```
class CircularArray:
  def __init__(self, lin, st, sz):
    # Initializing Variables
    self.start = 0
    self.size = 0
    self.cir = []

    # if lin = [10, 20, 30, 40, None]
    # then, CircularArray(lin, 2, 4) will generate
    # cir = [40, null, 10, 20, 30]

    # To Do.
    # Hints: set the values for initialized variables

  # Print from index 0 to len(cir) - 1
  def printFullLinear(self): #Easy
    # To Do
    pass # Remove this line

  # Print from start index and total size elements
  def printForward(self): #Easy
    # To Do
    pass # Remove this line



  def printBackward(self): #Easy
    # To Do
    pass # Remove this line

  # With no null cells
  def linearize(self): #Medium
    # To Do
    pass # Remove this line

  # Do not change the Start index
  def resizeStartUnchanged(self, newcapacity): #Medium
```

```
    # To Do
    pass # Remove this line

  # This method will check whether the array is palindrome or not
  def palindromeCheck(self): #Hard
    # To Do
    pass # Remove this line


  # This method will sort the values by keeping the start unchanged
  def sort(self):
    # To Do
    pass # Remove this line

  # This method will check the given array across the base array and if they are equivalent
interms of values return true, or else return false
  def equivalent(self, cir_arr):
    # To Do
    pass # Remove this line


  # the method take another circular array and returns a linear array containing the common
elements between the two circular arrays.
  def intersection(self, c2):
    # To Do
    pass # Remove this line


# Tester class. Run this cell after completing methods in the upper cell and
# check the output


lin_arr1 = [10, 20, 30, 40, None]


print("==========Test 1==========")
c1 = CircularArray(lin_arr1, 2, 4)
c1.printFullLinear() # This should print: 40, None, 10, 20, 30
c1.printForward() # This should print: 10, 20, 30, 40
c1.printBackward() # This should print: 40, 30, 20, 10


print("==========Test 2==========")
```

```
c1.linearize()
c1.printFullLinear() # This should print: 10, 20, 30, 40


print("==========Test 3==========")
lin_arr2 = [10, 20, 30, 40, 50]
c2 = CircularArray(lin_arr2, 2, 5)
c2.printFullLinear() # This should print: 40, 50, 10, 20, 30
c2.resizeStartUnchanged(8) # parameter --> new Capacity
c2.printFullLinear() # This should print: None, None, 10, 20, 30, 40, 50, None


print("==========Test 4==========")
lin_arr3 = [10, 20, 30, 20, 10, None, None]
c3 = CircularArray(lin_arr3, 3, 5)
c3.printForward() # This should print: 10, 20, 30, 20, 10
c3.palindromeCheck() # This should print: This array is a palindrome


print("==========Test 5==========")
lin_arr4 = [10, 20, 30, 20, None, None, None]
c4 = CircularArray(lin_arr4, 3, 4)
c4.printForward() # This should print: 10, 20, 30, 20
c4.palindromeCheck() # This should print: This array is NOT a palindrome


print("==========Test 6==========")
lin_arr5 = [10, 20, -30, 20, 50, 30, None]
c5 = CircularArray(lin_arr5, 5, 6)
c5.printForward() # This should print: 10, 20, -30, 20, 50, 30
c5.sort()
c5.printForward() # This should print: -30, 10, 20, 20, 30, 50


print("==========Test 7==========")
lin_arr6 = [10, 20, -30, 20, 50, 30, None]
c6 = CircularArray(lin_arr6, 2, 6)
c7 = CircularArray(lin_arr6, 5, 6)
c6.printForward() # This should print: 10, 20, -30, 20, 50, 30
c7.printForward() # This should print: 10, 20, -30, 20, 50, 30
print(c6.equivalent(c7)) # This should print: True
```

```
print("==========Test 8==========")
lin_arr7 = [10, 20, -30, 20, 50, 30, None, None, None]
c8 = CircularArray(lin_arr7, 8, 6)
c6.printForward() # This should print: 10, 20, -30, 20, 50, 30
c8.printForward() # This should print: 10, 20, -30, 20, 50, 30
print(c6.equivalent(c8)) # This should print: True


print("==========Test 9==========")
lin_arr8 = [10, 20, 30, 40, 50, 60, None, None, None]
c9 = CircularArray(lin_arr8, 8, 6)
c6.printForward() # This should print: 10, 20, -30, 20, 50, 30
c9.printForward() # This should print: 10, 20, 30, 40, 50, 60
print(c6.equivalent(c9)) # This should print: False


print("==========Test 10==========")
lin_arr9 = [10, 20, 30, 40, 50, None, None, None]
c10 = CircularArray(lin_arr9, 5, 5)
c10.printFullLinear() # This should print: 40, 50, None, None, None, 10, 20, 30
lin_arr10 = [5, 40, 15, 25, 10, 20, 5, None, None, None, None, None]
c11 = CircularArray(lin_arr10, 8, 7)
c11.printFullLinear() # This should print: 10, 20, 5, None, None, None, None, None, 5, 40, 15, 25
output = c10.intersection(c11)
print(output) # This should print: [10, 20, 40]
```