

Graph Theory - S009 (E045)Solved Challenges **0/1**[Back To Challenges List](#)**Photograph Height****ID:11157 Solved By 401 Users**

There are **N** students in a high school class. For simplicity, the students are named **A, B, C, ...** (up to the **Nth** letter of the alphabet). No two students have exactly the same height.

The Principal of the school wants to organize a class photograph where the students are arranged in ascending order of heights. The class teacher has been asked to list the students in ascending order of heights to send to the photographer.

The teacher does not know the heights of the various students. However, there are a set of **K** photographs, which between them have all the students in that class. From each of the photographs, the teacher obtained a list of students in ascending height order and needs to combine them into one list of students in ascending height order.

It is possible that the photographs do not fully determine the ordering in the list. In this case, as it is vacation time, the teacher needs to write to some of the students to send her their exact height, so that she can create the complete list.

The objective is to write a program that can list the students to whom she must write to determine their heights. If it is not necessary to write to any student, the output should be **N/A**

Boundary Condition(s): $2 \leq N \leq 26$ $1 \leq K \leq 10$ **Input Format:**

The first line contains N and K separated by a comma.

The next K lines, each contains a comma separated list of the students in ascending order from the corresponding photograph.

Output Format:

The first line contains the list of students whose height must be known to create the ordered list separated by space or N/A.

Example Input/Output 1:

Input:

8,3

D,C,E,F,G,H

C,A,E

D,C,B,E

Output:

A B

Explanation:

Here N is **8** and K is **3**. The students are **A, B, C, D, E, F, G** and **H** (the first N letters of the alphabet).

From photographs 1, 2 and 3, we can determine that A, B, C and D are shorter than E, F, G, H, and that **E, F, G, H** are the last four in the ordered list.

From the first photograph, D is shorter than C, and from the second photograph, C is shorter than A. From the third photograph, C is shorter than B. The possible orders of the first four could be **D, C, A, B** or **D, C, B, A**. No information is present in any of the photographs which says which order is correct. Hence the teacher must write to both B and A to determine their heights so as to establish the correct order. The output must be sorted in alphabetic order, and hence the output is **A B**.

Example Input/Output 2:

Input:

8,3

D,C,E,F,G,H

C,A,B,E

D,B

Output:

N/A

Explanation:

Here N is **8** and K is **3**, and so there are 8 students and 3 photographs. As the students are named the first N letters of the alphabet, the students are **A, B, C, D, E, F, G** and **H**.

The first photograph shows the ordering as D, C, E, F, G, H, that is D is shorter than C who is shorter than E and so on. The second photograph shows that C is shorter than A, who is shorter than B who is shorter than E. The third photograph shows that D is shorter than B.

From photograph 1, E, F, G and H are all taller than C, and are in that order. From photograph 2, A and B are shorter than E and in that order. Hence A, B, C and D are all shorter than E, F, G and H. Hence the last four in the final ordered list must be **E, F, G, H** in that order.

From the first photograph, D is shorter than C, and from the second photograph, C is shorter than A who is shorter than B. Hence the order of the first four is **D, C, A, B**.

Hence the full list is **D, C, A, B, E, F, G, H**. The full order can be determined without writing to any student to get their height. Hence the output is **N/A**.

Max Execution Time Limit: 500 millisecs

Ambiance

Python3 (3.x)



Reset

```
1 import collections
2
3 N,K = map(int,input().strip().split(","))
4
5 string = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
6 names = string[0:N]
```

```
7 photos = []
8 for k in range(K):
9     photos.append(''.join(input().strip().split(",")))
10 print(photos)
11
12 missing = False
13 for name in names:
14     relCount = 0
15     queue = collections.deque()
16     visited = []
17     queue.append(name)
18     visited.append(name)
19     while(len(queue)!=0):
20         student = queue.popleft()
21         for seq in photos:
22             if(student in seq):
23                 succ = seq[seq.index(student):]
24                 for succStud in succ:
25                     if(succStud not in visited):
26                         queue.append(succStud)
27                         visited.append(succStud)
28                         relCount+=1
29
30
31
32     while(len(queue)!=0):
33         student = queue.popleft()
34         for seq in photos:
35             if(student in seq):
36                 pred = seq[:seq.index(student)]
37                 for predStud in pred:
38                     if(predStud not in visited):
39                         queue.append(predStud)
40                         visited.append(predStud)
41                         relCount+=1
42
43
44     if(relCount!=N-1):
45         print(name,end=" ")
46         missing = True
47
48
49 if(missing==False):
50     print("N/A")
51
52
53
54
55
56
57
58
59
60
61
62
```

Code did not pass the execution



Input:

Expected Output:

Your Program Output:

Save

Run

- ☐ Run with a custom test case (Input/Output)