Problem 1. 平、閏年判定

(Time Limit: 1 second)

問題描述 :

試撰寫一個程式，可由鍵盤讀入一個 4 位數的整數，代表西洋的年份，然後判別這個年份是否為閏年（每四年一閏，每百年不閏，每四百年一閏，例如西元 1900 雖為 4 的倍數，但可被 100 整除，所以不是閏年，同理， 2000 年是閏年，因可被 400 整數，而 2004 當然也是閏年，因可以被 4 整除）。

輸入說明:

第一行輸入一正整數 N(1≤N≤10)，表示共有 N 筆測資，每筆測資輸入一 4 位數的整數，表示西元年份。

輸出說明:

輸出閏年(Bissextile Year)或平年(Common Year)。

範例:

Sample Input:	**Sample Output:**
2 2000 2003	Bissextile Year Common Year

Problem 2. 解讀神秘的密碼

(Time Limit: 2 seconds)

問題描述 :

你突然發現螢幕上出現幾行詭異的字串，隨後收到一封電子郵件。信上說這幾行字其實是反轉句，將它還原就可以得到原本想傳達的訊息。請寫出一程式來還原這些反轉句。

輸入說明:

第一行輸入一正整數 N(1≤N≤10)，表示共有 N 筆測資，每筆測資為一英文字串，字串長度最多不超過 50 個字元。

輸出說明:

此字串的反轉句。

範例:

Sample Input:	**Sample Output:**
3 retupmoc ecneics dlroW olleH	computer science Hello World

Problem 3. 吃麵包

(Time Limit: 3 seconds)

問題描述 :

有 N 個麵包在你面前排成一排，每個麵包都有不同的飽足程度，你想知道把其中某一段的麵包吃掉會達到多少飽足度。

輸入說明:

第一行輸入一正整數 S(1≤S≤10)，表示共有 S 組測資，每組測試資料的開頭會有兩個正整數 N, M (1≤N , M ≤100,000)，N 代表麵包的總數量， M 代表欲測試資料的數量，接下來一行有 N 個不超過 1,000 的正整數，依序代表每個麵包的飽足度，接下來 M 行每行有兩個數字 Q，R (1 ≤ Q ≤ R ≤ N)，代表你想要吃掉第 Q 個到第 R 個麵包。

輸出說明：

對每組測資輸出 M 行，代表吃完每組麵包的總飽足程度。

範例:

Sample Input:	**Sample Output:**
1 3 3 1 2 3 1 3 1 2 2 3	6 3 5

Problem 4. 計算分數

(Time Limit: 2 second)

問題描述 :

給定兩個分數 a 及 b ，將 a +b 的結果同樣以分數呈現，且請將 a +b 的結果化為最簡分數。例如 1/2+1/3 ，你要輸出 5/6 。

輸入說明:

輸入的第一行為一個正整數 S(1≤S≤10)，代表共有幾筆測資。

每筆測資包含 2 個分數 a 及 b ，以逗點隔開。每個分數都是以 x/y 的格式表示，其中 x 和 y 都是整數。

輸出說明:

對於每筆測資，請輸出 a +b 的結果，並以分數表示，且分數要化為最簡分數。

範例:

Sample Input:	**Sample Output:**
2 1/2,1/3 1/2,1/6	5/6 2/3

Problem 5. A long railway

(Time Limit: 3 seconds)

Problem Description

There is a long railway having N stations. From north to south, the stations are labelled consecutively from 0 to N-1. There are M trains, and each of them goes back and forth between two terminal stations. For train i, the two terminal station are denoted by s1(i) and s2(i), where 0<=s1(i)<s2(i)<N. Each train always makes a brief stop at every station between s1(i) and s2(i), including the two terminal stations.

Now, we want to organize a tour from station 0 to station N-1. However, there may be no train go directly from station 0 to N-1, and in this case we need to transfer several times. Since transferring from trains to trains is troublesome, your job is to compute the minimized number of trains.

For example, suppose that N=10, and there are 5 trains. The terminal stations are as follows: s1(0)=0, s2(0)=4; s1(1)=0, s2(1)=3; s1(2)=4, s2(2)=7; s1(3)=6, s2(3)=9; s1(4)=5, s2(4)=8. Then, we can take train 0 from station 0 to 4, and transfer by train 2 from station 4 to 6, and finally by train 3 from station 6 to 9. Therefore the minimized number of trains is three.

Input Format

The first line has an integer t ($1 \le t \le 10$) representing the number of test cases.
For each test case, the input of a test case consists of three lines. The first line contains two integers N and M, 1<N<=10000 and 0<M<1000, which are the numbers of stations and trains, respectively. The second line contains M-1 integers s1(0), s1(1),…,s1(M-1); and the third line contains s2(0), s2(1),…,s2(M-1). Any two consecutive numbers in the same line are separated by a space.

Output Format

Output the minimized number of trains in one line. If there is no solution, then output -1.

Example

Sample Input:	**Sample Output:**
1 10 5 0 0 4 6 5 4 3 7 9 8	3