class MyCircularQueue:

def \_\_init\_\_(self, k: int):

self.cur\_size = 0

self.capacity = k

self.queue = [0] \* k

self.front, self.rear = 0, -1

def enQueue(self, value: int) -> bool:

if self.isFull():

return False

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

self.cur\_size += 1

self.queue[self.rear] = value

return True

def deQueue(self) -> bool:

if self.isEmpty():

return False

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

self.cur\_size -= 1

return True

def Front(self) -> int:

return self.queue[self.front] if self.cur\_size else -1

def Rear(self) -> int:

return self.queue[self.rear] if self.cur\_size else -1

def isEmpty(self) -> bool:

return self.cur\_size == 0

def isFull(self) -> bool:

return self.cur\_size == self.capacity