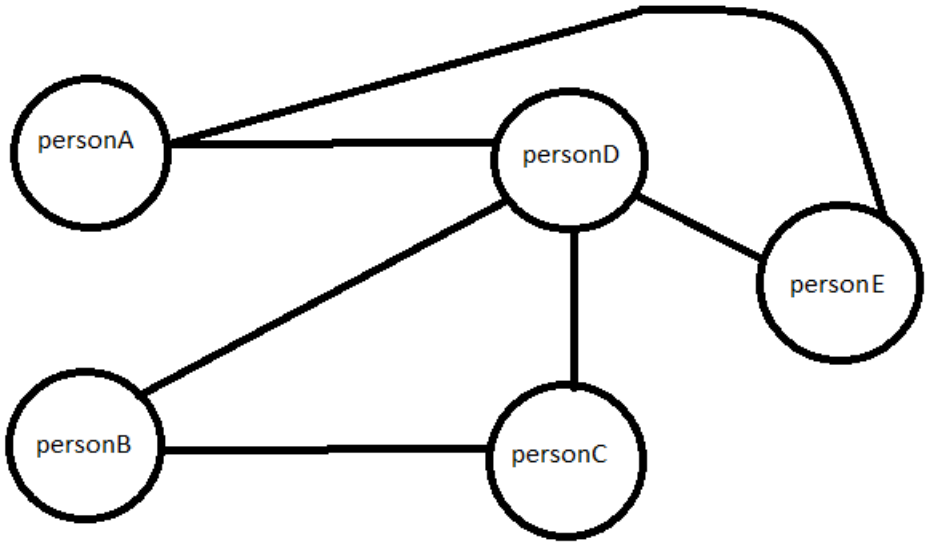
FRIEND SUGGESTER

This is a simple c program to for suggesting friends that may know each other through mutual friends. This program uses the concept of arrays and matrix to solve this problem. Feel free to modify the code.

Author : HARSHVARDHAN PATIL

Email : 007harsh007patil@gmail.com

Example:



In the above case the circle represents the name of the person and the lines represents the friendship from one person to another. This is not easy to take as a input in c.

To simplify this this above diagram can be represented as:

	personA	personB	personC	personD	personE
personA	0	0	0	1	1
personB	0	0	1	1	0
personC	0	1	0	1	0
personD	1	1	1	0	1
personE	1	0	0	1	0

0 is given in the matrix if there is no friendship and 1 is given if there is friendship.

This can be represented as an 2D array.

Then the algorithm in the code is applied for suggesting the friends.

Output:

```
Enter the no of entries:5
Enter the 1th entry
Enter the name:personA
Enter the no of friends:2
Enter the name of friend:personD
Enter the name of friend:personE
Enter the 2th entry
Enter the name:personB
Enter the no of friends:2
Enter the name of friend:personC
Enter the name of friend:personD
Enter the 3th entry
Enter the name:personC
Enter the no of friends:2
Enter the name of friend:personB
Enter the name of friend:personD
Enter the 4th entry
Enter the name:personD
Enter the no of friends:4
Enter the name of friend:personA
Enter the name of friend:personE
Enter the name of friend:personB
Enter the name of friend:personC
Enter the 5th entry
Enter the name:personE
Enter the no of friends:2
Enter the name of friend:personA
Enter the name of friend:personD
0       0       0       1       1
0       0       1       1       0
0       1       0       1       0
1       1       1       0       1
1       0       0       1       0

personA can know personB through personD

personA can know personC through personD

personB can know personA through personD

personB can know personE through personD

personC can know personA through personD

personC can know personE through personD

personE can know personB through personD

personE can know personC through personD
```