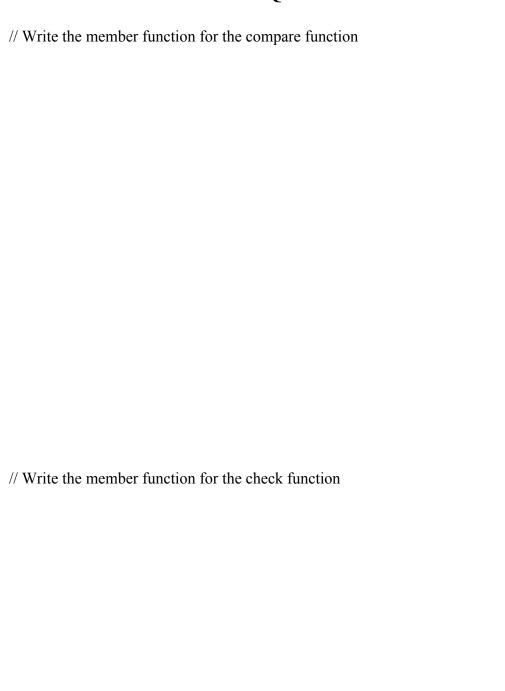
QUESTIONS ON CLASSES:

};

In this problem, you will implement the following Date class. This problem tests your understanding of declaring and using objects. Use const where appropriate.

```
date.h:
#pragma once
class Date {
 public:
     // WRITE THE DEFAULT CONSTRUCTOR
     // WRITE ANOTHER CONSTRUCTOR (WITH ALL DATA MEMBERS)
     // WRITE ONE SETTER (WITH ALL DATA MEMBERS)
     // WRITE THREE GETTERS (ONE PER DATA MEMBER)
     // WRITE AN Equal FUNCTION THAT RETURNS TRUE
           IF CALLER DATE IS EQUAL TO THE DATE PARAMETER
     // WRITE A Compare FUNCTION THAT RETURNS TRUE
           IF CALLER DATE IS EARLIER THAN DATE PARAMETER
 private:
     int day, month, year;
     // WRITE A Check FUNCTION THAT RETURNS TRUE
           IF VALUES OF ALL DATA MEMBERS ARE VALID
```

date.cpp:
#include "date.h"
// Write the default constructor, which sets month to 1, day to 1, year to 2000
// Write the 3-parameter constructor; use Check() and set values to default if needed
// Write one of the accessor functions
// Write the member function for the equal function



Date-Program.cpp:

```
#include "date.h"
#include <iostream>
using namespace std;
int main() {
       // declare a Date object named today and set to the current date
       // (e.g., month 12, day 9, year 2015) using the 3-parameter constructor
       // declare a Date object named dueDate using the default constructor
       // use the setter function to set dueDate to 12/15/2015
       // use the Compare function, and print a message that indicates whether the
               due date is past OR the due date is not past.
       // declare a vector of 31 Date objects named dec (for December)
       // populate dec with the 31 days in December 2015
       // use the accessors to display dec[24]'s date, formatted as mm/dd/yyyy, using at()
       // use the Equal function and print EXAM!! if dec[14] is equal to parameter today
      // have today be the parameter and use the vector at function
```

QUESTIONS ON ARRAYS:

}

```
#include <iostream>
using namespace std;
const int NUM ROWS = 2;
const int NUM COLS = 3;
void SumCols(int nums[NUM ROWS][NUM COLS], int tots[], int rows, int cols);
void PrintArray(int nums[NUM COLS], int count);
int main() {
       // Declare a 2D array of integers named numbers with NUM ROWS rows
      // and NUM_COLS cols
       // Declare a 1D array of integers named totals with size NUM COLS
       // Create a loop to initialize all the values in totals to 0
       // Create a nested loop to initialize all the values in numbers to row # * column #
       // For example, numbers[0][0] will be 0, numbers [2][2] will be 4
       // Create a function call to SumCols, pass in the array numbers, the array totals,
               NUM ROWS and NUM COLS as the number of rows and columns
       // Create a function call to PrintArray to print the values in totals
```

$ \begin{tabular}{ll} void SumCols(int nums[NUM_ROWS][NUM_COLS], int tots[\], int rows, int cols) { \\ for (int c=0; c < NUM_COLS; ++c) \\ for (int r=0; r < NUM_ROWS; ++r) \\ tots[c] += nums[r][c]; \\ \end{tabular} $
<pre>void PrintArray(int nums[NUM_COLS], int count) { for (int i=0; i<count; ++i)="" <<="" cout="" endl;="" nums[i]="" pre="" }<=""></count;></pre>
Write the contents of the array numbers.
Write the contents of the array totals.
In SumCols, why don't you pass the array tots by reference??