#The Circular Queue Class

class CQueue:

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

self.items=[]

self.capacity=capacity

self.size=0 #size is how many items are currently in the queue

self.head=0

self.tail=0

def enqueue(self, item):

if self.size+1>self.capacity:

raise IndexError("The Queue is full, you can't add anymore!")

else:

if len(self.items)+1>self.capacity:

self.items.insert(tail, item)

else:

self.items.append(item)

self.size+=1

self.tail=(self.tail+1)%self.capacity

def dequeue(self):

if self.size<1:

raise IndexError("The Queue is empty, you can't dequeue!")

else:

temp=self.items[self.head]

self.items[self.head]=None

self.size-=1

self.head= (self.head+1)%self.capacity

return temp

def peek(self):

if self.size<1:

raise IndexError("The Queue is empty, you can't peek!")

else:

return self.items[self.head]

def isEmpty(self):

return self.head==self.tail

def isFull(self):

return self.size==self.capacity

def getCapacity(self):

return self.capacity

def getSize(self):

return self.size

def clear(self):

self.items=None

self.items=[]

self.size=0

self.head=0

self.tail=0

def \_\_str\_\_(self):

return "A queue with items " + str(self.items)