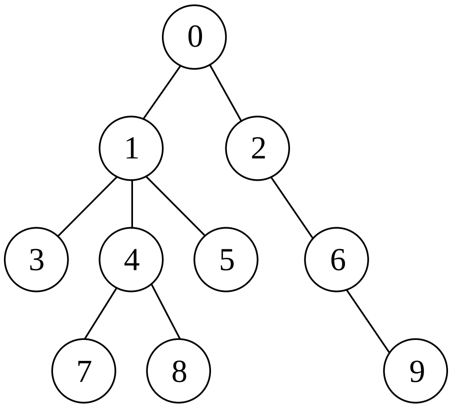
一〇八學年度高級中學資訊學科能力競賽

台中區複賽程式設計試題（二）

7. 最大親等計算 (佔分 10 分)

親等的算法為由自己起算，往上算到與要計算者最接近的共同祖先處，再往下算到那個人，歷經幾代就是幾親等。舉例來說，下圖為某一家族之血緣關係圖，家族之每一成員均以數字代號表示。成員 1 與 2 是成員 0 的小孩；成員 3、4、5 是成員 1 的小孩，依此類推。成員 1 與 9 為 4 親等；成員 0 與 7 為 3 親等。給定某一成員 A，家族中與其關係最遠的成員，他們之間的親等即為 A 之最大親等。在圖中，與成員 1 關係最遠的是成員 9，親等數為 4，因此成員 1 之最大親等為 4；與成員 9 關係最遠的有成員 7 與 8，親等數均為 6，因此成員 9 之最大親等為 6。在本題中，給定某一家族之血緣關係圖及家族之某一成員 A，請找出 A 之最大親等數與該親等之任一家族成員。



輸入說明：

第 1 列有 1 個數字 N，代表家族總人數。

第 2~N 列各有以空白隔開的 2 個數字 P 與 Q，代表 Q 為 P 之小孩。

第(N+1)列有 1 個數字 **A**，代表要找其最大親等之成員。

輸出說明：

依序印出 **A** 之最大親等數與該親等之任一家族成員，中間以空白鍵分隔。

範例輸入：

```
10
0 1
1 3
1 4
1 5
4 7
4 8
0 2
2 6
6 9
1
```

範例輸出：

```
4 9
```

8. 普通分數之四則運算 (佔分 10 分)

一個分數的分子與分母如果都是非 0 整數，我們稱此一分數為普通分數(Common fraction)。寫一個程式可以進行數個正的普通分數的加減乘除四則運算，並將其結果以約分化簡之後的一個普通分數(分子與分母都是整數而且是最小的一組整數)來輸出。

輸入說明：

數個正的普通分數(兩個正整數以/隔開)，兩兩之間以運算子+(加)、-(減)、*(乘)、/(除以)之中的一個來隔開。

輸出說明：

運算結果，一個已約分為最簡的普通分數。注意先乘除後加減的計算順序。若輸入格式錯誤，一律輸出-1。

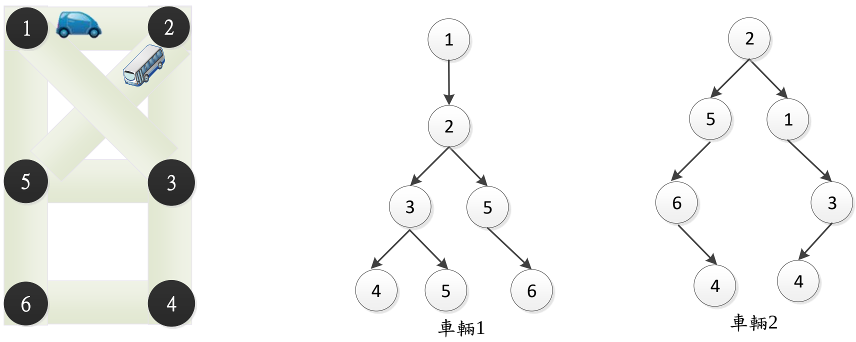
範例輸入：

2/3+7/4+1/12
3/5-7/6*15/6+8/3/2/1
2/3+7/4+1/12/9

範例輸出：

5/2
-59/60
-1

9. 車輛軌跡比對 (佔分 10 分)



(a) 範例地圖　　(b) 車輛 1 與車輛 2 的軌跡樹

為了記錄每台車輛行駛的軌跡，我們建立特殊的樹狀結構（稱為軌跡樹），每個節點記錄每台車在地圖上，如圖(a)，經過的路口 i (1≤i≤100)，樹根為此車輛的啟始路口，每一個節點的子節點是此節點代表的路口的一個相鄰路口。如圖(b)所示，車輛 1 由路口 1 出發，行經路段(1, 2)，到達路口 2。經過路口 2 後有兩種走法，一是行經路段(2, 3)走到路口 3，另一種則是行經路段(2, 5)走到路口 5。

現在為了要找出這些車輛是否經過共同的路口，需要寫程式進行車輛之間軌跡樹的比對，來求得最長的連續路口為何，符合規定條件的共同且連續的路口數必須大於或等於 2。在此範例中，答案為由路口 2，經過路口 5，到達路口 6。

輸入說明：

第一列代表測試資料有幾組。
接下來各組資料的首列為一個正整數 m，表示所有的路口數；次列有兩個正整數 n 與 p，n 是車輛 1 所有軌跡的數目，而 p 則是車輛 1 的啟始路口(樹根節點)。後面的 n 列的每一列有多個正整數 v ($1 \leq v \leq m$)，代表車輛由啟始路口 p，行經的每個路口 v 的軌跡。接著再列出車輛 2 資訊。

輸出說明：

依序列出各組兩車之間最長的共同且連續路口（符合條件的路口數必須大於或等於 2），兩路口之間用空格分隔。假如找不到符合上述條件的軌跡，則輸出**-1**，若有多個最長連續路口，則任一路徑均是本題答案。

範例輸入：

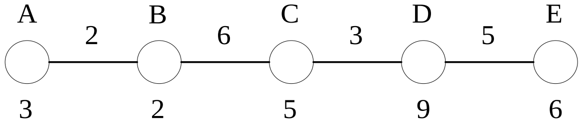
```
2
6
3 1
1 2 3 4
1 2 3 5
1 2 5 6
2 2
2 5 6 4
2 1 3 4
6
2 1
1 5 6 4
1 2 5 6
1 2
2 3 4 6
```

範例輸出：

```
2 5 6
-1
```

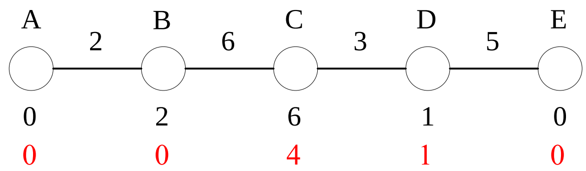
10. **交通號誌設定** (佔分 10 分)

在市區的交通網路中，於交叉路口設置交通號誌（紅綠燈）是不可避免的事。在相鄰兩個路口，綠燈亮的時間若互相配合得當則行車順暢；反之在一個路口通過了綠燈到下一個路口卻是紅燈則需要等待。考慮一條直路上的交通號誌時間設定問題，如下圖：



現在，有一位駕駛人於時間 0 要從 A 出發，最後的目的點是 E。路段上的數字表示走完該路段所需之秒數，例如 A 到 B 需 2 秒，B 到 C 需 6 秒。假設交通號誌只有紅燈與綠燈，其週期為 T 秒，即每個路口紅燈亮 T/2 秒，綠燈亮 T/2 秒，持續循環。上圖中，各點的數字為該點**綠燈**開始亮的時間，其介於 0 與 T-1 之間。假設 T＝10，到達 E 點之時間可如下計算：

A 點需等 3 秒(因為 0 出發，但 3 才變為綠燈)；到 B 點為時間 5，而 B 點此時為綠燈(2≦t<7 為綠燈，7≦t<12 為紅燈)，故不需等待。到 C 點為時間 11，而 C 點之 5≦t<10 及 15≦t<20 為綠燈，故需等待 4 秒，當時間 15 才能再從 C 出發；到 D 點為時間 18，而 D 點綠燈時間為 9≦t<14 及 19≦t<24，故需等待 1 秒。D 至 E 之長度為 5，故於時間 24 抵達 E(因已到達 E，故不必計算 E 之等待時間)。所以駕駛人於時間 0 從 A 出發，在時間 24 會抵達 E 點。如果有另一位駕駛人於時間 0 從 E 走向 A，其抵達 A 的時間亦可算出。注意：如果某一點目前為綠燈，則雙向均可通行，若為紅燈，則雙向均不可通行。如果將綠燈亮的時間加以變更，如以下 2 組的設定：



我們可以算出從 A 到 E 的總時間為 16，從 E 到 A 的總時間亦為 16。此設定亦是最好的設定，因為途中均無需等待紅燈。本題即是要找出設定綠燈亮的時間，使得兩個方向從時間 0 出發(第一點為起點)，於抵達目的點(最後一點)所需要的時間相同或者最多相差 1。在符合條件的眾多設定中，找出兩者所花時間合計為最小者。若有多組設定方式，則設定之時間加總最小者為本題答案。(如 00410)

輸入說明：

測試資料有三列，第一列為路段個數 n，n<15；第二列有一個非負整數，為週期 T(一定為正偶數，0<T<10000)；第三列為 n 個路段之長度(均為整數，且小於 10000，以空白鍵分隔)。

輸出說明：

印出兩列資料，第一列為兩個方向所需時間的總和，第二列印出 n+1 個點(含首尾兩點)**綠燈開始亮的時間**(均介於 0 與 T-1 之間)，各時間以空白鍵分隔。

範例 1 輸入：

4
10
2 6 3 5

範例 1 輸出：

32
0 0 4 1 0

範例 2 輸入

2
1000
700 200

範例 2 輸出：

1801
0 201 0