**Gitlin's Match Game**

For this project you will be implementing a match game.  
**Pairs of symbols will be randomly placed on a rectangular grid.**  
The object of the game is to find all the pairs within a time-limit based upon the number of pairs.  
  
**In your implementation of the game you must use a vector of vectors.**  
  
In the beginning of the game, the user specifies the dimension of the board. The attached PDF document [DirectionsPDF.pdf](https://bbhosted.cuny.edu/bbcswebdav/pid-33154352-dt-content-rid-138631650_1/xid-138631650_1)  shows all messages and the format of the output that you must reproduce. And, shows how the game progresses. In the document, the screen shots were done at a MS Command Prompt, and you see white letters on a black background. In Eclipse, the colors will be reversed.   
  
Once the board is established the user will see numbers 1,2,3 ... displayed in a rectangular layout determined by his/her specifications for rows and columns. If the rows entered were 5 and the number of columns was 4, ten pairs of symbols would be hidden behind labels of 1 through 20. Then the user will ask to see the symbols behind two numbers.  After a short period of time the screen is cleared. If the two symbols matched, they will stay in view for the rest of game. If they did not match, the numbers "covering" them will be seen.  
  
**In your implementation you could use two vectors of vectors.**  
One is to store the original hidden pairs of symbols,  
that always stays hidden.  
The other is used to display the board to the game player. Originally this second vector of vectors, will only contain the numeric labels. But as the game  progresses and user finds pairs that match, the numbers will be replaced by the underlying symbols.  
  
You must implement one function, that can be used to display the "hidden" board and the  "visible" board/  
Please name this function printBoard and implement according to the following prototype:  
    void printBoard( const vector< vector Y> >& board);  
where Y is a data-type of your own choice [ it is **not** a template function], but I will give you some suggestions.

As I wrote above: ***In your implementation you could use two vectors of vectors.***If you choose a different manner to implement the game, you can **add** parameters to printBoard **but** the data-type of the first parameter must be as shown..  
  
**Below I will talk about a board with 20 slots,**  
**but your configured board could have more or less slots.**  
  
Hints:  
1. For a board with 10 hidden pairs, the user will be picking slot numbers  from 1 to 20. With a little thought you should be able to map the slot number to a row and column. Integer arithmetic, namely division and use of the % operator will be quite helpful.  
  
2. Above I mentioned <vector Y> saying Y could be your own data-type. I was able to implement the game having Y be either string or char. You choice of Y could be different.  
  
Regardless, on the board that is visible to user,  
I needed to differentiate between matched symbol and numeric slot still hiding a symbol.  
When implementing the game using "string", I used the fact that string values "1", "2", ..., "20"   
are less than "A", "B", ..., "Z" according to the ASCII collating sequence.  
  
When I implemented the game usng "char", I used the fact that char values (small integers) 1, 2, ...20 come before character values 'A', 'B', ..., 'Z'.  
Though in order to print a char value used as a numeric slot number, a cast to an int was needed:  
    char c = 7;  
    cout << (int) c;  //prints a seven  
while   
    cout << c;  //outputs an alert (rings the bell)

And, in order to make it easier to grade your code, please have the result of multiplying the rows and columns in a variable named **product**:

      int product;

**For this and all assignments to follow name the file that you submit should  start with the first letter of your first name, followed by your last name, followed by the assignment number and a ".cpp" suffix. If I was doing this one my file would be named JGitlin5.cpp**