is a positive integer and $1 \le p \le 1000$.
  - The number of runs *r* is a positive integer and $1 \le r \le 500$.
  - Every player has to vote for each run, they only vote for 'Y' as "yes" or 'N' as "no". There is no invalid ballot in any run.
  - The eliminated votes will not be counted in the later runs.
  - If everyone gives the same vote or the number of votes for positive and negative are the same, then no one gets eliminated from the game.
  - The game will definitely go for *r* runs. The winners are those who still survive after the last run.
- **Chinese**
  - 遊戲參賽者人數 *p* 介於 1 到 1000 之間。
  - 每次遊戲總共進行 *r* 回合，*r* 介於 1 到 500 之間。
  - 每位參賽者投 Y 表示是，N 表是否，沒有無效票。
  - 已被淘汰者的票在下一回合中不會被計算。
  - 當所有人都投相同的票或是兩方票數相同時，該回合不淘汰任何人。
  - 遊戲一定剛好進行 *r* 回合，還留在遊戲中的即是贏家。

Input Format

- **English**

The input consists of a number of test cases. The first line of the input file contains an integer indicating the number of test cases to follow. The first line of each test case contains two integers *p* (the number of players) and *r* (total number of runs) separated by a space. Then for the following *p* lines, each line *i* consists of *r* characters separated by a space, which are the votes for player *i* in each run.

- **Chinese**

第一行是一個整數代表測試資料有幾筆。每一筆測試資料的第一行有兩個正整數 *p*（參賽人數）及 *r*（遊戲回合數），以一個空格隔開。接下來有 *p* 行，每行有 *r* 個字元，每個字元間以一個空格隔開，第 *i* 行代表參賽者 *i* 在每一回合所投的票。

Output Format

- **English**

The output contains one line for each test case. Each line contains one or several integers to indicate the winners of the game. If there is more than one winner, output the winners in an increasing order.

- **Chinese**

針對每一筆測試資料，將贏家輸出於一行。如果有一個以上的贏家，將它們從小到大輸出，贏家之間以一個空格隔開。

Example

Sample Input:	Sample Output:
2 5 3 Y Y N Y Y N Y N Y Y N N Y Y Y 3 4 Y Y N N Y N Y N N Y Y N	3 4 3

A2

The Robots

(Time Limit: 1 second)

Problem Description

- **English**

Two robots are placed on a M (rows) x N (columns) grid, with x-coordinate from 0 to M-1 and y-coordinate from 0 to N-1. The first robot has F_1 units of fuel and the second robot has F_2 units of fuel. When a robot moves one unit on the grid, it will burn one unit of fuel. If there is not enough fuel, the robot cannot move. Given the first robot always moves North for N_1 time and then moves East for E_1 time. The second robot always moves East for E_2 time and then moves North for N_2 time. The two robots repeat their respective two moves until they run out of fuel.

Note: moving East will result in robot's x-coordinate increase by 1 and moving North will result in robot's y-coordinate increase by 1.

If a robot moves off the given grid, then it will wrap around and appear at the other end of the grid. For example, if given a M=7, N=6 grid, if a robot is at (5, 5), moving North by one unit will put the robot at (5, 0). If two robots move to the same grid point at the same time, the collision results in explosion. Given the initial position of the two robots (X_1, Y_1) and (X_2, Y_2), and their initial fuel status (F_1 and F_2), determine if the two robots will collide and explode.

- **Chinese**

假設有兩個機器人，在 M(水平) x N(垂直)單位的方格上行動。方格的 x 座標為 0, …, M-1, y 座標為 0, …, N-1。第一個機器人有 F_1 單位的燃料，而第二個有 F_2 單位的燃料，兩個機器人在一單位的時間內若還有足夠的燃料（需一單位的燃料才能移動），就都會移動一格，如果燃料不夠，就不會移動。 在一開始的 N_1 時間內第一個機器人會往北方移動，然後接下來的 E_1 的時間會往東；第一個機器人會重複這兩個步

驟直到耗光燃料。 第二個機器人的行動稍微不一樣，他在一開始的 E_2 時間會先往東移動，然後在接下來的 N_2 時間內往北移動，並且重複這兩個步驟直到耗盡燃料。

註記：往東移動相當於機器人 x 座標增加一單位；往北移動相當於機器人 y 座標增加一單位。

假如有某個機器人移動超出地圖範圍，他會在地圖上的另一端重新出現。 例如，如果 M = 7, N = 6, 機器人在(5, 5)並且往北走，他會重新在(5, 0)出現。另外，如果兩個機器人移動到同一個格子上，兩個就會撞在一起而爆炸。現在給定第一個機器人的起點(X_1, Y_1)，第二個機器人的起點(X_2, Y_2)，兩個機器人的燃料量(F_1 and F_2)，請判斷這兩個機器人會不會爆炸。

Input Format

English

First line of input contains an integer indicating the number of test cases to follow. For each test case, only one line of input containing the following 12 integers in sequence M, N, X_1, Y_1, E_1, N_1, F_1, X_2, Y_2, E_2, N_2, F_2. Constraints are as follows.

- 10000 > M, N > 0
- 0 <= X_1, X_2 < M
- 0 <= Y_1, Y_2 < N
- (X_1, Y_1) is not (X_2, Y_2)
- N_1, E_1, N_2, E_2 > 0
- 0 <= F_1, F_2 <= 10000

Chinese

輸入第一行有一整數代表測試組數。接下來每一行代表一組測試資料，每一行有下列 12 個整數 M, N, X_1, Y_1, E_1, N_1, F_1, X_2, Y_2, E_2, N_2, F_2, 且符合下列條件限制：

- 10000 > M, N > 0
- 0 <= X_1, X_2 < M

- $0 <= Y_1, Y_2 < N$
- (X_1, Y_1) is not (X_2, Y_2)
- $N_1, E_1, N_2, E_2 > 0$
- $0 <= F_1, F_2 <= 10000$

Output Format

English

For each test case, if robots collide and exploded, output on a single line “robots explode at time T” where T is the time of collision, else, output on a single line “robots will not explode”.

Chinese

針對每一組測資，如果機器人爆炸了，請輸出一行 "robots explode at time T"，T 是爆炸的時間。否則請輸出"robots will not explode"。

Example

Sample Input	Sample Output 1
2 7 6 2 0 9 2 100 3 5 2 7 100 7 6 2 0 9 2 6 3 5 2 7 0	robots explode at time 5 robots will not explode

A3

The Center of a Tree

(Time Limit: 3 seconds)

Problem Description

- **English**

For any two nodes in a graph, the length of a path connecting these two nodes is the number of edges in the path. The distance of two nodes is the length of a shortest path connecting these two nodes. The eccentricity of a node is the longest distance from it to any of the other nodes, and the center of a graph is the node with minimum eccentricity. A tree is a graph in which, for any two nodes, there is a unique path. In this problem, your task is to write a program to compute the center of a tree. If there is more than one center, output the one with the smallest label.

- **Chinese**

在一個圖形裡，一條連接任二點路徑的長度為此路徑的邊數，而二點間的距離為連接這二點的最短路徑之長度。一個點的 eccentricity 為該點到其他各點距離之最大者，一個圖形的中心點(center)則為具有最小 eccentricity 之點。一個樹是一個圖形，它的任二點具有一條唯一的路徑。本題的工作是要找出所給樹的中心點，如果中心點不只一個，請輸出編號最小的那一個。

Technical Specification

- **English**
  - All the given numbers are integers and there is no sign.
- **Chinese**
  - 每個所給數字皆為整數且無正負符號

Input Format

- **English**

The first line is an integer which indicates the number of test cases. Each test case contains *n* lines defining an *n*-node tree. The first line consists of an integer *n*, $1 \le n \le 10000$, which is the number of nodes in the tree. The nodes are given by their unique labels which are integers from 0 to *n*-1. In the next *n*-1 lines, each line consists of two numbers *x* and *y* which are separated by a space indicating a tree edge (*x*, *y*).

- **Chinese**

第一列是一個整數代表測試資料有幾筆。每一筆測試資料包含 *n* 列定義一個樹，第一列是一個數字 *n*，表示此樹的節點數，$1 \le n \le 10000$，每一個節點用一個數字唯一代表，*n* 個點的編號由 0 到 *n*-1，剩下的 *n*-1 列標示此樹的邊，每一列都含二個編號 *x* 和 *y*，以空白隔開，代表一個邊(*x*, *y*)。

Output Format

- **English**

For each test case, output the center in one line. If there is more than one center, output the one with the smallest label.

- **Chinese**

針對每一筆測試資料，輸出此樹的中心點編號，若中心點不只一個，請輸出編號最小者。

Example

Sample Input:	Sample Output:
2 5 0 1 1 2 1 3 3 4 8 0 2 1 2 2 3	1 3

3 4 3 5 5 6 6 7	

A4

Pan's Labyrinth（羊男的迷宮）

(Time Limit: 3 seconds)

Problem Description

- **English**

Do you know Pan? You may or may not, after all, it is hard to draw a conclusion on his existence. He is the eccentric shepherd god. No one knows the real purpose of his doings, only that He always has evil intentions. Pan's favorite pastime is to sabotage people's nightmares. He manipulates people's dreams, blurring the lines between dreams and reality to make the dreamer miserable. He likes to hide the key of dreamer's precious memories in the Labyrinth, and if the dreamer cannot find it before waking up, all of his most important memories will disappear without a trace. Then, he can only continue to wander in the nightmare, or give up all the important memories and wake up in despair. Yes, telling you about this is because you have entered the Labyrinth in a dream and your key of memories has been hidden somewhere in the Labyrinth by Pan. The Labyrinth has *n* * *m* rooms, each room with four doors namely East (E), South (S), West (W), and North (N). You don't know when or what will wake you up, and going through doors increase the risk of being waked up. So you want to avoid going through the doors as much as possible and reach the room where the key is hidden within the shortest time, otherwise the consequences could be a disaster! However, Pan didn't make it easy for you! The switch of each door is designed to confuse the dreamer. When you enter the room from different doors, it will trigger a switch so some of the doors become invisible.

The following figure is an illustration of Pan's Labyrinth. The northwest corner is the room (0, 0). The letter E on the upper, left, and bottom in room (0, 0) indicate if you enter the room from North, West, and South door, you can only see East door. The letter S on the right indicates that if you enter the room (0, 0) from east door, then you will see south door is open. Suppose you are pushed into Labyrinth at room (0, 0) from the north (N) door, while Pan tells you that your key is hidden in the room (2, 2). You can see only one door (E) in room (0, 0), so

you have no choice but to go from (0, 0) through E to arrive at (0, 1). At room (0, 1), you may choose from doors E and S. If you choose the east gate, according to the orientation of the door being opened you will find the path leading you sequentially through the doors and rooms E (0, 2), S (1, 2), W (1, 1), W (1, 0) respectively. If you choose the south gate instead, then your path will be S (1, 1), W (1, 0). Of course the south gate is a much better choice here. After reaching room (1, 0), there are doors S and N waiting for you to choose. Choosing N would send you back to room (0, 0) and start over, hence S is a better way to go. After S (2, 0), E (2, 1), and N (1, 1), you will then face three choices. Since W and N both lead to repeated paths, E is the right way to go. By going through E (1, 2), N (0, 2), W (0, 1), W (0, 0), S (1, 0), E (1, 1), again you face the choice between S and E. Since E will bring you back to the old path, you should choose S to reach (2, 1), then going through the east gate again you will finally arrive the room (2, 2). Counting the doors including the one leading into the Labyrinth, you will go through at least 15 doors to get to where your key of memory is hidden.

E E (0,0) S E	S ES (0,1) W E	S S (0,2) SW W
E ES (1,0) NS N	W SE (1,1) W NEW	W N (1,2) W NW
E NE (2,0) N N	E N (2,1) W E	W N (2,2) NW N

● Chinese

你知道羊男嗎？或許你知道，當然你也可能不知道，畢竟祂的存在與否是很難定論的。傳聞祂是一位行事詭異的牧羊神，沒有人知道祂所做任何一件事的真正目的，唯一知道的，就是祂總是心懷鬼胎。羊男最喜歡捉弄正在做惡夢的人們，祂操弄夢境，使現實與幻夢糾結，將開啟做夢者珍貴記憶的一把金鑰藏放在祂所創造的迷宮中，如果做夢者不能在醒來之前找到，所有他心中最重要的記憶都將消失無踪，而他只能選擇徘徊在可怕的夢境中，或是放棄所有重要的記憶在絕望中醒來。是的，告訴你這些，就是因為你已進入夢境，正身處在迷宮中的某處，羊男則是笑嘻嘻地把對你最重要的記憶之鑰，藏放在迷宮中的另一處。這個詭譎的迷宮有 n*m 個房間，每個房間有東(E)南(S)西(W)北(N)四道門，沒有人知道你甚麼時候會醒來，然而每經過

一道門，你被攪擾醒來的風險就會增加，所以你要盡可能的減少所需要經過的門，好在最短的時間內到達金鑰藏匿處，否則後果不堪設想！然而羊男設計了各門的開關，當人們從不同的門走入房間時，就會觸動開關使部分的門開啟而其餘的門隱藏不見，因此人們可能重複到達同一個區域卻不自知，或是被所看到的景象迷惑。

下圖是羊男迷宮的示意圖，最西北角的房間為(0,0)。圖中上方、左方及下方的 E 分別代表當你是從北門、西門、及南門進入(0,0)時，都只能看到東門是開啟的，右邊的 S 則表示從東門進入該房間時可以看到的有開的是南門。現在假設你是從北方的門被丟進下圖迷宮中(0,0)的位置，而羊男告訴你，你的記憶之鑰就藏在(2,2)。你會發現這個房間只有東門是開著的，所以毫無選擇的你從(0,0)經由東門(E)抵達(0,1)，這時你會發現有東門(E)和南門(S)必須作選擇，如果選擇東門(E)的話，會發現路徑會是 E(0,2)、S(1,2)、W(1,1)、W(1,0)，而選擇南門(S)的路徑則是 S(1,1)、W(1,0)，可見南門(S)是比較好的選擇。在(1,0)會面臨南門(S)與北門(N)的選擇，因北門(N)會回到(0,0)且只有重複同樣的路線，因此這時應該選擇南門(S)。之後你會依序經過的門和房間分別為 S(2,0)、E(2,1)、N(1,1)，此時看見北門(N)、東門(E)、西門(W)同時開啟，而如果選擇西門(W)或選擇北門(N)都將繞回之前走過的路徑，因此選擇東門(E)才是正確的。之後你會依序經過的門和房間分別為 E(1,2)、N(0,2)、W(0,1)、W(0,0)、S(1,0)、E(1,1)，此時在南門(S)與東門(E)之間若選擇東門(E)又會回到之前的路徑，因此選擇南門(S)到達(2,1)再通過東門(E)就抵達了記憶之鑰藏匿處(2,2)。如此一來包含你一開始被丟入迷宮的門在內，你至少會經過 15 道門才能到達記憶之鑰藏匿處。

E E (0,0) S E	S ES (0,1) W E	S S (0,2) SW W
E ES (1,0) NS N	W SE (1,1) W NEW	W N (1,2) W NW
E NE (2,0) N N	E N (2,1) W E	W N (2,2) NW N

Technical Specification

- **English**
  - There are *m* rows and *n* columns of rooms in the Labyrinth where $2 \le m$,

$n \le 10$. The room is denoted by coordinates (r, c) where the northwest corner is room (0, 0) and the southeast corner is room (m-1, n-1).

- There are four doors in each room, E represents the east door, S represents the south door, W represents the west door and N represents the north door. After entering the room, the doors you see will not lead you out of the Labyrinth.
- There are k dreamers in the Labyrinth, where $1 \le k \le 5$.
-

- **Chinese**
  - 迷宮中的房間排成 m 列 n 行，每個房間以座標(r, c)表示，西北角為(0,0)，東南角為(m-1, n-1)，且 $2 \le m, n \le 10$。
  - 每個房間均有東南西北四個門，分別以字母 E S W N 表示，進入房間後能見到的門不會走出迷宮之外。
  - 每個迷宮中夢遊者人數為 k，且 $1 \le k \le 5$。

Input Format

- **English**

The first line is an integer indicating the number of test cases. The first line of each test case contains two integers m and n to represent the size of the Labyrinth followed by $m * n$ rows. Each row has two integers r and c, and four strings, each separated by a space, representing the visible doors in the room (r, c) when entering the room from east, south, west and north door respectively. The next line is an integer k which indicates the number of dreamers in the Labyrinth, followed by k lines, each line containing a character d and four integers a, b, x, y, each separated by a space, to represent a dreamer thrown into the room (a, b) from door d and his key of memory in the room (x, y). The next test case is immediately followed.

- **Chinese**

第一行是一個整數代表測試資料有幾筆。每一筆測試資料第一行有兩個整數 m 與 n 代表迷宮大小。之後有 $m*n$ 行，每行有兩個整數 r 與 c 及四個字串，彼此間有一個空格隔開，四個字串依序代表分別由東南西北門進入迷宮房間(r,c)後可以看到的門。下一行有一個整數 k 表示共有 k 個做夢者被丟進這個迷宮。接著有 k 行，每行有一個字元 D 後面接著四個整數 a, b, x, y，代表做夢者由 D 門被丟進迷宮中房間

(a, b)，而他的記憶之鑰在房間(x, y)中。之後緊接著下一筆測資。

Output Format

- **English**

  For each test case, for each dreamer, output the minimum number of doors (including the door he entered the Labyrinth through) he has to go through in order to reach the destination in one line. After that, output a line of 10 '-' to indicate the end of that test case. If there is no way to reach the destination, output string "Are you kidding me?" in one line.

- **Chinese**

  針對每一筆測試資料，對每個做夢者輸出他至少需要經過多少個門（包含他所被丟入的門）才能抵達他的記憶之鑰所在處於一行。之後輸出 10 個'-'代表該筆測資結束。對無法抵達記憶之鑰房間的做夢者，輸出一行字串"Are you kidding me?"。

Example

Sample Input:	**Sample Output:**
2 3 3 0 0 S E E E 0 1 W E ES S 0 2 SW W S S 1 0 NS N ES E 1 1 W NEW SE W 1 2 W NW N W 2 0 N N NE E 2 1 W E N E 2 2 NW N N W 2 N 0 0 2 2 W 1 1 0 0	15 5 ---------- Are you kidding me? 3 ----------

3 4 0 0 S E E ES 0 1 W E S W 0 2 S W S S 0 3 SW W S W 1 0 N NE NS ES 1 1 NS N NS E 1 2 N N N S 1 3 N N N SW 2 0 N N E E 2 1 N EW NE E 2 2 N NEW N EW 2 3 W NW N W 2 E 0 2 2 0 N 0 1 1 0	

A5

尋找最經濟的產品升級方式

(Time Limit: 2 seconds)

Problem Description

- **English**

k-dominate skyline query is a useful query that can be used to find out "*competitive*" product in the database. A product *p* is said to *k-dominate* another product *q* if there are *k* (≤ *d*) dimensions in which *p* is **better than or equal to** *q* and is **better** in at least one of these *k* dimensions. A product that is not *k*-dominated by any other products is said to be a *k*-dominant skyline product.

For example, consider a person looking for a suitable cell phone. He/she may care about the following three features: weight, standby time, camera pixel. Phone *C* in Table 1 is 2-dominated by phone *B* because *C*'s first and second attribute values are worse than B's first and second attribute values. Therefore, phone *C* is not a 2-dominant skyline product. Similarly, Phone *A* is not a 2-dominant skyline product as there are at least 2 dimensions in which Phone *A* is worse than Phone *B*. Note that in Table 1 Phone *B* and Phone *D* are 2-dominant skyline products because they are not 2-dominanted by any other phones in Table.

Table 1 A set of products.

	1st dimension	2nd dimension	3rd dimension
Phone	Weight (users prefer **smaller** values in this attribute)	Standby time (users prefer **larger** values in this attribute)	Pixel (users prefer **larger** values in this attribute)
A	150	120 hours	2.0
B	140	200 hours	2.0
C	180	120 hours	3.0
D	190	200 hours	3.0

For a manufacturer, it is of interest to provide ***competitive products*** (i.e., *k*-dominant skyline products) instead of uncompetitive ones. Thus, given you an uncompetitive product (i.e., the product that is *k*-dominated by other products in the market), please write a program to identify a cost-minimal way to upgrade the uncompetitive product so that the product will become a *k*-dominant skyline product.

In the following, we use p= {p[1], p[2],..., p[d]} to denote a product, where $p[i]$ is the i-th attribute value of p. The attribute value of each dimension is normalized to [1, 1000]. We assume *smaller* values on each dimension are preferable. For example, p_1={6, 3, 7} and p_2={4, 3, 4} are two 3-dimensional products (i.e., d = 3). p_2 can 2-dominate p_1 as $p_2[1]<p_1[1]$ and $p_2[3] < p_1[3]$.

For simplicity, we assume that the upgrading cost of making a product p become another product p' is the L_1 distance between p and p'. In the aforementioned example, the cost of upgrading p_1 to p_2 is:

$$cost(p_1, p_2) = \sum_{i=1}^{3} |p_1[i] - p_2[i]| = |6-4| + |3-3| + |7-4| = 5$$

We also assume a valid upgrading product p' should obey the following constraint:

$$p'[i] \leq p[i] \; \forall i$$

Thus, p_3 = {4,5,4} is not a valid upgrading product for p_1={6, 3, 7} because $p_3[2] > p_1[2]$.

Given you a product p= {p[1], p[2],..., p[d]}, a k value ($k \leq d$), and a set to existing product D in the market. Please write a program to identify a cost-minimal way to upgrade p so that p will become a k-dominant skyline product in the market (i.e., p cannot be k-dominated by other product in D). Note that the consumers do not like "**copycat**". You cannot upgrade your product q so that q is "identical" to another product in D. For example, there are two existing products p_1 and p_2 in the market (i.e., D = {p_1, p_2}), where p_1={5,7,8} and p_2={4,8,9}. Assume k = 2 and your uncompetitive product is q = {6,7,8}. You cannot upgrade q to q' = {5,7,8} as q' is identical to an existing product p_1.

- **Chinese**

在資料庫的領域中，*k-dominate skyline query* 常被使用來找尋資料庫中具“競爭力”的產品。給定一個產品 *p*，若我們說 *p* 可以 k-dominate 另一個產品 *q*，那表示 *p* 至少有 *k* 個維度（$k \le d$)比 *q* 來得好，或一樣好，但 *p* 在這 *k* 個維度中，至少有一個維度，*p* 比 *q* 還要好。若一個產品它都沒被其他的產品 *k*-dominate 的話，那我們說這個產品是一個 *k*-dominant skyline 產品。

舉例而言，有一位使用者想要買一隻手機，他/她以底下的三個特性做為篩選的條件：重量，待機時間，以及像素。在 Table 2Table 1 中，Phone *C* 被 Phone *B* 2-dominate 了，因為 *C* 的第 1 及第 2 個維度的屬性值比 *B* 的第 1 及第 2 個維度的屬性值來得差。因為 Phone *C* 不是 2-dominant skyline 產品。Phone *A* 也不是一個 2-dominant skyline 產品，因為它至少有 2 個維度的屬性值，比 Phone *B* 來得差。注意在 Table 2 中，因為 Phone *B* 及 Phone *D* 無法被別的產品給 2-dominante，所以它們是 2-dominant skyline 產品。

Table 2 市場中現有的產品。

	1st dimension	2nd dimension	3rd dimension
Phone	重量 (對於這個屬性，使用者偏好較小的值)	待機時間 (對於這個屬性，使用者偏好較大的值)	像素 (對於這個屬性，使用者偏好較大的值)
A	150	120 hours	2.0
B	140	200 hours	2.0
C	180	120 hours	3.0
D	190	200 hours	3.0

對一個製造商而言，他一定想要提供有競爭力的產品（即 *k*-dominant skyline 產品)，而非沒有競爭力的產品。所以，給你一個 uncompetitive product（也就是會被市場上其他商品給 *k*-dominate 的產品)，請找出一個方法，讓這個產品可以用「最經濟」(也就是最便宜)的方式，升級為 *k*-dominant skyline 產品。

在底下，我們用 $p = \{p[1], p[2], ..., p[d]\}$來表示一個產品 *p*，其中 $p[i]$為 *p* 的第 *i* 個屬性值。每個屬性值都正規化到[1,1000]之間。我們假設使用者偏好屬性值較小的資料。舉例而言，$p_1=\{6, 3, 7\}$ and $p_2=\{4, 3, 4\}$是兩個 3 維度的產品（亦即 $d=3$)。因為 $p_2[1]<p_1[1]$ and $p_2[3] < p_1[3]$（較小的屬性值，較佳)，所以 p_2 可以 2-dominate p_1。

為了簡化問題，我們假設升級的成本為兩個產品間的 L_1 distance。在前例中，將 p_1 升級為 p_2 成本如下：

$$cost(p_1, p_2) = \sum_{i=1}^{3} |p_1[i] - p_2[i]| = |6-4| + |3-3| + |7-4| = 5$$

我們也假設一個合法的升級產品 p'，應該符合底下的規範：

$$p'[i] \leq p[i] \; \forall i$$

所以，對於 p_1={6, 3, 7}而言，p_3 = {4,5,4}並不是一個合法的升級產品，因為 $p_3[2] > p_1[2]$。

給你一個產品 p= {p[1], p[2],..., p[d]}，一個 k 值，還有一群市場上已經存在的產品 D。請你寫一個程式，找出一個最經濟的方式，其可以將 p 升級為 k-dominant 產品（即 p 不會被 D 中的產品給 k-dominate）。請注意，由於使用者不喜歡「**模仿者**」，所以你不可以將產品升級為現有市場中的任何一個產品。舉例而言 D={p_1, p_2}為市場上現有的產品，其中 p_1={5,7,8}，且 p_2={4,8,9}。令你的產品為 q={6,7,8}，且 k=2。你不可以將 q 升級為 q'={5,7,8}，因為 q' = {5,7,8} = p_1。

Technical Specification

- **English**
  - $d = 5$
  - $1 \leq p[i] \leq 1000 \; for \; i = 1 \; to \; d.$
  - $p[i] \in Z \; \forall i$
  - $1 \leq k \leq d - 2$
  - $1 \leq |D| \leq 40000$
  - The number of test cases is between 1 and 10.
- **Chinese**
  - 資料維度為 5
  - 產品的每個屬性值（即 $p[i]$）介於 1 到 1000 之間
  - 每個產品的屬性值都是整數
  - k 的值介於 1 到 d-2 間
  - 現有市場的產品個數（即 D 中所包含的產品數）介於 1 到 40000 個
  - 測資的個數介於 1 到 10 之間

Input Format

English

The input consists of two parts.

The first part is a database containing a set of |*D*| existing products in the market.

The second part contains information of several test cases.

The first line of the first part is a single line containing an integer |*D*| indicating the number of existing products in the market. Each of the next |*D*| lines indicates an existing product. Each product is a single line containing *d* integers delimited by a space. The *i*-th integer indicates the *i*$-th attribute value of the product.

The first line of the second part is a single line containing an integer *t* indicating the number of test cases in the problem.

The first line of each test case is an integer indicating the *k* value.

The second line of the test case is an uncompetitive product. The product is a single line containing *d* integers delimited by a space.

Chinese

測資包含了兩個部分，第一部分是一個資料庫，其中包含了|*D*|筆現有市場的產品。第二部分則包含了數個 test case 的資訊。

第一部分的第一行是一個整數|*D*|，其代表了現有市場上產品的個數。其後的|*D*|行每一行都是一個現有市場的產品。每個產品包含了 *d* 個整數，每個整數用空白隔開。第 *i* 個整數代表了這個產品的第 *i* 個屬性值。

第 2 部分的第 1 行是一個整數 *t*，其代表了共有幾個 test case。

每個 test case 的第一行是一個整數，其代表了 *k* 值。

Test case 的第二行代表了一個沒有競爭力的產品。產品包含了 *d* 個整數，每個整數用空白隔開。

Output Format

- **English**

For each test case, output the minimum cost that can upgrade the uncompetitive product to a *k*-dominant skyline product.

- **Chinese**

針對每一筆測試資料，請輸出一個最少的成本，廠商可以利用這個成本將產品升級為 *k*-dominant skyline 商品。

Example

Sample Input:	**Sample Output:**
2 5 20 20 20 20 20 5 20 20 20 1 1 20 20 20 20 20	30

A6

Spy Pass

特務密碼

(Time Limit: 3 seconds)

Problem Description

- **English**

An enemy spy Linya encodes a secret file by a password. For the security reason, Linya produces several fake passwords by adding some characters into some positions. For example, if Linya's real password is "314159265359", Linya can produce several fake passwords A, B and C as A=312415859265359、B=3173415926581359 and C=763141578926536359. Then, Linya transfers the encoded file and the fake passwords to his organization via different methods. Linya and his organization agree that the real password is the longest common subsequence of all the fake passwords. Now, we intercept Linya's encoded file and several fake passwords. Can you help us to recover Linya's password? Because we may not intercept "ALL" the fake passwords, the longest common subsequences may be not unique.

- **Chinese**

敵國秘密特務組織特聘間諜林雅將重要情報使用密碼鎖在一個檔案之中，密碼與檔案透過眾多不同的途徑傳遞到該國特務中心，林雅將密碼寫下好幾份混淆密碼(例如 A、B、C 三份)，但在每一份混淆密碼中隨意不定數的位置隨意地增加不定數字元。假設林雅的密碼為 314159265359，林雅製作 A、B、C 三份可以如下 312415859265359、3173415926581359、763141578926536359，林雅與組織約定密碼就是所有傳遞副本的最長共同子序列。我國情報人員截獲了數份混淆密碼，但是實在不知道該如何計算回原始密碼，你可以幫忙嗎? 但是由於並非所有混淆密碼都截獲，所以算出來的最長共同子序列可能不唯一。

Input Format

- **English**

The first line is an integer M which indicates the number of test cases. The first line of each test case is a number N indicating the number of intercepted fake passwords. The following N lines are N strings representing the N intercepted fake passwords.

- **Chinese**

第一行是一個整數 M 代表測試資料有 M 筆。每一筆測試資料的第一行為一個數字 N 表示截獲 N 份混淆密碼，接連的 N 行為 N 份混淆密碼。

Output Format

- **English**

For each test case, output the number of longest common subsequences, and output all the longest common subsequences ordered by lexicographic ordering.

- **Chinese**

針對每一筆測試資料，第一行輸出其最長共同子序列的個數 K。然後依照字典排序法將 K 個最長共同子序列印出。

Technical Specification

- **English**
  - The characters of Linya's password are digits and English alphabets.
  - The length of the fake password is at most 32.
  - 2<=N<=4.
  - 1<=M<=10.
- **Chinese**
  - 已知林雅密碼及混淆密碼所使用的字元為數字及英文字母。
  - 每一個混淆密碼最長 32 字元。
  - 2<=N<=4.
  - 1<=M<=10.

Example

Sample Input:	Sample Output:
2 2 abcdfgh abccfdsg 3 3124158592654359 3173415926581359 763141578926536359	2 abcdg abcfg 1 314159265359

A7

Taxi dispatch

(Time Limit: 3 seconds)

Problem Description

- **English**

A taxi company advertises the "**fast calling service**" that their taxi will arrive at the position of the customer in 5 minutes after calling. Suppose there are *m* free taxis and *n* customers in different positions at some time, the company needs a good dispatching algorithm for serving their "fast calling service" to maximum number of customers. In addition, we assume that the average speeds of the taxis are the same and one taxi serves for one customer.

- **Chinese**

某計程車公司標榜他們公司的「快速叫車服務」，也就是說計程車在叫車後五分鐘內（含五分鐘整）即可抵達乘客所在的位置。假設在某一個時間點，有 *m* 台空計程車和 *n* 位乘客分散在不同位置，計程車公司需要一個好的派車演算法，使得有最多的乘客享受到「快速叫車服務」。另外，我們假設計程車的平均速度相同而且一台計程車只能服務一位乘客。

Technical Specification

- **English**
  - The numbers of free taxis and customers are *m* and *n* respectively, where $1 \le m, n \le 100$.
  - The positions of the taxis and customers are denoted by integer coordinates (x,y) , where $1 \le x, y \le 100$. The distance between two points is calculated by Hamilton distance. That is, the distance between (x_1,y_1) and (x_2,y_2) is $|x_1-x_2|+|y_1-y_2|$.
  - The average speed of the taxis is 1 unit distance per minute on the coordinate plane.
- **Chinese**

- 空計程車和乘客的數量分別是 m 和 n，其中 $1 \le m, n \le 100$。
- 計程車和乘客的位置以整數座標(x,y)表示，其中 $1 \le x, y \le 100$。兩點間的距離以曼哈頓距離計算，也就是說(x_1,y_1)和(x_2,y_2)的距離為$|x_1-x_2|+|y_1-y_2|$。
- 計程車的平均速度為每分鐘移動座標平面上 1 單位距離。

Input Format

English

The first line is an integer which indicates the number of test cases. The first line of each test case contains two integers m and n ($1 \le m, n \le 100$) indicating the numbers of free taxis and customers respectively and separated by a space. The next m lines indicate the coordinates of taxis, in which each line contains two integers separated by a space. The following n lines indicate the coordinates of customers, in which each line contains two integers separated by a space.

Chinese

第一行是一個整數代表測試資料有幾筆。每筆測資的第一行包含兩個以空格隔開的整數 m 和 n ($1 \le m, n \le 100$)，分別代表空計程車和乘客的數量。接下來的 m 行代表計程車的座標，每行包含兩個以空格隔開的整數；再接下來的 n 行代表乘客的座標，每行包含兩個以空格隔開的整數。

Output Format

English

For each test case, output the maximum number of customers served by "fast calling service" in a line.

Chinese

針對每一筆測試資料，在一行中輸出享受到「快速叫車服務」的最多乘客數量。

Example

Sample Input:	Sample Output:

2 2 2 3 4 1 2 11 71 15 18 5 3 1 2 3 4 5 6 7 8 9 10 9 15 14 10 12 12	0 1

A8

Electric Vehicle

(Time Limit: 2 seconds)

Problem Description

- **English**

The Intensive Thunder Spark Automobiles (ITSA) is planning to test its newest electric vehicle recently. This new model can perfectly convert the potential energy into electricity, and its battery is capable to store infinite electricity. However, the cost of charging such battery is very high. In order to minimize the cost of the test drive, ITSA has to find the least amount of energy sufficient to finish the test drive.

The vehicle is allowed to enter n cities numbered from 1 to n. The test drive starts at city s, and its destination is city t. The altitude of city i is h_i meters for $1 \leq i \leq n$. There are m bidirectional roads $r_1, r_2, \ldots, r_m$, and r_j ends at city u_j and city v_j where $u_j, v_j \in \{1, \ldots, n\}$ for $1 \leq j \leq m$. When the vehicle passes road r_j, the energy stored in its battery will change by the following two factors.

1. Direction: Let Δh_j is the difference between the altitudes of both ends of r_j. Since the vehicle can convert potential energy into electricity perfectly, the stored energy will increase by Δh_j units if it moves from the higher end to the lower end. Otherwise, the stored energy will decrease by Δh_j units.
2. Friction: The friction will drain w_j units of energy from the battery.

If the vehicle does not have enough energy, then it cannot pass the road. For example, road r ends at city u and city v where $h_u = 10$ and $h_v = 20$, and the friction will drain $w = 5$ units of energy when the vehicle passes road r. The vehicle cannot pass road r from city u to city v if its battery stores less than 15 units of energy, since passing r in this direction needs (20-10)+5=15 units. On the other hand, the vehicle can always pass road r from city v to city u even if its battery is empty, and the vehicle will gain (20-10)-5=5 units of energy in this direction.

Please write a program to compute the least amount of energy stored in the battery which is sufficient to finish the test drive from city *s* to city *t*.

Chinese

密集閃電火花汽車 (ITSA) 計畫於近日測試最新型的電動車。這部新車能夠將位能完全無耗損的轉換成電能，而且所使用的電池能夠無限制的儲存電能。但這電池充電所需的耗費很大，為了節省測試駕駛的花費，ITSA 必須要算出完成測試駕駛所需的最小能量。

這部測試車被允許駛入 n 個編號為 $1, ..., n$ 的城市，測試駕駛自 s 開始，終點為 t。對 $1 \le i \le n$，城市 i 的海拔分別為為 h_i 公尺。城市之間總共有 m 條雙向道路，其中第 j 條道路的兩端點分別為城市 u_j 跟城市 v_j (此處 u_j, v_j 為城市編號，即 $u_j, v_j \in \{1, \ldots, n\}$)。當測試車通過第 j 條道路時，電池所儲存的電量，將依據下列兩個因素改變。

1. 方向：令 Δh_j 為第 j 條道路兩端點的高度差。如果測試車從高處往低處走，電池的電量會增加 Δh_j 單位的能量，反之將會減少 Δh_j 單位的能量。
2. 摩擦力：摩擦力將會耗損 w_j 單位的能量。

如果測試車沒有足夠的能量，那麼他就不能夠通過該道路。舉例來說，某道路的兩端點是 u 跟 v，海拔分別是 h_u=10 及 h_v=20，通過該道路摩擦力將耗損 w=5 單位的能量。如果測試車的能量不足 15 單位，則無法從 u 往 v 的方向，因為此方向需要(20-10)+5=15 單位的能量。然而，從 v 往 u 的方向，這測試車總是可以通過，還因通過這條路獲得(20-10)-5=5 單位的能量。即便測試車沒電，仍舊可以通過此道路，從 v 前往 u。請撰寫一個程式，計算出從 s 前往 t，測試車所需的最小初始能量是多少。

Technical Specification

- **English**
  - There are at most 20 test cases, and all numbers are integral.
  - The number of cities is no more than 5000. I.e., $n \le 5000$.
  - The altitude of any city is a number between -10^5 and 10^5. I.e., $-10^5 \le h_i \le 10^5$ for $1 \le i \le n$.
  - The number of roads is no more than 20000. I.e., $m \le 20000$.
  - The friction will drain at most 10^5 units of energy when the vehicle passes a road. I.e., $0 \le w_j \le 10^5$ for $1 \le j \le m$.

- **Chinese**
  - 至多有 20 筆測試資料，所有數字均為整數。
  - 至多有 3000 個城市，即 $n \leq 5000$。
  - 城市海拔是介於-10^5與10^5之間的數字，即對 $1 \leq i \leq n$，$-10^5 \leq h_i \leq 10^5$。
  - 至多有 20000 條道路，即 $m \leq 20000$。
  - 通過任一條道路，摩擦力所耗損的能量至多 10^5 單位，即對 $1 \leq j \leq m$，$0 \leq w_j \leq 10^5$。

Input Format

- **English**

The input consists of a number of test cases. The first line of the input file contains an integer indicating the number of test cases. The first line of the test case contains four integers *n*, *m*, *s*, *t* separated by blanks. *n* is the number of cities, and *m* is the number of roads. The test drive begins at city *s*, and it ends at city *t*. The second line contains *n* integers h_1, h_2, ..., h_n separated by blanks where h_i is the altitude of city *i*. The following *m* lines describe the information of roads. The *j*-th of these *m* lines contain 3 integers u_j, v_j and w_j separated by blanks. The two ends of road r_j are city u_j and city v_j, and the friction will consume w_j units of energy when the vehicle passes road r_j.

- **Chinese**

第一行是一個整數代表測試資料有幾筆。每一筆測試資料的第一行有四個數字 *n*、*m*、*s*、*t*，由空格隔開。*n* 代表有多少城市，*m* 代表有多少道路，*s* 代表測試駕駛起點，*t* 代表測試駕駛終點。測試資料的第二行有 *n* 的以空格隔開的數字 h_1、h_2、...、h_n，其中 h_i 代表第 *i* 個城市的海拔高度。接下來 *m* 行代表所有道路的資訊，其中的第 *j* 行有三個以空白隔開的數字 u_j、v_j、w_j，其中城市 u_j 與城市 v_j 為第 *j* 條道路的兩個端點，w_j 代表通過該道路，摩擦力所耗損的能量。

Output Format

- **English**

The output contains one line for each test case. Each line contains one integer indicating the least amount of the energy initially stored in the battery which is sufficient to finish the test drive.

- **Chinese**

針對每一筆測試資料輸出一行，該行要有一個數字，代表完成測試駕駛所需的最小初始能量。

Example

Sample Input:	Sample Output:
2 4 3 1 4 10 2 3 1 1 2 4 2 3 1 3 4 3 4 4 1 4 0 3 -3 0 1 2 4 2 4 4 1 3 5 3 4 5	0 8

A9

Lottery

(Time Limit: 2 seconds)

Problem Description

● English

To increase the public interest in the nationwide lottery, a new lottery game is being considered. The new lottery game involves picking a lottery ticket that consists of 4 alphanumeric characters (i.e. 'A', 'B', …, 'Z', and '0', '1', …, '9') and a lucky number w_i. Each week, at most 50,000 lottery tickets will be sold and the computer system ensures that no two tickets are exactly the same. In addition to awarding possibly one winning lottery ticket, persons picking the correct lucky number W of the week will also share a special pot prize. The winning lucky number for each week is the number of pair of lottery tickets sold with lottery alphanumeric characters differ in P number of positions, where P=1, 2, 3, or 4. In the example below, if N=6 lottery tickets are sold in a week: {ABCD, BC0D, DCAB, A5CF, ABFF, ABCB}, then the winning lucky number for the week with P=2 is 5 (ABCD/A5CF, ABCD/ABFF, A5CF/ABFF, A5CF/ABCB, ABFF/ABCB).

Please write a program to determine the winning lucky number for a given week of lottery tickets sold.

● Chinese

樂透彩公司為了刺激買氣，推出了一個新樂透彩券玩法。買樂透彩券時購買者可自行選擇 4 個數字或英文大寫字母的組合，作為每週對獎用。但彩券公司每週最多只賣出 50,000 張樂透彩券，且電腦會確保任意兩張彩券上的數字字母組合不會相同。購買彩券者可同時另選擇一個整數 w_i，若 w_i 與該週幸運數字 W 相同，則可共享樂透彩公司特別準備的獎金。每週的幸運數字產生方式如下：

樂透彩公司會事前公布一整數 P，P 為 1、2、3 或 4，用以產生該週幸運數字 W：即賣出的彩券中，差異恰為 P 位的彩券總對數。

例如若某週賣出 N=6 張彩券，分別為 {ABCD, BC0D, DCAB, A5CF, ABFF, ABCB}，且若 P=2，則該週幸運數字為 5 (ABCD/A5CF，ABCD/ABFF，A5CF/ABFF，A5CF/ABCB，ABFF/ABCB)。

請寫一個程式計算該週的幸運數字。

Input Format

- **English**

The first line an integer indicating the number of test cases to follow. For each test case, the first line contains two numbers N, P separated by space. N (2<=N<=50,000) is the number of lottery tickets sold and P is the chosen number (P=1, 2, 3 or 4) used to determine the lucky number. The next N lines each contain a 4-alphanumeric character lottery ticket.

- **Chinese**

第一行有一正整數，代表測試資料組數。每一組測試資料的第一行有兩個以空白分開的數字 N 與 P。N (2<=N<=50,000) 代表賣出的彩券數，P 為 1, 2, 3, 或 4，為樂透彩公司用以計算該週幸運數字用。接下來的 N 行每行各有一張彩券上的四個數字或大寫英文字母。

Output Format

- **English**

Output a single integer, which is the lucky winning number.

- **Chinese**

寫出一整數代表該週之幸運數字。

Example

Sample Input	Sample Output
1 6 2 ABCD BC0D DCAB A5CF	5

ABFF ABCB	

A10

Satan's Test

(Time Limit: 1 second)

Problem Description

- **English**

The pirates of the Caribbean discover a mysterious island in the Atlantic ocean. However, the notorious Satan is sleeping over there and waken up by these pirates. Satan is very angry at them and force each of them to do a devil test. The pirate will be killed if not passed. In the devil test, Satan will say a dark message, letter-by-letter, and the pirates have to respond him the real message, letter-by-letter, immediately. Fortunately, the oldest pirate knows how the dark messages is generated from real one. The rule is described below.

Given a real message, its corresponding dark message is created by first sorting all the circular rotations. For example, given a message: GCA. Three rotated messages, GCA, CAG, and AGC are generated. The rotated messages are then sorted into lexicographical order (i.e., AGC, CAG, GCA). The last letters in the sorted messages (i.e., CGA) will be the dark message spoken by Satan. However, the oldest pirate still doesn't know which rotated message is the real one. Fortunately, he observed that Satan always smiles when speaking of the last letter belonging to the real message. For instance, Satan smiled when speaking of the 3rd letter (A) of the dark message CGA. Note that if there are multiple identical letters in the dark message, Satan only smile at the letter of the real message but not at other identical letters.

Given a string of dark message (e.g., CGA) and the index of letter when Satan smiled (e.g., 3), write a program that helps pirates answer the real message (i.e., GCA).

- **Chinese**

一群加勒比海盜在大西洋中發現一個神祕的島嶼。可是惡名昭彰的撒旦

正在島上睡覺，被這些海盜吵醒。撒旦非常生氣，強迫每一個海盜進行一個惡魔的測驗，沒通過就會殺死他。在測驗之中，撒旦會講出一則黑暗的訊息，海盜必須馬上回答出背後隱藏的真正訊息。還好裡面最老的海盜知道黑暗訊息是如何產生的，其規則如下所述。

給定一則真實的訊息(例如，GCA)，要產生其對應之黑暗訊息，首先須把真實訊息的所有旋轉字串產生並排序。例如 GCA 可產生總共三種旋轉字串 GCA、CAG、及 AGC。將這些旋轉字串根據字典順序排序，其順序將變成 AGC、CAG、及 GCA。將這三個排序後字串的最後一個字，取出並結合，此即為撒旦所說出口之黑暗訊息(CGA)。可是最老的海盜依然不知道在這些旋轉訊息之中，哪一個是真實的。幸好，他發現當撒旦唸到屬於真實訊息的最後一個字母時，會露出一抹微笑。例如，當撒旦唸出黑暗訊息(GCA)的第三個字母 A 時露出了微笑。請注意若黑暗訊息有多個相同字母時，則撒旦只會在唸到屬於真實訊息的結尾字母時才微笑，其他相同字母並不會笑。

給定任一則黑暗訊息(例如，CGA)，以及撒旦露出微笑的字母編號(例如，3)，請寫出程式幫助海盜快速回答出真實訊息(例如，GCA)。

Technical Specification

- **English**
  - Each message letter is from standard alphabet {A, B, C, …, Z}.
  - The length of dark message ranges from 3 to 1500 letters.
  - The index of letter when Satan smiles ranges from 1 to the length of dark message.
- **Chinese**
  - 每則黑暗訊息的字母是標準英文字母集{A, B, C, …, Z}
  - 黑暗訊息長度介於 3 到 1500 個字母之間
  - 撒旦露出微笑的字母編號介於 1 至黑暗訊息長度之間

Input Format

- **English**

  The first line contains the number of test cases. Each test case consists of two consecutive lines. The first line stores the dark message spoken by Satan. The second line consists of one number standing for the index of letter when Satan smiled.

- **Chinese**

  第一行包含一個數字代表測資總數量。每一筆測資為連續二行。每一筆測資第一行是撒旦所念出之黑暗訊息。第二行為一個數字，代表撒旦露出微笑的字母編號。

Output Format

- **English**

  For each test case, output the real message in one line.

- **Chinese**

  針對每一筆測試資料，於每一行輸出真實訊息。

Example

Sample Input:	Sample Output:
2 CGA 3 NNBAAA 4	GCA BANANA

A11

Smart Patrol

(Time Limit: 6 seconds)

Problem Description

- **English**

Happy City is of rectangular shape. It is a lovely and crime-free place. Nevertheless, for safety concern, there are still a couple of police stations, each of which has one or more policemen that take responsibility to patrol the streets nearby, as illustrated in Figure 1.

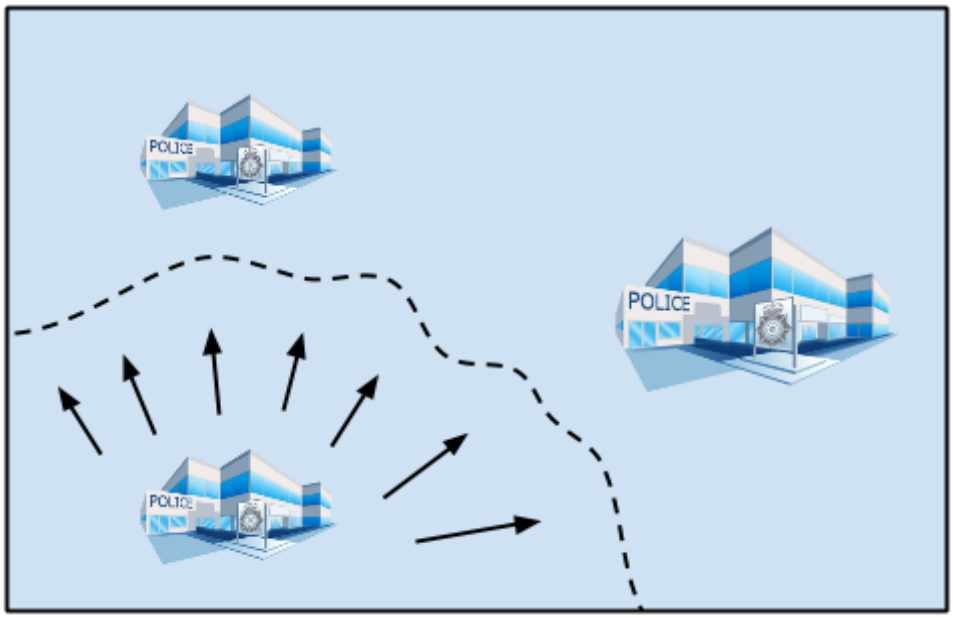


Figure 1. Each police station has a patrol area.

Joseph, as the new mayor of Happy City, would like to reduce the burden of those policemen while ensuring the whole city is well guarded. The idea is to reorganize the patrol area of each police station such that (1) the patrol area of each police station is minimized, (2) the union of all patrol areas covers the whole city, and (3) the size of patrol area of a police station is proportional to the number of policemen in that station.

The idea is simple, but to solve the problem is difficult. Thus, Joseph has come up with a simplified (and mathematical) way to define the patrol area of a police station. For a station *s* having *k* policemen, its patrol area is defined by a square that has width $round(k^{1/2} \times c)$ meters and is centered at *s*, where *c* is a constant and $round(k^{1/2} \times c)$ is the value of $k^{1/2} \times c$ rounded to the nearest integer. For example, if the position of *s* is (8, 4), $k = 10$, and $c = 3$, the patrol area of *s* is a 9×9 square centered at (8, 4). In this way, we can easily visualize the patrol area of each police station, as shown in Figure 2. Note that the patrol area of a police station may partially go outside of the city or overlap with that of another police station, which is acceptable.

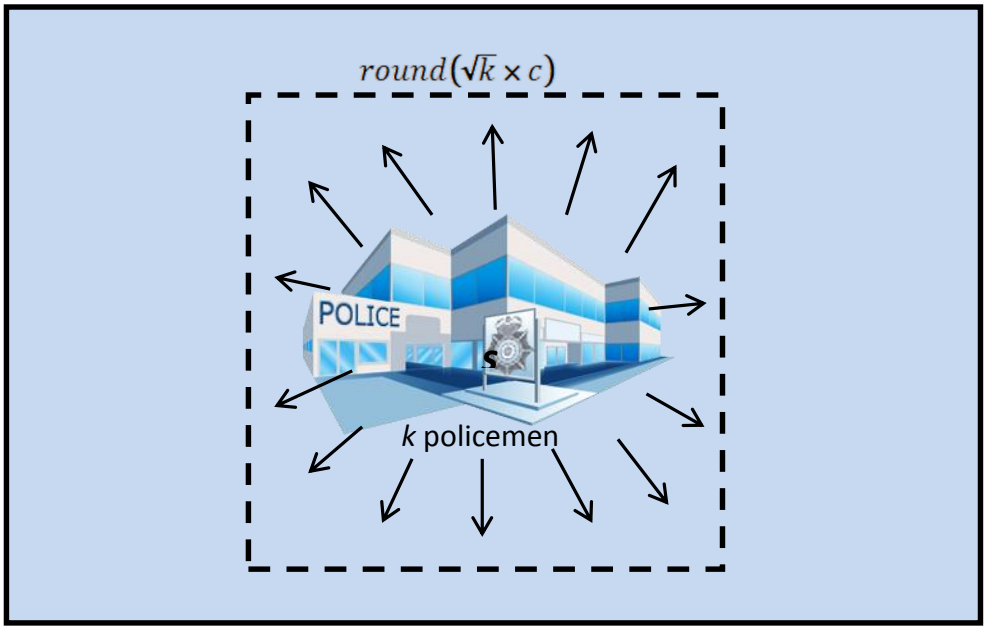


Figure 2. The square patrol area defined by Joseph.

Joseph is wondering how large the constant *c* is? If *c* is too small, there will be some regions not patrolled. See Figure 3 for an example. On the contrary, if *c* is too large, the patrol area will be heavily overlapping and will increase the burden of policemen. See Figure 4 for an example.

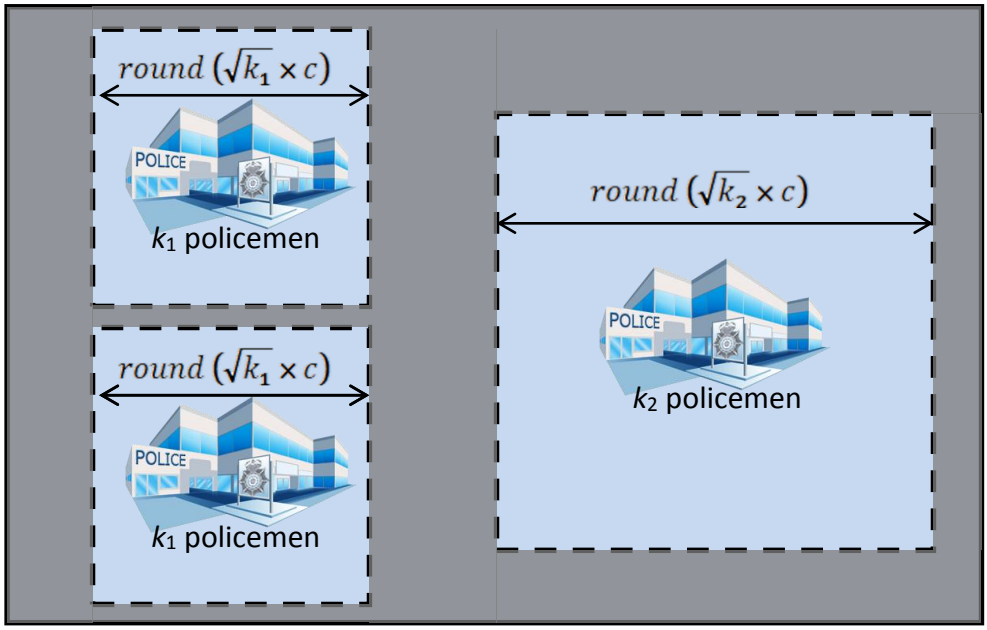


Figure 3. A small constant c will leave some regions not being patrolled.

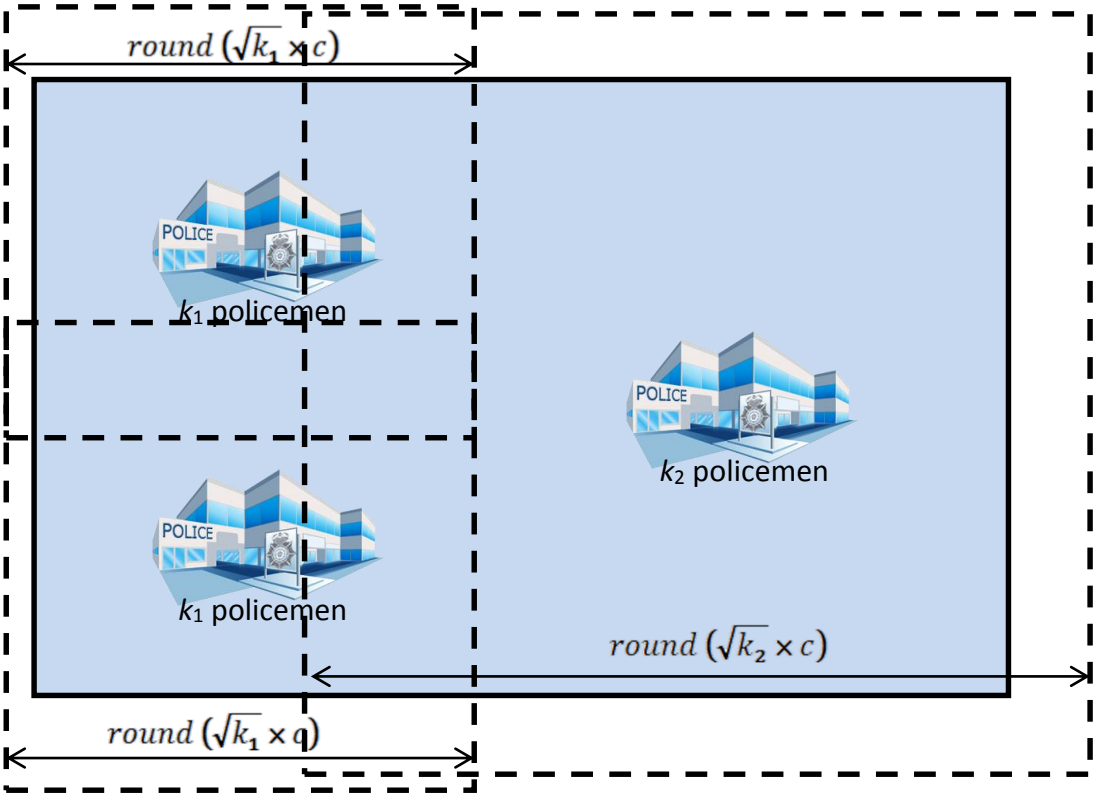
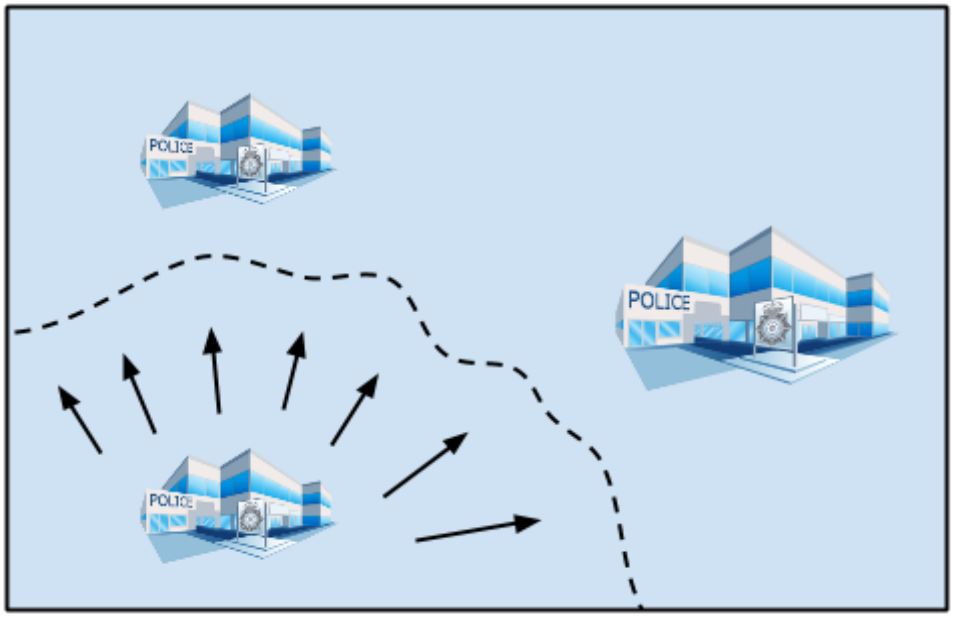


Figure 4. A large constant c will increase the burden of policemen.

Now, Joseph asks for your help. Please help Joseph determine the smallest *c* such that every place in the city is within the patrol area of at least one police station. Since *c* could be an ugly floating-point number, you only need to derive the smallest integer *c* such that the resulting patrol areas cover the whole city.

- **Chinese**

"快樂城市"是一個長方形的區域，也是一個沒有犯罪發生的地方。即便如此，城市中仍有警察局負責巡邏轄區附近的區域，如圖一所示。

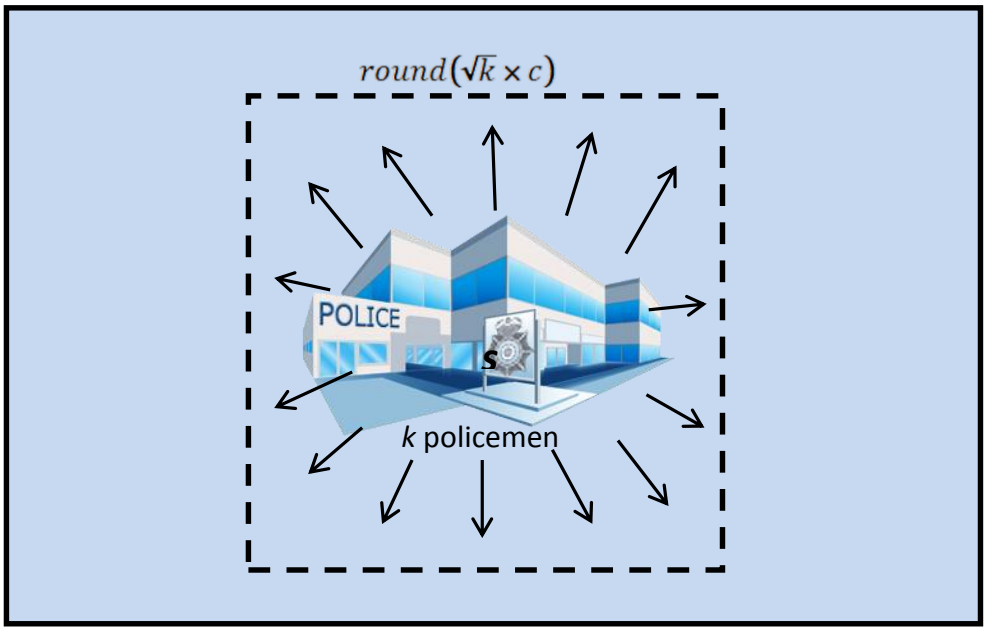


圖一. 每間警察局有負責巡邏的區域

身為快樂城市的新市長，約瑟夫想要重新規劃每間警察局負責的巡邏區域，希望藉此能減輕警察的工作量，但又同時確保整個城市都在巡邏範圍內。因此約瑟夫對於重新規劃巡邏區域有了以下的構想：(1) 每間警察局的巡邏區域越小越好、(2) 所有的巡邏區域必須覆蓋整個城市、(3) 每間警察局的巡邏區域須與該警察局的警察人數成正比。

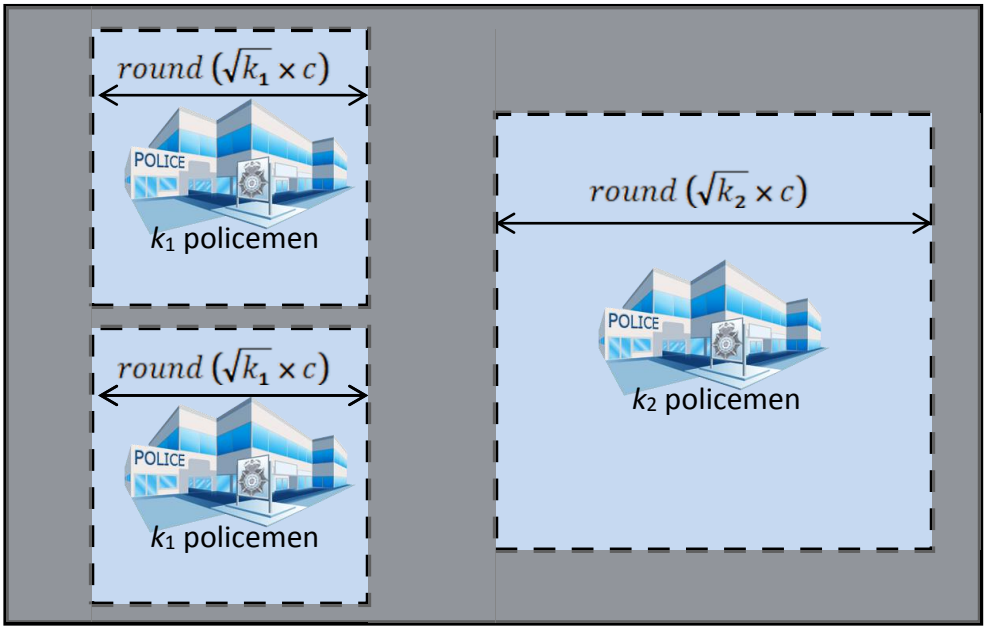
構想雖然說起來簡單，要按此構想重新規劃巡邏區域卻不是件容易的事。因此約瑟夫想到一個簡化的方法來劃定巡邏區域：對一個有 k 位警察的警察局 s 而言，該警局所負責的巡邏範圍是一個以 s 為中心、邊長為 $round(k^{1/2} \times c)$ 公尺的正方形，其中的 c 是一個常數，$round(k^{1/2} \times c)$ 則是 $k^{1/2} \times c$ 四捨五入後的整數值。例如: 假設 s 的位置為 $(8, 4)$, $k = 10$, 且 $c = 3$, 則 s 的巡邏範圍是一個以 $(8, 4)$ 為中心的 9×9 正方形。如圖二所示，用此方式可輕易規範每個警察局的巡邏範圍。當然以此方式定義的巡邏範圍有

可能會部分落在城市之外，不同警局的巡邏範圍也有可能會重疊，這都是合理的情況。

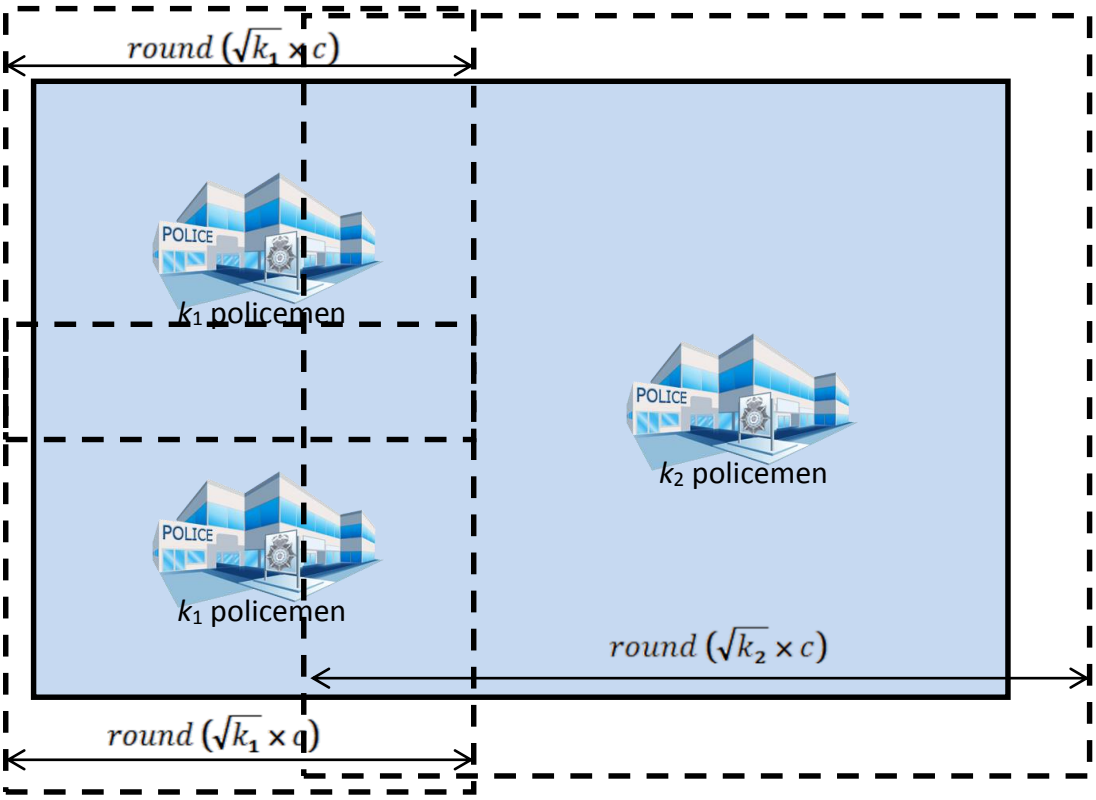


圖二. 約瑟夫新制定的正方形巡邏範圍

約瑟夫很納悶到底這個常數 c 的值是多少？如果 c 太小，城市中會有部分區域沒有被巡邏範圍覆蓋到。相反地，如果 c 太大，每個警察局的巡邏範圍會有過度重疊的情形，造成警察不必要的負擔。圖三與圖四分別給了 c 太小與太大的例子。



圖三. 常數 c 太小會造成部分區域沒有被巡邏到



圖四. 常數 c 太大會增加警察的負擔

現在約瑟夫尋求你的幫忙，請你幫約瑟夫找到一個最小的常數 c，讓城市中的每個地區都在警察巡邏範圍內。由於最佳的 c 可能是一個很醜的浮點數，你只要幫忙找到一個最小的整數 c 讓巡邏範圍可以覆蓋著整個城市即可。

Technical Specification

- **English**
  - Happy City is a rectangular region where the bottom-left corner is at (0, 0) and the top-right corner is at the integer coordinate (w, h), $1 \le w, h \le 10^7$.

- Each police station is located at integer coordinate (x, y), $0 \le x \le w$ and $0 \le y \le h$.
- The number of police stations is between 1 and 20000.
- The number of policemen in each police station is between 1 and 100.

- **Chinese**

- 快樂城市是個長方形的區域，左下角坐落於座標 $(0, 0)$，右上角坐落於座標 (w, h), $1 \le w, h \le 10^7$。
- 每間警察局坐落於座標 (x, y), $0 \le x \le w$ 且 $0 \le y \le h$。
- 警察局的數目介於 1 到 20000 之間。
- 每間警察局的警察人數介於 1 到 100 之間。

Input Format

- **English**

The first line is an integer t, $1 \le t \le 15$, indicating the number of test cases. Each test case begins with two integers w and h ($1 \le w, h \le 10^7$), which specify the width and height of Happy City, respectively. The next line contains a single integer n, ($1 \le n \le 20000$), which denotes the number of police stations. Then, n lines follow and each line consists of three integers k ($1 \le k \le 100$), x ($0 \le x \le w$), and y ($0 \le y \le h$), indicating that there is a police station with k policemen locating at (x, y).

- **Chinese**

第一行是一個整數 t ($1 \le t \le 15$)，代表測試資料組數。每一筆測試資料的第一行是兩個整數 w 及 h ($1 \le w, h \le 10^7$)，分別代表快樂城市的寬與高。下一行包含了一個整數 n ($1 \le n \le 30000$)，代表警察局的數目。接下來的 n 行，每一行包含了三個整數 k ($1 \le k \le 100$), x ($0 \le x \le w$), y ($0 \le y \le h$)，其中 k 代表的是該警察局的警察人數，數對 (x, y) 則代表警察局的座標。

Output Format

- **English**

For each test case, print a line containing the test case number (beginning with 1) followed by an integer, which is the smallest integer *c* that Joseph wanted to know. Use the format of the sample output.

- **Chinese**

針對每一筆測試資料，在一行內輸出測試資料的編號 (從 1 開始)，接著輸出約瑟夫想知道的最小的整數 *c* 的值。請參考 Sample Output 的格式。

Example

Sample Input:	Sample Output:
2 12 8 3 4 2 2 16 8 4 4 2 6 12 8 3 4 2 2 10 8 4 4 2 6	Case 1: 2 Case 2: 3

Illustrations of Sample Input