In this project you will be implementing and simulating a very simple computer network system. You will be using C++ object oriented programming, and you have to use objects. The computer network is shown in the following figure:

![Diagram of computer network system]

The first object is computer. In each computer there is a buffer for incoming message. It has a size of 500. There is also a counter to count the number of message sent. It has two main functions, send out messages and receive messages. There are a total of 6 computers, c1 through c6.

1. Each message has to be processed before they are sent so there is a delay. The message they send will be the number of the destination. Sure enough the destination can’t be the sender itself. For example if c1 wants to send a message to c5, then after 3 time units (a random delay) a message “5” is sent.

2. When a computer sense that there is incoming message, it will take that message and put it in the incoming buffer without any delay.

The second object is router. There are two routers, r1 and r2. Computer c1 through c3 are connected to r1, and c4 through c6 are connected to r2. Each router has a buffer of size 300. Each of them has two functions, send out message and receive message.

1. The router can only send out a message in its buffer after a random delay. It looks at the message and determines if the message belongs to the local network or the network on the other side. If it belongs to local network, it is sent to the destination computer directly. If it belongs to the other network, it is sent to the other router. For example if r1 has a message for c6, it will send out a message “6” and r2 will receive it. Then the message is deleted from r1’s buffer.
2. When a router sense that a message is coming message, it will receive that message and put it at the end of its buffer without any delay, and return 1. If the buffer is full and a message comes in, the return signal would be 0.

There is a file with all the backbone written for you already (network_blank.cxx). You have to fill out the blanked part using what you have learned in lecture about classes in C++. Basically you have to do the following:

For example in the computer class, implement the buffer, \textit{inmsg}. Implement functions like \texttt{send()}, \texttt{step()} and \texttt{receive()}. In \texttt{send()}, it will generate a random time delay and also a random destination. It will then put that message in the register (variable) \texttt{outmsg}. Also implement a function \texttt{step()} that will decrement the delay by 1 (time unit). For \texttt{receive()}, it will put the incoming message in \texttt{inmsg} immediately (therefore all the messages in c1’s \texttt{inmsg} should be all 1’s, etc). Implement everything in such a way that the user can only interact with this class through the public functions (hint: what should be in public and what should be in private?)

For the router class, implement the buffer, \textit{queue}. Implement function \texttt{send()} that generates a random delay, for which the message at the top of the buffer will be sent when delay becomes zero. There is also a function called \texttt{step()} doing the exact same job as the one in computer object. When there is an incoming message for a router, you will use the following syntax to put that data at the end of the queue. For example if r1 gets a message for c5:

\begin{center}
\texttt{variable} = r1 + 5; // \texttt{variable} = 0 if success, 1 if failed
\end{center}

Again, implement everything so that user can only interact with the object through the public functions (by definition an operator is also a function).

Part of the \texttt{main()} is done for you already. It basically has a global clock simulator. During each clock cycle, it will determine if each computer or router has message to send or generate. For computer object if the delay becomes zero, that message will be sent to a router, and a new message is generated. For router the message at the top of the buffer would be sent and either put in a computer’s buffer or the other router’s buffer. Then it will decrement the delay in each computer or router. Please read the \texttt{main()} carefully to determine how the functions in each class should be implemented, otherwise you won’t be able to get it work. Only part of the \texttt{main()} is done, you have to finish the remaining.

Each computer and router will have its own seed to generate delay. They are as follows:

\begin{center}
\begin{tabular}{|c|c|c|c|c|}
\hline
C1 & 5 & C4 & 7 & R1 & 3 \\
\hline
C2 & 6 & C5 & 3 & R2 & 7 \\
\hline
C3 & 4 & C6 & 5 & & \\
\hline
\end{tabular}
\end{center}

You have the freedom to determine how to use the seeds to generate the random numbers.
Note that you are allowed, and encouraged, to add in any other variables or functions where you think may help in writing your code. In fact by just implement the functions mentioned above would not be closed to enough. All the guidelines here are general, so you are left with freedom where you can decide your own style of implementing the algorithm.

**GOAL:** the goal of this project is to determine if all the 6 machines can send out 100 messages successfully. Since the router has a finite buffer, overflowing it will result in failure of the whole network system. If successful, determine the number of messages received by each computer (careful how many messages in total). If failed, register at what time unit the system failed and how many messages are sent and received.

Your result should, and most likely, be different with other people in the class, since your random function might be different with other people. Also, the challenge of this project is to understand and implement classes, and also on how to use them. The algorithm for the network is not that critical.

If you have any problems you can send emails to henrykuo@ee.ucla.edu (Henry Kuo) or mattia@ee.ucla.edu (Maged Attia).