

Problem 1. 考試測驗

(Time Limit: 1 second)

問題描述：

某間學校舉辦英文檢定測驗，若是學生通過該測驗，則能通過畢業門檻。老師準備的英文測驗中分為三個項目當作考試，分別是聽力、閱讀、口說。每一個項目測驗滿分皆為 100 分，總分為 300 分。想要通過測驗有兩種方式。方式一：若是三個項目分數皆為 60 分以上（包含 60 分）即為通過測驗。方式二：若三個項目中有其中一個項目分數未滿 60 分，但三個項目分數總和超過 220 分，也可算通過測驗。若三個項目中有其中一個項目分數未滿 60 分，而三個項目分數總和也沒能超過 220 分，可獲得補考機會。若是三個項目中有兩個項目不及格，但另一個項目成績高於 80 分（包含 80 分），也可獲得補考的機會。其餘皆判定為無法通過測驗。

輸入說明：

第一行為一個整數 N ，代表共有 N 組測試資料。之後有 N 行，每一行有 3 個非負整數（範圍皆為 0 到 100），分別代表該名學生聽力、閱讀、口說的測驗分數。

輸出說明：

若是通過測驗，則輸出“P”。若是需要補考，則輸出“M”。若是無法通過測驗，則輸出“F”。每組答案結果輸出於一行。

範例：

Sample Input:	Sample Output:
5	P
95 86 100	M
30 60 80	P
80 55 88	F
20 30 60	M
10 80 10	

Problem 2. 求質數問題

(Time Limit: 2 seconds)

問題描述：

輸入一個整數 n ，求小於等於 n 的最大質數。

輸入說明：

從鍵盤輸入一個整數 n ， $1 < n \leq 1000$ ，且 n 為整數。

輸出說明：

輸出小於或等於 n 的最大質數。

範例：

Sample Input:	Sample Output:
10	7
100	97

Problem 3. 對話機器人

(Time Limit: 2 second)

問題描述：

我們想要開發能與人自動對話的機器人，初步功能希望能達到問候的功能，也就是說當人跟機器人問候時，機器人也會回覆問候語。為了使對話機器人的回應不會一成不變，請設計輪流回覆不同的問候語。輸入和回應的問候語包含："Hi", "Hello", "How do you do" 和 "How are you"。若收到第一次問候語，則回答 "Hi"；第二次則回答 "Hello"；第三次則回答 "How do you do"；第四次則回答 "How are you"；第五次又從 "Hi" 回答起，依此類推。若輸入不屬與上述問候語，則回應 "Sorry"，然後重新從 "Hi" 開始依序回答。

輸入說明：

程式輸入若干行測試資料，每一行包含一句問候語。字母大小寫不同，視為不同字串。

輸出說明：

對於每一筆測試資料，輸出回應的問候語於一行中，然後換行。

範例：

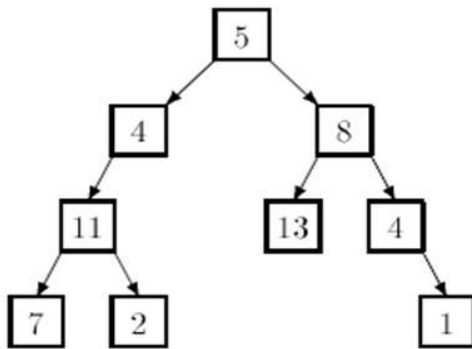
Sample Input:	Sample Output:
Hi	Hi
Hi	Hello
Hello	How do you do
How do you do	How are you
Hello	Hi
Bye	Sorry
How are you	Hi

Problem 4. 樹的走訪方式

(Time Limit: 3 seconds)

問題描述：

給定一個二元樹狀結構，每個節點儲存一個大於零的正整數，寫一個程式以前置（pre-order）走訪樹的方式，輸出走訪節點所儲存的數值。例如下圖二元樹狀結構的前置走訪輸出為：5 4 11 7 2 8 13 4 1。相鄰的數字用單一空格區隔。



一個二元樹狀圖可以利用由上而下、由左至右的方式逐層描述每個節點的內容。節點內容包含三個數字，第一個數字表示節點儲存的數值，第二個數字表示此節點是否存在左子節點（left child），第三個數字表示此節點是否存在右子節點（right child）。存在子節點用 1 表示，不存在用 0 表示。假設相鄰的任兩個數字用單一空格區隔，則上圖二元樹狀結構的表示方式為：

5 1 1 4 1 0 8 1 1 1 1 1 1 1 1 1 1 3 0 0 4 0 1 7 0 0 2 0 0 1 0 0

輸入說明：

測試資料包含若干行，每一行資料為一個二元樹狀結構的數字表示方式（必須符合規定）。每個二元樹至少包含一個節點。

輸出說明：

針對每一筆測試資料以前置走訪方式輸出節點所儲存的數值於一行中，數值間以一個空格隔開，行末進行換行。

範例：

Sample Input:	Sample Output:
1 1 1 2 0 0 3 0 0	1 2 3
5 1 1 4 1 0 8 1 1 1 1 1 1 1 1 1 1 3 0 0 4 0 0	5 4 11 8 13 4

Problem 5. Counting dominance in two sequences

(Time Limit: 3 seconds)

Problem Description

Given two arrays f and g , compute the number of pairs (i, j) that satisfies $f[i] > g[j]$.

Input Format

The input starts with an integer in a line, which is the number of test cases. Each case consists of three lines. The first line is the number of elements in the arrays, and the second and third lines contain the N elements of arrays f and g , respectively, first f and then g . All elements are 32-bit integers and N is between 1 and 50000. Note that two elements are separated by one space.

Output Format

Print out the number of pairs that satisfies $f[i] > g[j]$ in one line for each case.

Example

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