Assignment 1: “Hello World!”

Type in, compile, link, and run the following C program:

```c
#include <stdio.h>

int main ()
{
    printf ("Hello World!\n");
    return 0;
}
```

Type in, compile, link, and run the following C++ program:

```cpp
#include <iostream>

int main ()
{
    std::cout << "Hello World!" << std::endl;
    return 0;
}
```

Answer the following questions:

1. How could each of the above programs be modified to generate a preprocessor error?
2. How could each of the above programs be modified to generate a compiler error?
3. How could each of the above programs be modified to generate a compiler warning?
4. How could each of the above programs be modified to generate a link error?
Assignment 2: Taylor series for sine and cosine

Part a

The Taylor series for $\sin x$ and $\cos x$ are given by

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$$

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots$$

Write a C program that calculates the sine and cosine of a number supplied by the user using the above series. Do not use the math library for this problem.

You should design your program so that it is accurate to 5 decimal places. For an alternating series, the magnitude of the error is less than the magnitude of the first neglected term. For example, if we use the approximation given by

$$\cos x \approx f(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!}$$

then the error is bounded as follows:

$$|\cos x - f(x)| < \frac{x^8}{8!}$$

So, for accuracy to 5 decimal places, we require that

$$\frac{|x^n|}{n!} < 0.000005$$

Thoroughly test your program using your calculator. Note that your program will only work for numbers with magnitude less than about 5.0; this is normal.

Here is a sample output:

```
Enter a floating point number: 0.9
Sine of 0.900000 = 0.783327
Cosine of 0.900000 = 0.621610
```

Part b

Use your program to calculate the sine and cosine of 30. Explain why the method does not produce the expected results. It may help to temporarily insert some statements in your program that print out intermediate results.

Part c

The sine and cosine of a number should satisfy the following relationship:

$$\sin^2 x + \cos^2 x = 1$$

Have your program use this relationship as a sanity check on its results, and print an error message when the results are not reasonable.
Assignment 3: Binary search using recursion

Write a recursive function in C to perform a binary search on an ordered list. Your function should:

- Split the list in half and search only the appropriate half for the target element.
- Terminate without recursion only when the list contains just one element.
- Return a true/false condition to the calling program to indicate whether or not the target element is found.

Write a main program to test your binary search function. You may test your program using a hard coded list of integers.

Find a formula for the maximum number of times the recursive function will call itself.

Assignment 4: Root finding using bisection

Write a C function to find the root of a mathematical function, $f(x)$, using the bisection method. Your C function should accept as arguments an interval containing a root of $f(x)$, and a pointer to a C function implementing $f(x)$. Write a main program to test your root finding function using $f(x) = (\ln x) - 1$.

Make your root finding function as “bullet proof” as possible. In other words, design your root finding function so that it will never fail without returning an error flag to the calling routine. Explain why you are convinced that your code is “bullet proof.”

Assignment 5: Doubly linked lists

Write a module in C to hold a doubly linked list of character strings. Your complete program will contain the three files in the inclusion diagram shown below:

```
book.c
linklist.c
book.h
```

You will write the source files linklist.c and book.c; the header file book.h is given below:

```c
#ifndef BOOK_H_INCLUDED
#define BOOK_H_INCLUDED

typedef struct PageStruct * Page;
#endif
```

3
typedef struct BookStruct * Book;

Page pageCreate (const char * str);
Book bookCreate ();
void bookAppendPageFront (Page page, Book book);
void bookAppendPageBack (Page page, Book book);
void bookAppendFront (const char * str, Book book);
void bookAppendBack (const char * str, Book book);
Page bookRemovePageFront (Book book);
Page bookRemovePageBack (Book book);
const char * bookRemoveFront (Book book);
const char * bookRemoveBack (Book book);
void bookPrint (Book book);

#endif

The file linklist.c should contain only the main function: all other functions and structures will be in book.c. Write a main function to thoroughly test the functions bookCreate, bookAppendFront, bookAppendBack, bookRemoveFront, bookRemoveBack, and bookPrint. The functions that are not tested in main will all be called by the functions that are tested in main. Be sure to test all cases in which the list is empty.

Assignment 6: Time of day object

Write a C++ class to store and manipulate time of day in hours, minutes, and seconds. Provide the following capabilities:

- Client access to the data should be restricted so that the data can never hold invalid values.
- Provide functions to increment or decrement the time of day by one second.
- Provide functions to increment or decrement the time of day by an arbitrary number of seconds.
- Provide a function to print the time of day in a readable format.

Write a main function to test all the above capabilities. For this assignment, you may put everything in one source file.

Assignment 7: Doubly linked lists redux

Rewrite assignment 5 in C++. 
Assignment 8: Complex root finding fractal

The polynomial \( f(z) = z^4 - 1 \) has four roots given by \( \pm 1 \) and \( \pm i \). These roots can be found using Newton’s method:

\[
z_{k+1} = z_k - \frac{f(z_k)}{f'(z_k)}
\]

The algorithm will converge to one of the roots of \( f(z) \); which root it converges to depends upon where in the complex plane the starting point, \( z_0 \), is located. Predicting which root a particular \( z_0 \) will converge to is not a simple task. In fact, the set of all starting points that converge to a particular root is a fractal set.

The code given below calculates a fractal image using Newton’s method and the polynomial \( f(z) = z^4 - 1 \):

```cpp
#include "QComplex.h"
#include "QImage.h"

int main ()
{
    double lo = -1.2, hi = 1.2, incr = 0.0025;
    int image_size = int ((hi - lo) / incr);
    QImage image (image_size, image_size);
    for (int x = 0; x < image_size; x++)
        for (int y = 0; y < image_size; y++)
        {
            QComplex<double> z (lo + x * incr, lo + y * incr);
            if (z.abs() > 1e-10)
                for (int n = 0; n < 30; n++)
                {
                    QComplex<double> z_cubed = z*z*z;
                    z -= (z*z_cubed - 1.0) / (z_cubed*4.0);
                }
            if (fabs(z.r) > fabs(z.i))
                image(x,y) = 255;
            else
                image(x,y) = 0;
        }
    image.tiffWrite ("fractal.tiff");
    return 0;
}
```

The fractal image created by the code above is shown below:
Your task is to create the C++ classes `QComplex` and `QImage` so that the code above works. You must provide all the necessary member functions without changing the given code. A module to help you write TIFF image files will be provided.

**Assignment 9: Digital filter simulation**

A digital Butterworth filter design produces the following transfer function:

\[ H(z) = \frac{0.004824(1 + z^{-1})^4}{1 - 2.3695z^{-1} + 2.31399z^{-2} - 1.05467z^{-3} + 0.18738z^{-4}} = \frac{0.0695(1 + z^{-1})^2}{1 - 1.3209z^{-1} + 0.06327z^{-2}} \cdot \frac{0.695(1 + z^{-1})^2}{1 - 1.0486z^{-1} + 0.2961z^{-2}} \]

Which leads to the following implementation:

\[
x[n + 1] = Ax[n] + Bu[n]
\]
\[
y[n] = Cx[n] + Du[n]
\]

Your assignment is to simulate the above system using state space methods. The system can be put in the form

\[
x[n + 1] = Ax[n] + Bu[n]
\]
\[
y[n] = Cx[n] + Du[n]
\]

where

\[
A = \begin{bmatrix}
1.3209 & -0.63274 & 0 & 0 \\
0.23066 & 0.025509 & 1.0486 & -0.29614 \\
0 & 0 & 1 & 0
\end{bmatrix}
\]
\[ B = \begin{bmatrix} 1 \\ 0 \\ 0.069457 \\ 0 \end{bmatrix} \]

\[ C = \begin{bmatrix} 0.016021 & 0.0017718 & 0.21175 & 0.048888 \end{bmatrix} \]

\[ D = [0.0048243] \]

You should create separate objects for a column vector, row vector, and matrix. All three classes should store data in a buffer that is managed by a common base class. The inheritance scheme is shown below.

```
DataBuffer
   / \       / \       / \\
Matrix       ColVector    RowVector
```