Create two files, a header and a source file. The following code should be used as a starting point:

C/C++ Supervision Sheet 2

1. The Vector3 class represents a 3-dimensional vector. Implement the non-default constructor, set(), print() and magnitude() member functions, so that compiling and running the code produces the following output (where numbers in angle-brackets are memory addresses):
   \[ \text{<a>: [1,2,2](3)} \]
   \[ \text{<b>: [2,6,9](11)} \]

2. Add the code on the right into the main function. The "multiply" operation should compute the cross product of the two vectors. Implement the required code to achieve this. The program should now produce this additional output:
   \[ \text{<c>: [6,5,2](8.06226)} \]

3. Insert the code on the right into the header file. The class represents an array-backed queue. Only pointers to items are stored. poll() returns and removes the head of the queue, and add(T) adds to the tail. add(Queue) adds all of the members of another queue. print() should send a comma-separated list of addresses of the items that are in the queue to standard output, enclosed in square brackets. Implement the member functions.

4. Add the code on the right into the main function, and ensure the following output is produced:
   \[ \text{[8,2,3](8.77496)} \]
   \[ \text{[<B>,<C>]} \]
   \[ \text{[<A>,<B>,<C>]} \]

5. Add the snippet on the right into the main function. The priority queue orders its elements from least to greatest. Vectors are ordered by magnitude. Use inheritance to implement the required functionality, and ensure the following output is produced:
   \[ \text{[<C>,<A>,<B>]} \]

6. Show how we can overload the I/O stream \"<<\" operator for Vector3 and Queue, to avoid using print(). (NB: you will need to make use of the friend keyword).

7. In its current form, Vector3 could produce a memory leak. Explain the circumstances in which this might occur, and how we can rectify this issue.

8. Show how Queue can be refactored into an abstract base class, allowing alternative implementations with e.g. a linked list.

9. Recall the last question on the first supervision sheet. Show how the example could be re-implemented in C++. Give three reasons for the superiority of the C++ solution.