2017/05/06 全國大專 ITSA 盃程式設計桂冠挑戰大賽

A. 挑戰組試題

13 題，共 43 頁

題號	題目	Time Limit (secs)
A1	Palindrome 迴文偵測	3
A2	循環數列 Cyclic numbers	1
A3	Dissimilarities	1
A4	Counting Tilt Squares	1
A5	Only one bicycle 只有一台腳踏車	1
A6	Potions recipe（魔藥配方）	3
A7	The Disjunctive Domination Problem on Trees	3
A8	破解古老文字 Cracking Ancient Text	3
A9	Borobudur Restoration	3
A10	Linear Extensions	2
A11	三個時間點(Three Points of Time)	2
A12	Longest Journey	3
A13	Treasure	2

A1
Palindrome 迴文偵測

(Time Limit: 3 seconds)

Problem Description

- **English**

A string is a palindrome if the string is the same as its reverse string. For example, "abcba" is a palindrome, but "abc" is not a palindrome.

- **Chinese**

給定一個字串，判斷其是否與其反轉字串相等，如果是的話，則稱之為迴文，如果不是的話，則稱之為非迴文。例如：abcba 為迴文，而 abc 為非迴文。

Input Format

- **English**

The first line is an integer *N* which indicates the number of test cases. Each test case occupies a line and contains a string consisting of only English alphabets.

- **Chinese**

第一行是一個整數 *N* 代表測試資料有 *N* 筆。每一筆測試資料為一行，該行含有一個由英文字母所組成的字串。

Output Format

- **English**

For each test case, output "Yes" if the string is a palindrome, and output "No" otherwise.

- **Chinese**

針對每一筆測試資料中，該字串如果是迴文，請輸出 "Yes"，否則請輸出 "No"。

Technical Specification

- **English**
  - Each string is a combination of capital and lower English alphabets.
  - The length of each string is at most 32.
  - $1 \leq N \leq 10$
- **Chinese**
  - 每一個字串英文字都是由大小寫字母所組成。
  - 每一個字串的長度最長為 32。
  - $1 \leq N \leq 10$

Example

Sample Input:	Sample Output:
4 aBcd abcba XYzzYX abCcba	No Yes Yes No

A2
循環數列 Cyclic numbers

(Time Limit: 1 second)

Problem Description

● English

Let P, Q and R be three given positive integers. For an ordered pair (x,y) of integers, we define a function
$F(x,y)= (Q*x + R*(x-y)^2+y) \pmod P$, mapping (x,y) to an integer between 0 and P-1, where (mod P) is the operation of taking the remainder after divided by P. Starting with two given integers as S(1) and S(2), we define an infinite sequence S(i), i=1,2,3,..., where S(i)=F(S(i-2),S(i-1)) for any i>2. For example, let P=3 and Q=2 and R=0. If S(1)=S(2)=1, then we have S(3)=2*1+1 (mod 3)=0, S(4)=2*1+0 (mod 3)=2, S(5)=0*2+2 (mod 3)=2. Similarly, S(6)=0, S(7)=1, S(8)=1. As you can see, in this example, S(7)=S(1) and S(8)=S(2). Since the next number depends only on the previous two numbers, we can conclude, the sequence S=(1,1,0,2,2,0,1,1,0,2,2,0,...) and the subsequence (1,1,0,2,2,0) will repeat forever. We say that the sequence is cyclic and has the cycle length 6. For another example, let P=4, Q=1, R=1, S(1)=2, and S(2)=2. Then, S=(2,2,0,2,2,0,...) where (2,2,0) repeats forever. Thus, the cycle length is 3.

In this problem, given P, Q, R and the starting numbers S(1) and S(2), your task is to find the cycle length. As shown in the above examples, a key point is to find the smallest j and k such that j<k, S(k)=S(j) and S(k+1)=S(j+1). In the first example, j=1, k=7, and the cycle length is 6; in the second example, j=1, k=4, and the cycle length is 3.

● Chinese

假設 P 與 Q 與 R 是給定整數，對於整數對(x,y)，我們定義
$F(x,y)=(Q*x + R*(x-y)^2+y) \pmod P$，其中(mod P)是指取除 P 的餘數。
從兩個給定的整數當作開始值 S(1)與 S(2)，我們定義一個無限序列 S(i), i=1,2,3,..., 其中 S(i)=F(S(i-2),S(i-1)) 對於任意 i>2。例如，P=3 且 Q=2，R=0。假設 S(1)=S(2)=1，那麼，我們可以得到 S(3)=2*1+1 (mod 3)=0, S(4)=2*1+0

(mod 3)=2, S(5)=0*2+2 (mod 3)=2。相似的，S(6)=0, S(7)=1, S(8)=1。你可以看到，在此例中 S(7)=S(1)且 S(8)=S(2)。因為每一個數都只與前兩數有關，我們可以推論序列
S=(1,1,0,2,2,0,1,1,0,2,2,0,...)，其中(1,1,0,2,2,0)會一直重複下去。我們說這個序列是循環的，且循環長度為 6。再看一例，若 P=4, Q=1, R=1，S(1)=2, and S(2)=2。那麼，S=(2,2,0,2,2,0,...)，其中(2,2,0)會一直循環，因此循環長度為 3。

在此問題中，你的任務是對於輸入的 P、Q、R 以及起始值 S(1) and S(2)，求出此序列的循環長度。如同前述例子所顯示，一個關鍵點是找出最小的 j 與 k 滿足 j<k, S(k)=S(j) and S(k+1)=S(j+1)。在第一個例子中 j=1, k=7 而循環長度是 6；在第二個例子中 j=1, k=4 而循環長度是 3。

Technical Specification

- **English**
  - 0<Q<P<500. 0<= R <10.
  - S(1)<P and S(2)<P are nonnegative integers.
- **Chinese**
  - 0<Q<P<500. 0<= R <10
  - S(1)<P and S(2)<P 是非負整數。

Input Format

- **English**

  The first line is an integer indicating the number of test cases. Each test case contains five numbers separated by spaces, *i.e.*, P, Q, R, S(1) and S(2), in one line.

- **Chinese**

  第一行是一個整數代表測試資料有幾筆。每一筆測試資料一行包含五個整數依序分別為 P, Q, R, S(1)與 S(2)，中間以一個空白隔開。

Output Format

- **English**

  For each test case, output the cycle length in one line.

- **Chinese**

針對每一筆測試資料，輸出循環長度，每筆輸出後輸出一個換行。

Example

Sample Input:	Sample Output:
3 3 2 0 1 1 4 1 1 2 2 313 37 2 1 2	6 3 104

A3
Dissimilarities

(Time Limit: 1 seconds)

Problem Description

● English

In biology, an organism is often represented by its gene sequence. Comparing the dissimilarity between two gene sequences is an important task in biological study. In this problem, a gene sequence is represented by a permutation of $\Sigma = \{1, 2, ..., n\}$. Let S be a gene sequence. For each pair of genes i, j, where $i, j \in \Sigma$ and $i \neq j$, define the *distance* between i and j on S, denoted by $d(S, i, j)$, to be the difference in their positions on S. For example, if $S = (3, 4, 2, 1)$, then $d(S, 1, 3) = 3$ and $d(S, 2, 3) = 2$.

Let X and Y be two gene sequences. For each pair of genes i, j, where $i, j \in \Sigma$ and $i \neq j$, define the *distance difference* of i and j on X and Y, denoted by $\delta(X, Y, i, j)$, to be the difference in their distances on X and Y. For example, if $X = (3, 4, 2, 1)$ and $Y = (2, 4, 1, 3)$, then $\delta(X, Y, 2, 3) = |d(X, 2, 3) - d(Y, 2, 3)| = |2 - 3| = 1$. The *dissimilarity* between X and Y, denoted by $\Delta(X, Y)$, is defined to be the total distance difference of all pairs of genes i, j, where $i, j \in \Sigma$ and $i < j$. That is,

$$\Delta(X, Y) = \sum_{i,j \in \Sigma, i<j} \delta(X, Y, i, j) = \sum_{i,j \in \Sigma, i<j} |d(X, i, j) - d(Y, i, j)|.$$

Please write a program to compute the dissimilarity between two given gene sequences X and Y.

● Chinese

在生物學中，我們常用一個基因序列表示一個生物，而比較基因序列的差異是生物學中重要的任務。在本問題中，我們以 $\Sigma = \{1, 2, ..., n\}$ 表示一個由 n 個基因所構成的集合，一個基因序列指的是由 Σ 中的基因所構成的一個排列。令 S 為一個基因序列，對於每一對基因 $i, j \in \Sigma, i \neq j$，我們定義 i, j 在 S 上的「距離」 $d(S, i, j)$ 為它們在序列 S 上位置的差異。例如，假如 $S = (3, 4, 2, 1)$ ，則 $d(S, 1, 3) = 3$，而 $d(S, 2, 3) = 2$。

令 X, Y 為兩個基因序列，對於每一對基因 $i,j \in \Sigma, i \neq j$，定義 i,j 在 X 和 Y 上的「距離差」 $\delta(X, Y, i, j)$ 為 i,j 在 X 與 Y 上的距離之差值。例如：假如 $X = (3, 4, 2, 1)$ 而 $Y = (2, 4, 1, 3)$，則 $\delta(X, Y, 2, 3) = |d(X, 2, 3) - d(Y, 2, 3)| = |2 - 3| = 1$。基因序列 X 與 Y 的「分歧」 $\Delta(X, Y)$ 定義為所有基因對 $i,j \in \Sigma, i < j$ 之距離差的總和。也就是，

$$\Delta(X, Y) = \sum_{i,j \in \Sigma, i<j} \delta(X, Y, i, j) = \sum_{i,j \in \Sigma, i<j} |d(X, i, j) - d(Y, i, j)|。$$

給定兩個基因序列 X 和 Y，請寫一支程式計算 X 與 Y 的分歧。

Technical Specification

- **English**
  - The size n of Σ is an integer between 2 to 20.
  - Each of X and Y is a permutation of $\{1, 2, ..., n\}$.
- **Chinese**
  - 基因集合 Σ 的大小 n 介於 2 到 20 之間。
  - 序列 X 與 Y 各是集合 $\{1, 2, ..., n\}$ 的一個排列。

Input Format

- **English**

  The first line is an integer indicating the number of test cases. Each test case contains three lines. The first line is an integer n, indicating the size of Σ. The second and third lines both contain n integers, which describe, respectively, the sequences X and Y. Consecutive integers in each sequence are separated by at least one space.

- **Chinese**

  第一行是一個整數代表測試資料有幾筆。每一筆測試資料包含三行。第一行有一個數字 n，表示基因集合 Σ 的大小。第二行與第三行皆有 n 個整數，分別描述序列 X 與 Y。連續的整數間各以一個以上的空白隔開。

Output Format

- **English**

For each test case, output the dissimilarity between X and Y in one line.

- **Chinese**

針對每一筆測試資料，輸出序列 X 和 Y 的分歧。每筆輸出佔一行。

Example

Sample Input:	Sample Output:
2 4 1 3 2 4 4 3 2 1 6 1 5 2 3 4 6 6 4 3 1 5 2	4 12

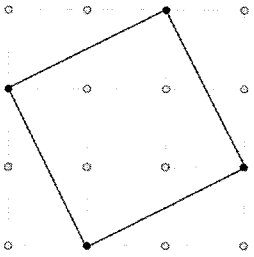
A4
Counting Tilt Squares

(Time Limit: 1 second)

Problem Description

● English

Given a grid with N rows and N columns of dots, we want to count how many different tilt squares can be in the grid. Each side of a tilt square is neither horizontal nor vertical. A tilt square must have the four vertices at the grid points. There can be many different lengths for the squares. Write a program to determine the number of such squares in the given grid. The following figure shows an example of a tilt square within a 4 by 4 grid.



● Chinese

給定一有 N 行及 N 列的格子點陣列，我們想計算其中有幾個傾斜正方型如上圖所示。 傾斜正方型的四個邊長度相等，每個邊既非水平也非垂直，且其頂點須為格子點，邊長可以有幾種不同的長度。請寫一程式在給定的格子方陣上計算各種相異傾斜正方型的個數，只要頂點不一樣的傾斜正方型就被視為相異的。

Input Format

● English

The first line is a positive integer indicating the number of test cases. Each

test case contains a positive integer N (2<= N <= 100) in a line.

- **Chinese**

  第一行是一個整數代表測試資料有幾筆。每一行有一筆測試資料包含一個正整數 N (2<= N <=100)。

Output Format

- **English**

  For each test case, output the number of different tilt square.

- **Chinese**

  針對每一筆測試資料，輸出相異傾斜正方型的個數。

Example

Sample Input:	Sample Output:
3 2 3 4	0 1 6

A5

Only one bicycle 只有一台腳踏車

(Time Limit: 1 second)

Problem Description

- **English**

A group of $n >= 1$ tourists want to travel from A to B and they have one bicycle. A maximum of two people can use that bicycle at one time. The bicycle must be used back and forth. Due to some reasons, different tourists take different time to use that bicycle and at most two people can use that bicycle at the same time. For two people on a bicycle, the bicycle will take the longer time of these two people to move from A to B. For example, suppose there are four tourists. Person 1 takes 1 minute, person 2 takes 3 minutes, person 3 takes 5 minutes, and person 4 takes 10 minutes to move from A to B. If person 1 and person 4 ride the bicycle first, 10 minutes have elapsed when they get to B. If person 4 returns to A with the bicycle, a total of 20 minutes have passed. Your job is to arrange the group to move from A to B using that bicycle within a minimum time.

- **Chinese**

有一個旅行團要從 A 城市到 B 城市旅遊，但他們只有一台協力腳踏車，可供最多 2 個人同時騎乘。由於每個人的腳力不同，使用該協力腳踏車從 A 城市到 B 城市所需的時間取決於速度較慢者。舉例來說，若有四個人，從城市 A 到城市 B 每個人的期程時間分別是第一個人需要一分鐘，第二個人需要三分鐘，第三個人需要五分鐘，第四個人需要十分鐘。那麼第一個人與第四個人先騎，那他們需要 10 分鐘到 B 城，若由第四個人將車子騎回來 A 城，那就已經花了 20 分鐘。現在你的責任是用最少的時間讓全團的人能使用那輛協力車從 A 城到 B 城。

Input Format

- **English**

The first line of the input is an integer m indicating the number of test data. The first line of the each test data is the number of tourists, *n*, and then each of the following lines contains the person id and the time used for that person to ride the bicycle, (*ID*, *time*).
※1≤ n ≤100,000

- **Chinese**

輸入的第一行是一個整數 m 表示共有 m 比測資。每筆測資的第一行是一個整數 n 代表旅行團的人數。之後有 n 行，每行有兩個整數，分別代表團員編號及騎乘那台協力腳踏車從 A 城到 B 城所需的時間。
其中 1≤ n ≤ 100,000。

Output Format

- **English**

The output is the minimum time.

- **Chinese**

每筆測資輸出所需最短時間於一行。

Example

Sample Input:	Sample Output:
3 4 1 1 2 3 3 5 4 10 4 1 12 2 10 3 5 4 1 4 1 7 2 6	20 28 20

3 5 4 1	

A6
tions recipe（魔藥配方）

(Time Limit: 3 seconds)

Problem Description

● English

Magician modulation of the potions have a variety of effects, to allow people to fly, invisible, powerful, swords and guns, etc., but the secret medicine is not easily modulated from, in addition to the material are hard to get, improper order of adding the material may also lead to poor efficacy..

The potency of the potion in addition to the type and amount of the material, the order of addition is also very important. If the order is not correct, the results of each drug interaction will decrease the efficacy. The potency of potions is the number of ingredients added in the correct order. For example, there are four kinds of materials, which are numbered 1, 2, 3, 4 in the order that they must be added and the amount of 2 is doubled. If they are adding by the order 1→2→2 →4→3, There are 4 of these materials adding by the correct order, the efficacy of strength is 3. If they are adding by the order of 3→2→4→1→2, which coincides with the correct sequence of the longest sequence is 3→4 , 2→4, 2→2 or 1→2, then the efficacy of strength can only reach 2.

When the magician taught apprentice, in order to keep controlling them, he often secretly stay in hand. He will only tell them all kinds of potions of medicine and weight, but not to tell them the order of adding the herbs. So the disciples will have to try by themselves. Now the problem is given the order of the disciple used to make the potion, could you tell the potency of the potion?

● Chinese

魔法師們調製出的秘藥有著各種各樣的效果，能夠讓人飛行、隱形、力大無窮、刀槍不入等等，但是秘藥也並非能輕易調製而成的，除了因為材料取得困難之外，調配順序不正確也可能導致藥效不彰。

魔藥調合的成效除了與材料的種類和份數有關之外，加入的先後順序也很重要，若加入順序不正確時，每個藥材相互作用的結果就會讓藥效強度打折扣，魔藥藥效強度即為依正確順序加入之藥材數量。舉例來說：現在有 4 種材料，依照必須加入的順序編號為 1、2、3、4，且編號 2 的藥材需要份數是兩份。若是以 1→2→2→4→3 的順序調製的話，因為有 4 份藥材有依照正確的相對先後順序加入，則藥效強度為 4，但若是以 3→2→4→1→2 的順序調製，其符合正確先後順序的最長序列為 3→4, 2→4, 2→2 或是 1→2，則藥效強度只能達到 2。魔法師傳授徒弟的時候，為了怕徒弟全學會了難以管制，往往暗留一手，只會告訴他們各種魔藥的藥材和分量，卻沒有告訴他們各藥材加入的順序。於是各弟子按自己的意思和經驗嘗試調製出各種不同的魔藥，想請問你每個弟子所調製的魔藥藥效強度為何？

Technical Specification

- **English**
  - Each potion needs no more than 1,000,000 kinds of materials
  - The number of herbs required for each potion does not exceed 1,000,000, and the herbs may be duplicated in a potion.
  - The correct order of adding the material is in the order of the material number
- **Chinese**
  - 魔藥需要的材料種類編號從 1 到 1,000,000 種
  - 每種魔藥需要的材料總份數 m 不會超過 1,000,000，同一材料可能有多份
  - 正確的加入順序即按材料編號順序加入

Input Format

- **English**

The first line of the input is a positive integer n indicating the number of test data. Each data has several non-negative integers (end by 0) in one line, represents the order of the material added by the disciple.

- **Chinese**

輸入第一行為一個正整數 n 代表共有 n 筆測資。之後有 n 行，每行為一筆測資，有數個非負整數，以 0 結尾，代表學徒將各藥材加入的順序。

Output Format

- **English**

For each test data, output the potency of the potion in one line.

- **Chinese**

每筆測資輸出該魔藥強度於一行。

Example

Sample Input:	Sample Output:
2 5 6 8 7 6 4 3 8 2 1 0 4 5 6 8 7 9 4 2 1 0	4 5

A7
The Disjunctive Domination Problem on Trees

(Time Limit: 3 seconds)

Problem Description

- **English**

Let $G = (V, E)$ be a graph. A vertex subset $D \subseteq V$ is a disjunctive dominating set of G if every vertex v in $V \setminus D$ is adjacent to a vertex of D or has at least two vertices in D at distance two from v. The disjunctive domination number of G is the cardinality of a minimum disjunctive dominating set of G. Given a tree T with n vertices, please write a program to compute the disjunctive domination number of T.

- **Chinese**

令 $G = (V, E)$ 為一個圖形，點集合 $D \subseteq V$ 稱為 G 的一個分離支配集，則任一個點 v 屬於 $V \setminus D$ 必須連接一個在 D 中的點或是至少有二個在 D 中的點和 v 的距離為 2。而一個圖形 G 的分離支配數則是其最小分離支配集的元素個數。給定一個含有 n 個點的樹 T，請寫一個程式計算 T 的分離支配數。

Technical Specification

- **English**

The range of the number n is from 1 to 50.

- **Chinese**

數字 n 的範圍從 1 到 50

Input Format

- **English**

The first line is an integer indicating the number of test cases. Each test case contains n lines. The first line indicates the number n, *i.e.*, the number of vertices in T. For the remaining n-1 lines, each line contains two numbers separating by a space to denote an edge. That is, $V = \{1, 2, \ldots, n\}$.

- **Chinese**

第一行是一個整數代表測試資料有幾筆。每一筆測試資料包含 n 列，第一列是一個整數代表樹 T 的點數，接下來的 n-1 列，每一列含二個整數，中間以一個空白隔開，代表一個邊。也就是點集合 $V = \{1, 2, \ldots, n\}$。

Output Format

- **English**

For each test case, output the disjunctive domination number of T.

- **Chinese**

針對每一筆測試資料，輸出樹 T 的分離支配數。

Example

Sample Input:	Sample Output:
2 7 1 2 2 3 3 4 4 5 5 6 6 7 9 1 2 2 3 2 4 4 5	2 2

4 6 4 7 7 8 7 9	

A8
破解古老文字 Cracking Ancient Text

(Time Limit: 3 seconds)

Problem Description

● English

You were a historical linguist. One day you got an ancient text that was written in some unknown language. After carefully inspecting this text, you found several surprising facts. First, there are exactly 52 letters in this language. Second, each word is taken as a contiguous fragment of the letters. Third, words are non-overlapped in the text. Fourth, there is no variation of words such as tense or plural forms. The most surprising fact is, there is no punctuation or space to separate distinct words! Its structure looks like ancient Chinese text that a reader has to learn the skill to add punctuation symbols by himself/herself in order to identify individual words. Thus, you need to label every word in the text. That is, try to locate its starting and ending positions in the text. Fortunately, you have a dictionary of the language that can help you identify the words in the text literally. In the ideal case, you can decompose the text as a series of words using the dictionary. However, this is not an easy task to be done because there is no clear boundary between two consecutive words, which would cause some degree of ambiguity. Moreover, your dictionary is incomplete, and thus many words are not included in your dictionary. Instead of this, you relax your goal and try to reduce the number of characters that are not labelled as part of the words in the text. For example, suppose the text is **abababb** and your dictionary is {**ab**, **ba**}. The first way to mark the text is as **ab** **ab** **ab** **b**, which leaves 1 character unmarked. Another way is as **a** **ba** **b** **ab** **b**, which leaves 3 characters unmarked. Accordingly, the first one is better than the second. Please write a program to find the minimum way to accomplish this task.

● Chinese

你是一位歷史語言學家，某日你找到一篇由某種文字所撰寫的古文，經過你仔細的審視，你發現了幾項驚人的事實。首先，你發現這篇古文，共使用 52 個字母；第二，古文中的單字是由連續片段的字母所組成；第

三，古文中的單字不會互相重疊；第四，單字沒有諸如時態或是複數之類的詞性變化。麻煩的是，單字間並沒有標點符號或是空白區隔！其結構有點類似中文的古文，需由讀者自行標註句讀並且斷詞。你需要標示這篇古文中每個單字所出現的位置，換言之，要確定每個單字開始與結束的位置。所幸你有一本關於此古文的字典，這可以幫助你進行單字的標示。理想上，應該是可以透由字典把古文拆解成一個個串接的單字，但這並不是件容易的事，因為相鄰的兩個單字間並沒有分隔的記號，因此辨識上會遇到某些歧異性。此外，你的字典並不完整，許多字在你的字典裡根本找不到。因此你調整你的目標，你相信，如果能降低沒被標註成單字的字元的總數，這樣應該有助於斷字的進行。比如說，假設古文為 **abababb**，而你的字典為{**ab**, **ba**}。第一種拆解法是 **ab ab ab b**，留下 1 個未被標註的字元；第二種拆解法是 **a ba b ab b**，留下 3 個未被標註的字元，顯然前者比後著優。你的任務是撰寫程式，找出能留下最少個未被標註的字元的標註方式。

Technical Specification

- **English**
  - There are at most 7 test cases. Each test case has at most 10 ancient texts.
  - The alphabet of the text is represented by the uppercase and lowercase English letters. There are 52 such letters.
  - Let n be the length of the text (i.e. the total number of characters), m be the number of words in the dictionary, and r be the length of the longest word in the dictionary. Then $1 \le n \le 100000$, $1 \le m \le 1000$ and $1 \le r \le 100$.
  - The words in the dictionary are distinct.
- **Chinese**
  - 測資至多有 7 筆，每筆至多有 10 段古文。
  - 古文所使用的字母集，以大小寫英文字母共 52 個表示之。
  - 令 n 為古文之字元總數，m 為字典所包含的總單字數，r 為字典中最長的單字的字元數，則 $1 \le n \le 100000$、$1 \le m \le 1000$ 且 $1 \le r \le 100$。
  - 字典中所有的單字皆相異。

Input Format

- **English**

The first line is an integer indicating the number of test cases. Each test case

contains two parts. The first part specifies the dictionary and the second part shows the ancient texts. The first part starts with the integer m in a line that gives you the number of words in the dictionary. Next, m lines follow, and each line gives you one word in the dictionary. Then the second part starts with an integer k, and the next k lines show you the ancient texts, one per line.

- **Chinese**

測資的第一行有一個整數，標示測資的總筆數。單筆測資共有兩部分，第一部分描述字典，第二部分描述古文。第一部分首行為字典之總字數 m，其餘 m 行各表示字典中一個單字。接下來第二部分首行為古文段落數 k，其餘 k 行各表示一段古文。

Output Format

- **English**

For each ancient text, find the optimal way for labelling and output its total number of unmarked characters.

- **Chinese**

針對每段古文找出最佳解，並輸出此時未被標註之字元總個數。

Example

Sample Input:	Sample Output:
2 2 a B 1 aaaAa 2 aB aBa 2 aBaBaa	1 1 7

aaaaaaa	

A9

Borobudur Restoration

(Time Limit: 3 seconds)

Problem Description

- **English**

Borobudur is a 9th-century Mahayana Buddhist temple in Central Java, Indonesia, which is the world's largest Buddhist temple, and also one of the greatest Buddhist monuments in the world. The temple consists of nine stacked platforms, six square and three circular, topped by a central dome. The temple is decorated with thousands of relief panels and Buddha statues. The central dome is surrounded by 72 Buddha statues, each seated inside a perforated stupa. The King moved the capital of the Medang Kingdom to the region of East Java after a series of volcanic eruptions, leading to the abandonment of Borobudur temple. As a consequence, Borobudur lay hidden for centuries under layers of volcanic ash and jungle growth.

Under British administration, the governor was informed about the discovery of Borobudur temple deep in a jungle. However, the entire temple was shattered into pieces without any complete statues or relief anymore. The restoration of Borobudur then was carried out using the principles of anastylosis by the United Nations Educational, Scientific and Cultural Organization (UNESCO), which aims to piece together the original statues and relief if two pieces come from the same place. However, it is extremely difficult and time-consuming to reconstruct the huge Borobudur temple. Owing to the advance of 3D scanning technologies, all these objects can now be scanned and digitalized, which allows computer software to piece together the broken objects. Each temple piece is represented as a string (e.g., ABC). And two temple pieces can be assembled into one piece if there are overlapping matched patterns (e.g., ABC and BCD can be merged into ABCD). Note that the lengths of stone strings can be different.

Unfortunately, because of the invaluable heritage of Borobudur, many countries and private sectors are competing to be the first team assembling the temple. Therefore, the UNESCO decided to encode all the stone pieces in order to

protect it from hacked and stolen by others. Each stone piece is appended with an exclamation point (e.g., ABC!). The circular rotations of all stone pieces are generated and sorted. For example, given two pieces: ABC! and BCD!. Eight circular rotations, ABC!, BC!A, C!AB, !ABC, BCD!, CD!B, D!BC, !BCD will be generated. The rotated messages are then sorted into lexicographical order (i.e., !ABC, !BCD, ABC!, BC!A, BCD!, C!AB, CD!B, D!BC). The last letters in the sorted strings (i.e., CD!A!BBC) form the final encoded string.

UNESCO then only releases this encoded string instead of original stone pieces. Each team can only collect partial stone pieces of the temple and identify the other pieces that can be assembled with the one at his hand from this encoded string. In this way no single company can obtain the entire set of stone pieces. The company that assemble most pieces will own this heritage for twenty years. In order to find the right stone piece to assemble, a team needs to know how many stone pieces matched with the one at his hand. For instance, given the same encoded stone string (CD!A!BBC), if the company holds two stone pieces "AB" and "BC," the answers are 1 and 2, respectively, because "AB" appears once and "BC" appears twice in the original stone strings (i.e., ABC and BCD). Note that overlapping strings are counted independently. For instance, "AA" appears twice in the "AAA" string

Given the encoded string of all stones (e.g., CD!A!BBC) and unencoded strings of partial stones (e.g., "AB" and "BC"), you are asked to output the frequency for each of the partial stone strings appeared in the original stone strings.

● Chinese

婆羅浮屠位於印尼爪哇島中央，是一座第九世紀大乘佛教所建立且現為世界最大之佛寺之一。這座佛寺包含九層平臺，六個方型台，與三座圓形台，圍繞著中央一座巨大的圓頂。整座佛寺用數以千計之浮雕與佛像裝飾。其中央圓頂被 72 尊座落於佛塔內之佛像所環繞。然而由於多次火山爆發，國王決定將首都遷往爪哇島東部，進而遺棄了婆羅浮屠佛寺。經過了數百年後，婆羅浮屠深埋在火山灰與叢林之中。

在英國統治印尼時期，當地執政者有一天被通知在叢林中有佛寺遺跡。然而整座佛寺已經被斷成無數石塊，沒有任何一尊佛像或浮雕被完整保留。為了重建此佛寺，聯合國教育、科學及文化組織(UNESCO)決定用考古學原物歸位方式將所有石塊拼湊回去。然而拼湊數以萬計之石塊實為巨大工程。

藉由 3D 電腦掃描技術，每一片石塊能夠被數位化保存，進而利用電腦技術將其拼湊回來。每一石塊數位化後皆變成一字串(例如 ABC)，而二片石塊若有重疊部分則代表可拼湊在一起。例如，ABC 和 BCD 可拼湊為 ABCD。請注意石塊字串長度可能不一。

然而由於婆羅浮屠是無價之世界遺產，許多國家和私人公司都想成為的一個重建此佛寺之團隊，來獲得所有權。因此，世界遺產組織決定先將所有掃描的石塊數位資訊進行加密以免被盜用。每一石塊所對應之數位字串的字尾都加上一驚嘆號(例如 ABC!)。接著產生每一字串的所有旋轉字串並做字母排序。例如，假設有二個石塊字串 ABC!和 BCD!，總共會有八個旋轉字串產生出來：ABC!，BC!A，C!AB，!ABC，BCD!，CD!B，D!BC，!BCD。此八個旋轉字串進行字母排序後變成：!ABC、!BCD、ABC!、BC!A、BCD!, C!AB、CD!B、D!BC)。這些排序好字串的最後一個字母串聯起來就形成加密過的字串(CD!A!BBC)。

UNESCO 決定將此加密過的字串開放給所有人。每一個團隊一次只能看到部分石塊，必須利用這加密字串，推測其手中每一石塊可與哪一石塊組合，而能最快協助推測出最多組合答案的團隊，將能獲得婆羅浮屠二十年的所有權。因此為了找出每一石塊能跟那些石塊拼湊，必須能迅速地知道該石塊在全部石塊中出現之頻率。例如，假設給定前一個加密過字串(CD!A!BBC)，則二個石塊"AB"和"BC"在全部原始石塊字串(此例為 ABC 和 BCD)出現頻率分別為一次與二次。請注意要獨立計算重疊字串之出現頻率。例如"AA"視為在"AAA"字串中出現二次。

給定一加密過之全部石塊字串，和一群未加密之石塊字串，請輸出每個未加密石塊字串在全部原始石塊字串中所出現之頻率。

Technical Specification

- **English**
  - Each letter in the original stone string is from standard alphabet {A, B, C, …, Z} and each original string always ends with an exclamation point '!'.
  - Each original stone string has length from 3 to 1000.
  - The number of original stone string ranges from 1 to 1000.
  - The length of encoded string ranges from 3 to 100000 letters from alphabet {A, B, C, …, Z, !}.
  - The number of query/unecoded stone strings ranges from 1 to 10.

- Each query/unencoded stone string has length from 1 to 1000 and each letter comes from alphabet {A, B, C, …, Z}.

- **Chinese**
  - 每個原始石塊字母是標準英文字母集{A, B, C, …, Z}*並以驚嘆號!做結尾。
  - 每個原始石塊字串長度從 3 到 1000。
  - 原始石塊字串個數為 1 到 1000。
  - 加密過石塊字串長度最長僅介於 3 到 100000 個字母之間，字母來自{A, B, C, …, Z, !}。
  - 待查詢未加密石塊字母個數介於 1 到 10 之間。
  - 每一未加密石塊字母長度介於 1 到 1000 之間，且字元來自{A, B, C, …, Z}。

Input Format

- **English**

The first line contains the number of test cases. Each test case starts with a line containing the encoded stone string. The next line is the number of query/unencoded stone strings (*N*). Each of the following *N* lines stores one unencoded/query stone string.

- **Chinese**

第一行包含一個數字代表測資總數量。每一筆測資第一行是加密過的全部石塊字串。下一行為一個數字代表要查詢的未加密石塊字串個數(N)。每一之後 N 行代表一個未加密石塊字串。

Output Format

- **English**

For each test case, output the frequency of each query stone string in one line.

- **Chinese**

針對每一筆測試資料內的每一個石塊字串，於每一行依序輸出其出現於所有原始石塊字串中之頻率。

Example

Sample Input:	Sample Output:
1 CD!A!BBC 3 AB BC ABCD	1 2 0

A10
Linear Extensions

(Time Limit: 2 seconds)

Problem Description

● English

Given a set S, a partially ordered relation is usually adopted to describe the order over some elements of S. If every two elements are ordered, then the partially ordered relation is said to be totally ordered. Precisely speaking, let $S = \{a_0, a_1, \ldots, a_{n-1}\}$. A binary relation on S can be represented by an nxn 0-1 matrix A, where two elements a_i and a_j are related if $A_{ij} = 1$. A partially ordered relation A on S satisfies

for $a_i \in S$, $A_{ii} = 1$.

for $a_i, a_j, a_k \in S$, if $A_{ij} = 1$ and $A_{jk} = 1$, then $A_{ik} = 1$.

for $a_i, a_j \in S$ and $i \neq j$, $A_{ij} = 1 \Rightarrow A_{ji} = 0$.

If A is a partial order and satisfies

for $a_i, a_j \in S$, $A_{ij} = 0 \Rightarrow A_{ji} = 1$,

then A is a total order relation on S.

A linear extension L of a partial order A is a totally ordered relation that preserves the partial order defined by A, that is, L defines a total order on S and

$\forall \{a_i, a_j\} \subseteq S \qquad A_{ij} = 1 \Rightarrow L_{ij} = 1$.

For example, let A be the matrix [[1, 0, 0], [0, 1, 1], [0, 0, 1]]. The possible linear extensions of A are [[1, 1, 1], [0, 1, 1], [0, 0, 1]], [[1, 0, 1], [1, 1, 1], [0, 0, 1]], and [[1, 0, 0], [1, 1, 1], [1, 0, 1]] so there are 3 linear extensions. Given a partially ordered relation A, count the number of linear extensions of A.

● Chinese

給一集合 S，一偏序關係通常用於描述 S 中某些元素的順序。若任二元素皆有序，則稱此偏序關係為全序。精確的說，令 $S = \{a_0, a_1, \ldots, a_{n-1}\}$。一 S 上的二元關係可以一 nxn 的 0-1 矩陣表示，其中元素 a_i 與 a_j 有關則設 $A_{ij} = 1$，反之則設 $A_{ij} = 0$。一 S 上的偏序關係 A 滿足

對 $a_i \in S$, $A_{ii} = 1$。

對 $a_i, a_j, a_k \in S$, 若 $A_{ij} = 1$ 且 $A_{jk} = 1$，則 $A_{ik} = 1$。

對 $a_i, a_j \in S$ 且 $i \neq j$，$A_{ij} = 1 \Rightarrow A_{ji} = 0$。

若 A 為一偏序關係且滿足

對 $a_i, a_j \in S, A_{ij} = 0 \Rightarrow A_{ji} = 1$,

則 A 為 S 上的一全序關係。

偏序 A 的線性擴展 L 為一保有偏序 A 的全序關係，即 L 為 S 上的全序關係，且

$\forall \{a_i, a_j\} \subseteq S \qquad A_{ij} = 1 \Rightarrow L_{ij} = 1$。

舉例來說，令 A 為矩陣 [[1, 0, 0], [0, 1, 1], [0, 0, 1]]。可能的線性擴展為 [[1, 1, 1], [0, 1, 1], [0, 0, 1]]、[[1, 0, 1], [1, 1, 1], [0, 0, 1]] 以及 [[1, 0, 0], [1, 1, 1], [1, 0, 1]]，故線性擴展個數為 3。 給一偏序關係 A，計算 A 的線性擴展個數。

Technical Specification

- **English**
  - The number of test cases is at most 5. For each test case, what given is an *n*x*n* matrix, where $n <= 300$, and the solution is at most 1100.
- **Chinese**
  - 測試資料筆數介於 1 到 5。每筆測試資料中，所給的為一 n 階方陣， $n <= 300$，且答案值至多為 1100。

Input Format

- **English**

The first line is an integer indicating the number of test cases. Each test case consists of n+1 lines, where the first line is the positive integer n indicating the size of S. Each of the following n lines is a row of the partially ordered relation A on S, where each entry is either 0 or 1, and two consecutive entries are separated by a space.

- **Chinese**

第一行是一個整數代表測試資料有幾筆。每一筆測試資料由 n+1 行構成，其中第一行為正整數 n，表示 S 的元素個數。以下 n 行每一行為偏序關係 A 之一列，含 n 個元素，相鄰元素以一空白分隔。

Output Format

- **English**

For each test case, output the number of the requested number of linear extensions.

- **Chinese**

針對每一筆測試資料，輸出所求之線性擴展個數。

Example

Sample Input:	Sample Output:
2 5 1 1 1 1 1 0 1 1 1 1 0 0 1 0 1 0 0 0 1 1 0 0 0 0 1 4 1 0 1 0 0 1 0 1 0 0 1 0 0 0 0 1	2 6

A11
三個時間點(Three Points of Time)

(Time Limit: 2 seconds)

Problem Description

- **English**

There are N time intervals. The i-th time interval is represented by [s(i),t(i)], where s(i) and t(i) are the starting and the finishing time. In addition, there is a weight w(i) associated with the interval. You are asked to choose three points of time to maximize the total weight of intervals containing at least one of your choices. An interval [s(i),t(i)] contains a point of time x if and only if $s(i) \leqq x \leqq t(i)$.

- **Chinese**

有 N 個時間區間，第 i 個時間區間以[s(i),t(i)]來表示，其中 s(i)是開始時間而 t(i)是結束時間，此外，每個區間有一個權重值 w(i)。請你選擇三個時間點，目標是使得至少包含一點的區間權重總合為最大。一個區間[s(i),t(i)]包含一個點 x 的定義為：$s(i) \leqq x \leqq t(i)$。

Technical Specification

- **English**
  - $N \leqq 70000$.
  - All s(i), t(i), w(i) are integers
  - $0 \leqq s(i) \leqq t(i) \leqq 2^{30}$.
  - $1 \leqq w(i) \leqq 1000$.
- **Chinese**
  - $N \leqq 70000$.
  - s(i), t(i), w(i) 均為整數
  - $0 \leqq s(i) \leqq t(i) \leqq 2^{30}$.
  - $1 \leqq w(i) \leqq 1000$.

Input Format

- **English**

  The first line is an integer indicating the number of test cases. There are at most 8 test cases. Each test case starts with a line containing N, and the following N lines are the data for the intervals, one line for one interval. Each line contains three numbers, *i.e.*, s(i), t(i), and then w(i), separated by a space.

- **Chinese**

  第一行是一個整數代表測試資料有幾筆。測資筆數不超過 8，每一筆測試資料第一行是整數 N，接下來 N 行是 N 各區間的資料，一行表示一個區間， 每一行包含三個整數依序分別為 s(i), t(i)與 w(i)，中間以一個空白隔開。

Output Format

- **English**

  For each test case, output the maximum total weight in one line.

- **Chinese**

  針對每一筆測試資料，輸出最大權重總和，每筆輸出後輸出一個換行。

Example

Sample Input:	Sample Output:
1 7 0 1 1 1 3 2 2 5 4 4 10 1 4 6 2 7 8 2 10 11 1	12

A12
Longest Journey

(Time Limit: 3 seconds)

Problem Description

● English

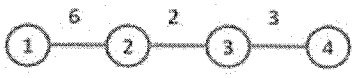
Tony is a local guide working at a travel agency, but actually he hates the job very much because he finds the tourists noisy. To make his work more fun, he decides to find an official way to waste the tourists' time, by arranging the longest journey. The journey that Tony wants to arrange contains n sites, labeled by $V_n = \{1, 2, \ldots, n\}$, and these sites are connected by roads to form a tree T (i.e., there is exactly one path to go from one site to another). Each edge in the tree is associated with a positive value to represent its length. We define a journey J as a permutation π of V_n, i.e., $\langle \pi(1), \pi(2), \ldots, \pi(n) \rangle$, where $\pi(i)$ denotes the ith site to visit. The distance between two nodes x and y in the tree T is denoted by $\text{dist}(x, y)$, and the length of J is the summation of distances between successive sites in J, which is

$$\sum_{i=1 \text{ to } n-1} \text{dist}(\pi(i), \pi(i+1)).$$

So, a journey has to visit every site exactly once, but between successive sites, we may by-pass other sites without visiting them. Furthermore, while travelling from a site to the next site in a journey, we must only travel on the shortest path. (Otherwise, the tourists would notice that you are wasting their time.) Tony wants to know how much time he can waste, so his target is to find the maximum length of a journey, among all possible permutations of sites.

Given the input tree of n sites with the lengths of the edges, write a program to help Tony to compute the length of the longest journey.

Example:



Suppose we have four sites 1, 2, 3, 4, forming a tree as shown in the above figure, where each edge is labeled with the distance between its endpoints. A journey ⟨1, 4, 2, 3⟩ will have length = 11 + 5 + 2 = 18, while another journey ⟨2, 4, 1, 3⟩ will have length = 5 + 11 + 8 = 24. The latter journey happens to be one of the longest journeys in this example, so that we should print 24 as the output.

● Chinese

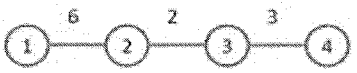
東尼是一個在旅行社工作的本地導遊，但是他很討厭這份工作，因為旅客總是很吵。為了使工作更有趣，他決定要安排最長的旅程來浪費旅客們的時間。東尼負責的旅程要包含 $V_n = \{1, 2, \ldots, n\}$ 裡的 n 個景點，而這些景點以道路相連成一棵樹 T（即任兩景點間只有剛好一條路徑）。樹每一條邊的長度為一正數值。我們定義一個旅程 J 為對景點集合 V_n 之元素的一個排列π，即 ⟨π(1), π(2), …, π(n)⟩，其中π(i)表示旅程 J 中第 i 個拜訪的景點。另外，我們以 dist(x,y)表示景點 x 和景點 y 在樹 T 之間的距離，而旅程 J 的長度則是依次拜訪景點時，所有相鄰景點之間的距離總和，即

$$\sum_{i=1 \text{ 至 } n-1} \text{dist}(\pi(i), \pi(i+1)).$$

故此一個旅程必須拜訪所有景點剛好一次，但是我們可以經過一個景點而不去拜訪它。再者當拜訪完一個景點到下一個要拜訪的景點時，我們必須走最短路徑。(如果我們浪費時間太明顯，旅客們會發現。) 東尼想知道他最多能浪費旅客多少時間，故他想計算出在不同排列下，最長旅程的長度。

現有輸入為 n 個景點所形成的樹 T 及 T 中各邊的長度，請寫一個程式來幫助東尼計算最長旅程的長度。

範例：



假設有四個景點 1, 2, 3, 4 如上圖所示形成一樹，其中各邊所標註的為其端點間之長度。若旅程為 ⟨1, 4, 2, 3⟩ 時，則有長度 11 + 5 + 2 = 18，而若旅程為 ⟨2, 4, 1, 3⟩ 時，則有長度 5 + 11 + 8 = 24，且後者之旅程實為此範例

的最長旅程之一，故此我們在此範例下應輸出 24 為結果。

Technical Specification

- **English**
  - There are at most 20 test cases.
  - The number n is an integer from 1 to 100,000.
  - Length of each edge is an integer from 1 to 1,000.
- **Chinese**
  - 至多 20 筆測試資料
  - 數字 n 為整數，範圍從 1 到 100,000
  - 各邊長皆為整數，範圍從 1 到 1,000

Input Format

- **English**

  The first line is an integer indicating the number of test cases. Each test case contains n lines, where the first line contains a positive integer n, and each of the next $n - 1$ lines contains three positive integers a, b, and c (separated by a white space), representing an edge that connects a and b, whose length is c.

- **Chinese**

  第一行是一個整數代表測試資料有幾筆。每一筆測試資料包含 n 行，其中第一行為一個正整數 n，之後的 $n-1$ 行各有三個整數 a, b, c （以一個空白隔開），代表一條以 a 和 b 為端點、長度為 c 的邊

Output Format

- **English**

  For each test case, output the length the longest journey in one line.

- **Chinese**

  針對每一筆測試資料，以一行輸出最長旅程的長度。

Example

Sample Input:	Sample Output:
2 4 1 2 6 2 3 2 3 4 3 6 1 2 5 2 3 9 2 4 4 4 5 8 5 6 7	24 94

A13
Treasure

(Time Limit: 2 seconds)

Problem Description

● English

There are n rooms connected by m passages in Mark's dungeon. Each passage connects two distinct rooms, and these two rooms are considered adjacent. For convenience, let us number the rooms from 1 to n. One day, you received the map of Mark's dungeon and found the following truths.

1. There is always a path between any pair of rooms.
2. There are k entrances of Mark's dungeon. You may send a probe into room v_i via entrance i.
3. A treasure chest is located in room 1.
4. An empty chest located in room n.

You plan to send your only probe to get the treasure, but Mark has casted a spell to protect it. The probe will be out of control immediately after entering any room of Mark's dungeon. When the probe enters room v, it will randomly go to a room adjacent to room v with probability $1/d_v$ if room v has d_v adjacent rooms. Luckily, there is still some good news for you. The teleportation function of the probe will be still working, and the chest will be transported to your house once the probe enters room 1 or room n. On the other hand, the bad news is that the probe will be destroyed immediately by Mark's spell after the teleportation has been done. Therefore, you cannot get the treasure if the probe enters room n before room 1. Write a program to compute the probabilities of getting the treasure by sending the probe to rooms v_1, …, v_k respectively. The program should help you to make the best decision.

● Chinese

馬克的地下城有 n 個房間與 m 條連接房間的通道。每條通道都恰好連接兩個相異的房間，而被一條通道連接的兩個房間就會被視為相鄰的。為了

方便起見，我們將房間由 1 到 n 編號。某天，你收到了馬克地下城的地圖，並發現以下的真相：

1. 任兩個房間之間必有路徑。
2. 馬克的地下城有 k 個入口，你可以從入口 i 送探測器到房間 v_i。
3. 有個寶箱放在房間 1。
4. 有個空箱放在房間 n。

你打算送你僅有的一台探測器去取得寶藏，但馬克已施法保護它。探測器只要進入馬克地下城的任一個房間便會失控。當探測器進入房間 v 時，它將以 $1/d_v$ 隨機前往與房間 v 相鄰的任一房間(d_v是與房間 v 相鄰的房間總數)。幸運的是，對你來說總還有些好消息。探測器的傳送功能仍正常運作，它會在進入房間 1 或房間 n 的當下將箱子傳送到你家。另一方面，壞消息是馬克的咒語在傳送完成時會即刻摧毀探測器。因此探測器在進入房間 1 之前便進入房間 n 會導致你無法取得寶藏。撰寫一個程式來計算把探測器經每個入口送進馬克地下城能取得寶藏的機率吧，這程式應該能幫你做出最好的決定。

Technical Specification

- **English**
  - There are at most 10 test cases.
  - The range of the number n is at most 2000.
  - The range of the number m is at most 10000.
  - The range of the number k is at most 2000.
  - Round the numbers in the output to the fifth decimal place.
  - You may assume the sixth decimal place of the answer is not 4 or 5 before rounding.
- **Chinese**
  - 至多有 10 筆測試資料
  - 數字 n 至多 2000
  - 數字 m 至多 10000
  - 數字 k 至多 2000
  - 輸出四捨五入至小數點下第五位
  - 可假定答案在四捨五入前小數點下第六位不是 4 或 5

Input Format

- **English**

The first line is an integer indicating the number of test cases. The first line of each test case contains three numbers n, m and k in one line separating by spaces. n, m, and k are the numbers of rooms, passages, and entrances respectively. The following m lines describe the passages. Each of them contains two integers u and v separated by a blank such that there is a passage connecting room u and room v. The m+1 (last) line of each test case contains k integers $v_1, \ldots, v_k$ to describe the entrances.

- **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包含三個整數分別為 n、m、k，以空白隔開，分別代表房間數、通道數、入口數。接下來的 m 行描述路徑。每一行包含以一個空白隔開的兩個整數 u 和 v，代表有一個通道連接房間 u 與房間 v。每筆測試資料的第 m+1 行(最後一行)有 k 個整數 $v_1, \ldots, v_k$ 描述入口。

Output Format

- **English**

For each test case, output k lines where the i-th line is the probability of getting the treasure by sending the probe to room v_i via entrance i.

- **Chinese**

針對每一筆測試資料，輸出 k 行，其中第 i 行代表由經入口 i 送探測器進房間 v_i，取得寶藏的機率。

Example

Sample Input:	Sample Output:
2 3 2 3 1 2 3 2 1 2 3 4 4 2	1.00000 0.50000 0.00000 0.33333 0.66667

1 3 2 4 2 3 4 1 2 3	