1. A basketball club is going to recruit five new members. The broker wrote the following program. In the object array newMembers, the character string (e.g., "Zhang San") is the name, and the integer (e.g., 22) is the age. However, the broker forgot the name and age of one member. By default, he expect that the name is "???" and the age is 0. Please write a class named "Member" in header file "member.h" along with "member.cpp" to make the program compile and run correctly as expected. The member variables "name" and "age" of class Member should be defined as "const" type. And the member function "printInfo" should also be defined as "const" type (i.e., the function cannot modify the members of the class). Note that the file main.cpp cannot be modified.

```
//main.cpp
#include "member.h"

int main()
{
   Member newMembers[5] = { Member("Zhang San", 22), Member("Li Si", 19),
Member("Wang Wu", 18), Member("Zhao Liu", 24) };
   for (int i=0; i<5; i++)
      newMembers[i].printInfo();
   return 0;
}
```

Tips: You are suggested to learn to use STL string for solving this problem: http://www.cplusplus.com/reference/string/string/string/

2. The Matrix: Come on! We are not talking about the movie, though the Virtual Reality[1] in the movie is fully aided by the matrix computations and transformations on the computer. You may know more about this in the "Computer Graphics" class in the future. But get your hands dirty now by implementing the simple matrix computations first! Given two input files "a.txt" and "b.txt" saving matrices a and b, basically what you need to do is as follows: (1) define a class named "Matrix", (2) load in file a.txt and construct Matrix object a, (3) load in file b.txt and construct Matrix object

[1] In fact, Virtual Reality is exploiting software and hardware of the computer to generate a simulation of an environment, such as an easeful classroom, a breathtaking action and so on.

b, (4) perform 'Matrix c = a.multiply(b);' to compute the product of a and b, (5) call 'c.save_product("aXb.txt");' to save the product result into a file named "aXb.txt". Look at the following user program "main.cpp", write other source files, e.g., "Matrix.h", "Matrix.cpp", "makefile", such that the program can compile and run as expected with the following inputs.

```
//main.cpp
#include "Matrix.h"
//…

int main()
{
   Matrix a("a.txt"), b; //a.txt stores a matrix of 6 rows and 8 columns
   b.load("b.txt"); //b.txt stores a matrix of 8 rows and 6 columns
   a.display(); //print matrix a
   b.display(); //print matrix b
   Matrix c = a.multiply(b); //compute the product
   c.display_product(); //print the product
   c.save_product("aXb.txt"); //get a matrix of 6 rows and 6 columns
   return 0;
}
```

a.txt

```
6 8
2 21 5 6 23 56 78 32
45 10 6 9 32 21 43 22
21 20 32 22 12 3 76 23
65 20 5 56 56 56 56 54
24 3 23 2 76 56 21 32
98 24 56 13 21 68 54 12
```

b.txt

```
8 6
45 76 23 256 78 32
6 4576 23 64 43 642
345 21 12 2465 76 267
5 2345 56 2311 3221 54
23 567 76 56 21 5632
687 876 24 786 2 12
32 234 852 68 2345 763
876 265 756 246 54 6884
```

Tips:

1. You are suggested to learn to use ifstream/ofstream for loading and saving files: http://www.cplusplus.com/reference/fstream/

2. Matrix multiplication:

$$\begin{bmatrix} 4 & 5 \\ 2 & 7 \\ 1 & 9 \end{bmatrix} \times \begin{bmatrix} 3 & 7 & 1 \\ 3 & 6 & 9 \end{bmatrix} = \begin{bmatrix} 4\times3+5\times3 & 4\times7+5\times6 & 4\times1+5\times9 \\ 2\times3+7\times3 & 2\times7+7\times6 & 2\times1+7\times9 \\ 1\times3+9\times3 & 1\times7+9\times6 & 1\times1+9\times9 \end{bmatrix} = \begin{bmatrix} 27 & 58 & 49 \\ 27 & 56 & 65 \\ 30 & 61 & 82 \end{bmatrix}$$