**1 (a) Stack ADT:**

* **isEmpty()**

returns True if stack is empty and False if it is not empty

* **push(item)**

append item to the stack

* **pop()**

return and pop the last item in the stack

* **top()**

return the last item

* **size()**

return the size of the stack

**1 (b) Implementing Stack ADT in Python:**

class Stack:

def \_\_init\_\_(self):

self.items = []

def isEmpty(self):

return self.items == []

def push(self, item):

self.items.append(item)

def pop(self):

return self.items.pop()

def top(self):

return self.items[len(self.items)-1]

def size(self):

return len(self.items)

**2 (a) AirQueue ADT:**

* **isEmpty()**

return True if the queue is empty and False if it is not empty

* **enqueue(item, priority)**

append the passenger to the queue. Priority should be {1,2 or 3} and item is the string.

* **dequeueFirst()**

pop and return the last item in the First class queue (priority = 1)

* **dequeueBusiness()**

pop and return the last item in the Business class queue(priority = 2)

* **dequeueEconomy()**

pop and return the last item in the Economy class queue (priority = 3)

* **dequeueAll()**

pop and print all the items in three queues according to their priorities (first are First class passengers, then Business, then Economy)

**2 (b) Implementing AirQueue ADT in Python:**

class AirQueue:

def \_\_init\_\_(self):

self.first = []

self.business =[]

self.economy =[]

def isEmpty(self):

return self.items == []

def enqueue(self,item,priority):

self.priority = priority

if self.priority == '1':

self.first.insert(0,item)

elif self.priority == '2':

self.business.insert(0,item)

elif self.priority == '3':

self.economy.insert(0,item)

def dequeueFirst(self):

return self.first.pop()

def dequeueBusiness(self):

return self.business.pop()

def dequeueEconomy(self):

return self.economy.pop()

def dequeueAll(self):

while self.first != []:

print self.first.pop(),

while self.business != []:

print self.business.pop(),

while self.economy != []:

print self.economy.pop(),