Data Structures and Programming

System Definitions

The goal of this exercise is to demonstrate programming ability by developing a simple software subsystem.

A software subsystem of an air-traffic control system is defined to manage a queue of aircraft (AC) in an airport. The aircraft queue is managed by a process which responds to three types of requests:

• system boot	used to start the system.
• enqueue aircraft	used to insert a new AC into the system.
• dequeue aircraft	used to remove an AC from the system.

AC's have at least (but are not limited to having) the following properties:

• AC type:	Passenger or Cargo
• AC size:	Small or Large

The process which manages the queue of AC's satisfies the following:

- There is no limit on the number of AC's it can manage
- Dequeue aircraft requests result in selection of one AC for removal such that:
  1. Passenger AC's have removal precedence over Cargo AC's
  2. Large AC's of a given type have removal precedence over Small AC's of the same type.
  3. Earlier enqueued AC's of a given type and size have precedence over later enqueued AC's of the same type and size.

(continued on next page)

System Implementation Tasks

1. Develop one or more data structures to hold the state of an individual AC.
2. Develop one or more data structures to hold the state of the AC queue.
3. Define data structures and/or constants needed to represent requests.
4. Develop code to handle the three types of requests and follows the above guidelines to implement an aircraft queue manager as efficiently as possible. Your code may be developed as multiple classes/procedures if necessary. To the greatest extent possible, show all of the code required to implement the system.
5. Develop some test code that boots the system and performs several enqueues and dequeues.
6. [Extra credit] Future programmers may want to go into the system and add different aircraft types (such as “business jet”) and sizes (such as “tiny”). Comment your code so that future programmers will know what they need to change to support extra aircraft types and sizes. Do not actually change your code; the current system still has only types and sizes as originally described on the first page.

END OF EXERCISE