Χ



(https://swayam-uat-central.appspot.com)



central.appspot.com/nc_details/AICTE)

suchetajjw47@gmail.com >

AICTE (https://swayam-uat-central.appspot.com/explorer?ncCode=AICTE) » Programming and Data Structures with Python (course)



Course outline

Practice Assignments

Practice Quiz 1

Quiz 1, Mon 25 Oct 2021

PDSP Assignment 1, due Tue 2 Nov 2021

PDSP Assignment 2, due Fri 12 Nov 2021

Quiz 2, Mon 8 Nov 2021

PDSP Assignment 3, due Wed 24 Nov 2021

Programming Assignment 3

Programming Assignment 3

Due on 2021-11-24, 23:59 IST

A queue is a sequence that allows you to add elements at the end (the rear) and remove elements from the front, like a normal queue in real life. A double-ended queue, or **deque**, allows insert and remove at both ends, so it supports four operations: insert-front, insert-rear, delete-front and delete-rear.

A partial definition of the Python class Deque is given in the code window with an implementation of insertfront(). The deque is stored in a sequence of Node objects. The Deque points to the first node (head) and last node (tail) in the sequence.

Complete the definition of Deque by adding functions insertrear(), deletefront() and deleterear(). Make sure the indentation for your code is correct with respect to the enclosing class definition.

Your function will be tested using the function testdeque() which is shown above the class definition. This function creates an empty Deque and then performs the list of operations provided to it. Each element of the list of operations is one of the following:

- ("if",v), insert v using mydeque.insertfront(v)
- ("ir",v), insert v using mydeque.insertrear(v)
- ("df",), use mydeque.deletefront() to remove and return the value at the front of the queue
- ("dr",), use mydeque.deleterear() to remove and return the value at the front of the queue

At the end of the sequence, testdeque() prints out the current deque, the list of values extracted by the deletefront() and deleterear() operations and the values of the nodes pointed to by the head and tail of the deque.

- Do not write commands to read any input or print any output.
- You may define additional auxiliary functions as needed.
- . In all cases you may assume that the value passed to the function is of

(/programming_2021/progassignment? the expected type, so your function does not have to check for malformed

PDSP Assignment 4, due Fri 17 Dec 2021

Quiz 3, Thu 16 Dec 2021

PDSP Quiz 4, Thu 23 Dec 2021

PDSP Assignment 5, due Fri 31 Dec 2021

inputs.

- There are normally some public test cases and some (hidden) private test
- "Compile and run" will evaluate your submission against the public test cases.
- "Submit" will evaluate your submission against the hidden private test cases. There are 10 private test cases, with equal weightage. You will get feedback about which private test cases pass or fail, though you cannot see the actual test cases.
- Ignore warnings about "Presentation errors".
- You can submit as many times as you like. Your final submission will be used for scoring.

Here are some examples to show how the code should work.

Private Test cases used for evaluation	Input	Expected Output	Actual Output	Status
Test Case 1	testdeque([("df",),("dr",), ("if",62),("if",2),("if",75), ("df",),("ir",16),("dr",),("dr",), ("df",),("dr",),("ir",81), ("ir",28),("ir",40),("if",74), ("dr",),("dr",),("ir",12), ("ir",15),("df",)])	[75, 16,	16, 62, 2, 40,	Passed
Test Case 2	testdeque([("ir",46),("df",), ("df",),("if",27),("dr",), ("ir",96),("ir",15),("dr",), ("df",),("if",98),("if",82), ("ir",94),("if",50),("if",84), ("dr",),("df",),("dr",),("if",65), ("dr",),("if",11)])	[11, 65, 50] [46, 27, 15, 96, 94, 84, 98, 82] 11 50\n	[11, 65, 50] [46, 27, 15, 96, 94, 84, 98, 82] 11 50\n	Passed

```
[84,
                                                    [84,
           testdeque([("if",96),("if",10),
                                                               75, 47,
                                                   75, 47,
           ("if",40),("if",1),("ir",84),
                                                               44, 32,
                                                   44, 32,
           ("dr",),("df",),("df",),
                                                               74]
Test Case
                                                   74] [84,
           ("if",44),("if",47),("if",75),
                                                               [84,
                                                                           Passed
3
                                                   96, 1,
           ("dr",),("ir",32),("ir",58),
                                                               96, 1,
                                                   40, 10,
           ("if",84),("if",35),("dr",),
                                                               40, 10,
                                                   58, 35]
           ("ir",74),("df",)])
                                                               58, 351
                                                   84 74\n
                                                               84 74\n
                                                               [87,
                                                    [87,
                                                               71, 48,
           testdeque([("if",17),("ir",43),
                                                   71, 48,
                                                               6, 17,
           ("if",37),("dr",),("ir",67),
                                                   6, 17,
                                                               67, 55,
           ("df",),("ir",55),("if",44),
                                                   67, 55,
Test Case
                                                               4, 58,
                                                   4, 58,
           ("ir",54),("df",),("if",6),("dr",),
                                                                           Passed
                                                               51]
           ("if",48),("ir",4),("if",71),
                                                   51] [43,
                                                               [43,
           ("if",27),("ir",58),("df",),
                                                   37, 44,
                                                               37, 44,
           ("if",87),("ir",51)])
                                                   54, 27]
                                                               54, 27]
                                                   87 51\n
                                                               87 51\n
                                                               [51,
                                                    [51,
           testdeque([("if",84),("dr",),
                                                               65, 40,
                                                   65, 40,
           ("ir",62),("ir",60),("dr",),
                                                               38, 89,
                                                   38, 89,
           ("ir",38),("if",99),("df",),
                                                               60]
Test Case
                                                   60] [84,
           ("df",),("if",40),("ir",59),
                                                               [84,
                                                                           Passed
5
                                                   60, 99,
           ("dr",),("if",65),("if",51),
                                                               60, 99,
                                                   62, 59,
           ("ir",89),("ir",60),("ir",86),
                                                               62, 59,
                                                   38, 86]
           ("if",38),("df",),("dr",)])
                                                               38, 86]
                                                   51 60\n
                                                               51 60\n
           testdeque([("ir",46),("dr",),
           ("df",),("if",56),("if",68),
                                                               [28, 2]
                                                    [28, 2]
           ("dr",),("dr",),("df",),("dr",),
                                                               [46,
                                                   [46, 56,
           ("ir",36),("dr",),("ir",34),
                                                               56, 68,
                                                   68, 36,
           ("df",),("df",),("if",50),
                                                               36, 34,
                                                   34, 0,
           ("ir",48),("if",0),("df",),
                                                               0, 47,
Test Case
                                                   47, 96,
           ("ir",96),("if",47),("df",),
                                                               96, 48,
                                                                           Passed
6
                                                   48, 41,
                                                               41, 50,
           ("if",41),("dr",),("dr",),("df",),
                                                   50, 29,
           ("df",),("ir",29),("df",),
                                                               29, 92,
                                                   92, 53,
                                                               53, 93,
           ("if",92),("df",),("df",),("df",),
                                                   93, 23]
           ("ir",23),("if",53),("df",),
                                                               23] 28
                                                   28 2\n
           ("if",2),("if",93),("df",),("dr",),
                                                               2\n
           ("if",28)1)
```

Test Case 7	<pre>testdeque([("dr",),("if",41), ("ir",87),("ir",55),("ir",25), ("if",1),("df",),("df",),("if",84), ("dr",),("if",59),("ir",30), ("dr",),("if",5),("ir",41), ("if",49),("ir",68),("df",), ("dr",),("ir",84),("dr",),("df",), ("dr",),("df",),("df",),("dr",), ("dr",),("ir",97),("ir",54), ("if",97),("ir",68),("ir",13), ("ir",71),("ir",66),("if",11), ("if",14),("ir",45),("df",), ("if",71),("df",)])</pre>	[11, 97, 97, 54, 68, 13, 71, 66, 45] [1, 41, 25, 30, 49, 68, 84, 5, 41, 59, 84, 55, 87, 14, 71] 11 45\n	54, 68, 13, 71, 66, 45] [1, 41, 25, 30, 49, 68, 84, 5, 41, 59, 84, 55, 87, 14,	Passed
Test Case 8	testdeque([("ir",2),("ir",79), ("dr",),("ir",52),("ir",80), ("if",21),("ir",52),("ir",63), ("if",45),("ir",52),("dr",), ("dr",),("dr",),("if",59), ("ir",36),("df",),("ir",59), ("dr",),("if",59),("if",31), ("dr",),("ir",97),("df",), ("ir",23),("if",78),("df",), ("if",5),("df",),("ir",86),("dr",), ("if",98),("if",42),("if",79), ("df",),("df",),("if",16), ("ir",95),("dr",),("if",16), ("ir",95),("dr",),("dr",)]) [16, 98, 59, 21, 2, 52, 80, 97] [79, 52, 63, 52, 45, 22, 59, 36, 31, 78, 5, 86, 79, 42, 95, 23] 16	[16, 98, 59, 21, 2, 52, 80, 97] [79, 52, 63, 52, 45, 22, 59, 36, 31, 78, 5, 86, 79, 42, 95, 23] 16 97\n	[16, 98, 59, 21, 2, 52, 80, 97] [79, 52, 63, 52, 45, 22, 59, 36, 31, 78, 5, 86, 79, 42, 95, 23] 16 97\n	Passed
Test Case 9	testdeque([("df",),("dr",), ("if",34),("if",55),("dr",), ("dr",),("ir",65),("if",87), ("df",),("ir",43),("if",96), ("df",),("ir",0),("dr",),("df",), ("if",71),("df",),("if",64), ("dr",),("if",34),("if",18), ("df",),("dr",),("dr",),("ir",89), ("df",),("if",84),("if",34), ("ir",51),("df",),("ir",77), ("dr",),("if",33),("ir",48), ("if",40),("if",21),("ir",81), ("if",7),("df",),("if",30)])	[30, 21, 40, 33, 84, 51, 48, 81] [34, 55, 87, 96, 0, 65, 71, 43, 18, 64, 34, 89, 34, 77, 7] 30 81\n	[30, 21, 40, 33, 84, 51, 48, 81] [34, 55, 87, 96, 0, 65, 71, 43, 18, 64, 34, 89, 34, 77, 7] 30 81\n	Passed

```
[35,
           testdeque([("if",46),("if",35),
                                                     [35,
                                                               46, 72,
           ("ir",94),("if",95),("if",7),
                                                   46, 72,
                                                               84, 81,
           ("if",61),("df",),("df",),("dr",),
                                                   84, 81,
                                                               27]
           ("if",53),("df",),("if",48),
                                                   27] [61,
                                                               [61, 7,
           ("ir",72),("df",),("df",),
                                                   7, 94,
                                                               94, 53,
           ("ir",84),("if",86),("ir",0),
                                                   53, 48,
Test Case
                                                               48, 95,
           ("dr",),("ir",4),("df",),("if",16),
                                                   95, 0,
                                                                           Passed
10
                                                               0, 86,
           ("if",0),("ir",51),("df",),
                                                   86, 0,
                                                               0, 51,
           ("if",45),("dr",),("ir",31),
                                                   51, 45,
                                                               45, 31,
           ("df",),("if",89),("dr",),
                                                   31, 89,
                                                               89, 32,
           ("ir",32),("df",),("dr",),
                                                   32, 80,
                                                               80, 4,
           ("if",80),("df",),("dr",),
                                                   4, 16]
                                                               16] 35
           ("ir",81),("df",),("ir",27)])
                                                   35 27\n
                                                               27\n
```

The due date for submitting this assignment has passed.

10 out of 10 tests passed.

You scored 100.0/100.

Assignment submitted on 2021-11-23, 17:56 IST

Your last recorded submission was :

```
def testdeque(1):
 2
     mydeque = Deque()
     extractedlist = []
 3
 4
     for op in 1:
        if op[0] == "if":
 5
 6
          mydeque.insertfront(op[1])
        elif op[0] == "ir":
 7
 8
          mydeque.insertrear(op[1])
 9
        elif op[0] == "df":
10
          v = mydeque.deletefront()
11
          if v != None:
12
            extractedlist.append(v)
        elif op[0] == "dr"
13
          v = mydeque.deleterear()
14
          if v != None:
15
16
            extractedlist.append(v)
17
     print(mydeque,extractedlist,mydeque.head.value,mydeque.tail.value)
18
     return
19
   ##########
20
21
22
23
   class Node:
        ef __init__(self):
self.value = None
self.next = None
     def
24
25
26
27
28
   class Deque:
          __init_
29
     def
                   (self):
30
        newnode = Node()
        self.head = newnode
31
32
        self.tail = newnode
33
34
     def isempty(self):
35
        return(self.head.value == None)
36
37
     def insertfront(self,v):
38
        if self.isempty():
39
          self.head.value = v
40
        else:
          newnode = Node()
41
          newnode.value = self.head.value
42
43
          newnode.next = self.head.next
```

```
44
          self.head.value = v
45
          self.head.next = newnode
          self.tail = newnode
while self.tail.next != None:
46
47
48
            self.tail = self.tail.next
49
        f __str__(self):
if self.head.value == None:
 50
 51
52
          return(str([]))
53
        else:
          ptr = self.head
54
55
          myl = [ptr.value]
 56
          while ptr.next != None:
57
            ptr = ptr.next
 58
            myl = myl + [ptr.value]
59
          return(str(myl))
60
 61
      62
63
      def insertrear(self,v):
 64
 65
        if self.isempty():
66
            self.head.value = v
67
        else:
            newnode = Node()
68
69
            newnode.value = v
70
            newnode.next = None
71
            node = Node()
            node = self.head
72
73
            while node.next != None:
74
              node = node.next
 75
            node.next = newnode
            while self.tail.next!= None:
76
 77
               self.tail = self.tail.next
 78
      def deletefront(self):
79
80
        if self.isempty():
81
            return
82
        if self.head.next == None:
83
            v = self.head.value
            self.head = Node()
84
85
            self.tail = Node()
86
            return v
87
88
             v = self.head.value
89
            self.head = self.head.next
            return v
90
91
92
      def deleterear(self):
93
        if self.isempty():
94
            return
95
        if self.head.next == None:
            v = self.head.value
96
            self.head = Node()
 97
98
            self.tail = Node()
99
            return v
100
        else:
101
            node = Node()
102
            prev = Node()
103
            node = self.head
104
            while(node.next!=None):
105
              pre\dot{v} = node
106
              node = node.next
107
            v = node.value
            self.tail = prev
108
109
            self.tail.next = None
110
            return v
111 import ast
112
113 def parse(inp):
114
      inp = ast.literal_eval(inp)
115
      return (inp)
116
116
117 | fncall = input().strip()
fncall find("(")
lnaren = fncall.find("(")
rparen = fncall.rfind(")")
120 fname = fncall[:lparen]
```

```
121 | farg = fncall[lparen+1:rparen]
122
123 if fname == "testdeque":
124
       arg = parse(farg)
125
       testdeque(arg)
126 else:
127
       print("Function", fname, "unknown")
128
Sample solutions (Provided by instructor)
     def testdeque(1):
       mydeque = Deque()
  3
       extractedlist = []
  4
       for op in 1:
         if op[0] == "if":
  5
         mydeque.insertfront(op[1])
elif op[0] == "ir":
  6
  8
         mydeque.insertrear(op[1])
elif op[0] == "df":
  9
 10
           v = mydeque.deletefront()
           if v != None:
 11
             extractedlist.append(v)
 12
 13
         elif op[0] == "dr":
           v = mydeque.deleterear()
 14
 15
           if v != None:
             extractedlist.append(v)
 16
       print(mydeque,extractedlist,mydeque.head.value,mydeque.tail.value)
 17
 18
       return
 19
 20
     ##########
 21
 22
 23
     class Node:
       def
 24
            __init__(self):
         se\overline{lf.value} = None
 25
 26
         self.next = None
 27
 28
     class Deque:
       def __init__(self):
    newnode = Node()
 29
 30
         self.head = newnode
 31
 32
         self.tail = newnode
 33
 34
       def isempty(self):
         return(self.head.value == None)
 35
 36
       def insertfront(self,v):
 37
         if self.isempty():
 38
           self.head.value = v
 39
 40
         else:
 41
           newnode = Node()
 42
           newnode.value = self.head.value
           newnode.next = self.head.next
 43
           self.head.value = v
 44
 45
           self.head.next = newnode
           self.tail = newnode
 46
           while self.tail.next != None:
 47
 48
             self.tail = self.tail.next
 49
         ef __str__(self):
if self.head.value == None:
 50
 51
 52
           return(str([]))
 53
         else:
 54
           ptr = self.head
 55
           myl = [ptr.value]
 56
           while ptr.next != None:
 57
             ptr = ptr.next
             myl = myl + [ptr.value]
 58
 59
           return(str(myl))
 60
       61
       # Complete the definition of Deque below this #
 62
       63
 64
       def insertrear(self,v):
         if self.isempty():
 65
 66
           self.head.value = v
```

```
67
                            else:
                                   newnode = Node()
   68
   69
                                   newnode.value = v
                                   newnode.next = None
   70
   71
                                   self.tail.next = newnode
   72
                                   self.tail = newnode
   73
   74
                     def deletefront(self):
   75
                            if self.isempty():
   76
                                   return
   77
                            retval = self.head.value
   78
   79
                            if self.head.next == None:
   80
   81
                                   self.head.value = None
   82
                            else:
                                   self.head = self.head.next
   83
   84
   85
                            return(retval)
   86
   87
                     def deleterear(self):
   88
                            if self.isempty():
   89
                                   return
   90
  91
                            retval = self.tail.value
   92
   93
                            if self.head.next == None:
                                   self.head.value = None
   94
   95
                                   return(retval)
   96
   97
                            ptr = self.head
   98
                            while ptr.next != self.tail:
  99
                                   ptr = ptr.next
100
101
                            ptr.next = None
                            self.tail = ptr
102
103
104
                            return(retval)
105
106 #########
107
108 import ast
109
110 def parse(inp):
111   inp = ast.literal_eval(inp)
                     return (inp)
112
113
fncall = input().strip()
fncall = input()
fname = fncall[:lparen]
118 farg = fncall[lparen+1:rparen]
119
120 if fname == "testdeque":
121 arg = parse(farg)
122
                     testdeque(arg)
123 else:
                     print("Function", fname, "unknown")
124
125
```