# Problem 1. 英文斷詞

## (Time Limit: 2 seconds)

### 問題描述 :

　　斷詞在自然語言的研究上是個很重要的步驟，主要就是將關鍵字從句子中斷出，英文的斷詞較為簡單，就根據句子中的空格將英文字隔開。

### 輸入說明:

　　第一行是一個整數 n(1≦n≦10)代表測試資料有幾筆。

　　每筆測資輸入一句英文敘述句，字元數≤1000。

### 輸出說明:

　　將輸入的句子進行斷詞，將斷出的關鍵字依照句子中的出現排序列印出。全部轉成小寫，列印出的關鍵字不得重複，關鍵字間以一個空格隔開，最後一個關鍵字後面進行換行。例如輸入 How do you do ，則輸出 how do you，最後必須有換行字元。

### 範例:

| Sample Input: | Sample Output: |
|---|---|
| 2<br>How do you do<br>I am so glad to hearing from you | how do you<br>i am so glad to hearing from you |

# Problem 2. 到底養幾隻豬?

## (Time Limit: 2 seconds)

**問題描述:**

　　某天老趙心血來潮拜訪他的好朋友老李,言談間得知老李家裡有養豬,由於現在世界人們愛吃豬,因此豬十分搶手。老趙想知道老李家中有幾隻豬,老李告訴老趙說他也不清楚,但是每天他都會試著數數看,結果是每 7 隻一數多 a 隻,每 11 隻一數少 b 隻,每 17 隻一數多 c 隻,你能幫老趙算出老李實際上「最少」有幾隻豬嗎 ?

**輸入說明:**

　　第一行為一個整數 n(1≦n≦10),代表測試資料有幾筆。

　　每一個測試案例包含三個數字,分別代表題目描述中的 a, b, 及 c。

　　(0≦ a ≦6;0≦ b ≦10;0≦ c ≦16)

**輸出說明:**

　　輸出老李「最少」有幾隻豬。最後必須有換行字元。

**範例:**

| Sample Input: | Sample Output: |
|---|---|
| 2<br>3 3 1<br>2 1 1 | 52<br>681 |

# Problem 3. 敵人來襲

## (Time Limit: 2 seconds)

### 問題描述：

阿奇卡王國是非常富裕的王國，所以其周圍的鄰國都非常覬覦阿奇卡王國的財富。有一天鄰國們終於一起集結並且發動總攻，進攻阿奇卡王國。因為阿奇卡王國的兵力不是很充足，所以國王現在想請你幫他寫一個該如何有效防禦的程式。

### 輸入說明：

輸入的第一個數字 T（$1 \leq T \leq 20$) 代表接下來有 T 筆測試資料。 每筆測試資料會有一個正整數 N 和 S（$1 \leq S \leq 4$），城池的城牆圍成正方形，N 代表一邊城牆的邊長 （$3 \leq N \leq 9$)。

當 S = 1 時，表示敵人從左前方來，所以你的防守兵力佈置要呈現右上到左下佈置。當 S = 2 時，表示敵人從右前方來，所以你的防守兵力佈置要呈現左上到右下佈置。當 S = 3 時，表示敵人從右邊來，所以你的防守兵力布置要聚集到右邊，S = 4 時，表示敵人從左邊來，所以你的防守兵力布置要聚集到左邊。

### 輸出說明：

請輸出防守兵力布置圖，防守的兵士們以 * 代表，不防守的地方則以 x 代表。每個 * 和 x 中間都相隔一個空白 ，最後必須有換行字元。

### 範例：

| Sample Input: | Sample Output: |
|---|---|
| 2 | x x * |
| 3 1 | x * x |
| 4 3 | * x x |
| | x x x * |
| | x x x * |
| | x x x * |
| | x x x * |
| | |

# Problem 4. 三角形的組合

## (Time Limit: 2 seconds)

**問題描述：**

桌上有很多長長短短的繩子，小明想要任選其中的三根將其組成一個三角型，請幫小明列出所有可能的組合方式。

**輸入說明：**

第一行為一個整數 n(1≦n≦10)，代表測試資料有幾筆。

每筆測資的第一行是 m(1≦m≦10)，表示總共有幾根繩子，其後 m 行每一行代表繩子的名字和繩子的長度，並以空白隔開。

**輸出說明：**

可以組合出三角形的繩子組合，每一行代表一種組合，並且每條繩子以空白隔開，最後必須有換行字元。

輸出順序依字母順序 a~z。

**範例：**

| Sample Input: | Sample Output: |
|---|---|
| 2 | a b c |
| 4 | a b d |
| a 8 | b c d |
| b 6 | a b c |
| c 4 | a b d |
| d 3 | a b f |
| 6 | a c d |
| a 8 | a c f |
| b 12 | a d f |
| c 10 | b c d |
| d 6 | b c e |
| e 20 | b c f |
| f 7 | b d f |
| | c d f |
| | |

# Problem 5. PERT Chart

## (Time Limit: 3 seconds)

## Problem Description

The Project Evaluation and Review Technique (PERT) is a model for project management designed to analyze and represent the tasks involved in completing a given project.

PERT chart is presented by a weighted graph. The vertex (called event in PERT) marks the start or completion of one or more tasks and consumes no time and uses no resources. Each edge (called activity in PERT) is the actual performance of a task which consumes time (and requires resources). An event is not "reached" (does not occur) until ALL of the activities leading to that event have been completed. No activity can commence until its immediately preceding event is completed. Accordingly, dependencies among activities can be expressed through the weighted graph for a PERT chart.

The critical path in a PERT chart is the longest possible continuous pathway taken from the initial event to the terminal event. It determines the total calendar time required for the project; and, therefore, any time delay along the critical path will delay the reaching of the terminal event. It is possible to find multiple critical paths if the calendar time required for finishing these paths are the same.

For example, the following sample PERT chart contains 8 vertices and 10 edges. The time required of the critical path is 18.
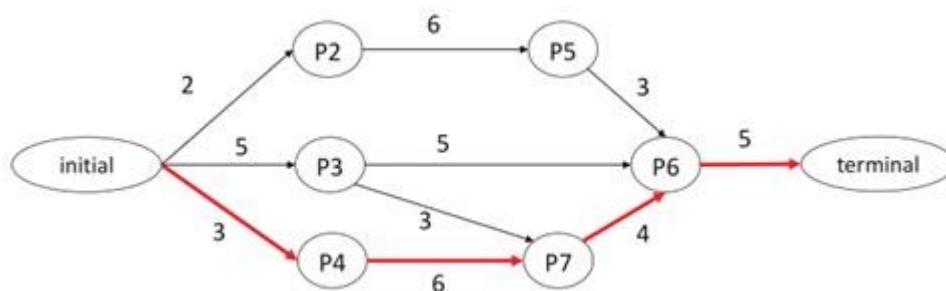


Please write a program to read the input data (a set of given edges in a PERT chart) to construct a weighted graph to represent a PERT chart, find the critical path in the PERT chart, and then output the total time required for the critical path.

## Input Format

Input begins with an integer $T$ $(1 \le T \le 10)$, the number of test cases. Each test case would be in the following format.

For the input data, it starts with the "case" keyword, i.e., a line contains "case". The case keyword is followed by a set of edges.

Each edge includes a source vertex label (String type), a destination vertex label (String type), and weight value (int type). The label of any vertex is a String whose length is less than 5.

These data items are separated by a space character.

An input data ends with the "end" keyword, i.e., a line contains "end".

The number of vertices is less than 100.

## Output Format

The standard output should contain only the required time of the critical path.

Please add a "newline" in the end of the output.

## Example

| Sample Input: | Sample Output: |
|---|---|
| 2 | 18 |
| case | 17 |
| initial P2 2 | |
| initial P3 5 | |
| initial P4 3 | |
| P2 P5 6 | |
| P5 P6 3 | |
| P3 P6 5 | |
| P3 P7 3 | |
| P4 P7 6 | |
| P7 P6 4 | |
| P6 terminal 5 | |
| end | |
| case | |
| initial P1 2 | |
| initial P2 3 | |
| P1 P3 7 | |
| P1 P4 5 | |
| P2 P4 2 | |
| P3 terminal 6 | |
| P4 terminal 10 | |

| end | |
|-----|--|