

## ***Homework 2***

*100 Points*

### ***2D Arrays***

**Project: Friends** (see next pages)

**Honors Projects:**

**Friends** (next pages)

**Celebrity** (next pages)

### **Grading**

	<b>Basic</b>	<b>Honors</b>
1. Declare arrays (good style)	– 5Points	– 5
2. main()	– 15	– 10
3. Get data from file	– 20	– 15
4. Write table	– 20	– 10
5. Lists of friends	– 20	– 20
6. Groups of two	– 20	– 20
7. Celebrity		– 20

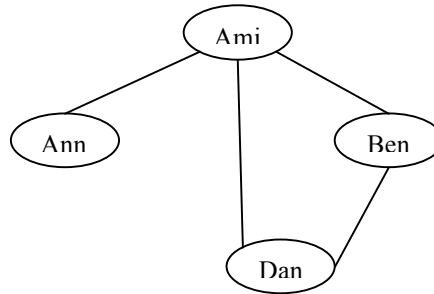
**UPLOAD** the source file(s), screen output included, and the input file.

CIS 22B  
Intermediate Programming Methodologies in C++  
Programming Assignments

## Project: Friends

Provide information about students belonging to a club. The club has up to **80** students. All students belonging to this club must have a **3** letter first name or nickname. The data file, **3club.txt**, begins with **n**, the number of students belonging to the club, followed by the names of the people, in alphabetical order. On the next n lines, for each student in the club, there is a list of 0s and 1s showing his friends: 1 – friends, 0 – not friends.

```
4
Ami
Ann
Ben
Dan
0 1 1 1
1 0 0 0
1 0 0 1
1 0 1 0
```



For instance the first line means: Ami has three friends: Ann, Ben, and Dan (in columns 1, 2, and 3 is 1; in column 0 is 0, because Ami does not count as her own friend). The second line means: Ann has only one friend: Ami.

Read the list of names into an array of strings. Read the remaining lines into a table.

Output should consist of the following:

1. The original table of 0s and 1s formatted as shown below:

	Ami	Ann	Ben	Dan
Ami	0	1	1	1
Ann	1	0	0	0
Ben	1	0	0	1
Dan	1	0	1	0

2. For each student in the club list its friends; include the number of friends too.

```
Ami (3): Ann, Ben, Dan
Ann (1): Amy
Ben (2): Amy, Dan
Dan (2): Amy, Ben
```

3. List all groups of two friends.

```
Ami - Ann
Ami - Ben
Ami - Dan
Ben - Dan
```

CIS 22B  
**Intermediate Programming Methodologies in C++**  
Programming Assignments

**Advanced Project: Celebrity** // see [22B\\_Hw\\_2\\_Honors.cpp](#)

Among  $n$  persons, a celebrity is defined as someone who is known by everyone but does not know anyone. The problem is to identify the celebrity, if one exists.

Your task is to write two functions and call them from `main()`.

- The first function's task is to display the table and the names: it has two parameters, the relationship table, and  $n$ , the number of persons, and no return value.
- The second function's task is to identify the celebrity: it has two parameters, the relationship table, and  $n$ , the number of persons. The value returned from the function is -1 if there is no celebrity, or the index that identifies the celebrity.