

七月在线 python 基础第三节课作业

Author: 粽子

2016/11/09

A. 螺旋矩阵：给定一个 $m * n$ 要素的矩阵。按照螺旋顺序，返回该矩阵的所有要素。

[[1, 2, 3],

[4, 5, 6],

[7, 8, 9]]

应该返回[1, 2, 3, 6, 9, 8, 7, 4, 5]

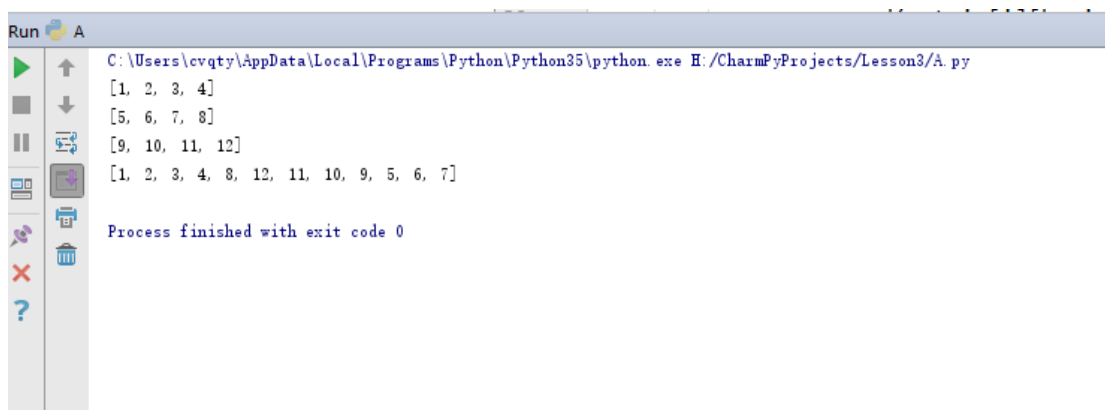
思路：就是正常的打印就好，从左往右，从上往下，从右往左，从下往上，代码如下：

```
1 #Lesson3 Homework A
2 def spiral_matrix(matrix, m ,n):
3     ans = []
4     beginX = 0
5     endX = n-1
6     beginY = 0
7     endY = m-1
8     while True:
9         #left -> right
10        for j in range(beginX, endX+1, 1):
11            ans.append(matrix[beginY][j])
12        beginY += 1
13        if beginY > endY:
14            break
15        #top -> buttom
16        for i in range(beginY, endY+1, 1):
17            ans.append(matrix[i][endX])
18        endX -= 1
19        if beginX > endX:
20            break
21        #right -> left
22        for j in range(endX, beginX-1, -1):
23            ans.append(matrix[endY][j])
24        endY -= 1
25        if beginY > endY:
26            break
27        #buttom -> top
28        for i in range(endY, beginY-1, -1):
29            ans.append(matrix[i][beginX])
30        beginX += 1
31        if beginX > endX:
32            break
33    return ans
```

```

34
35 mat = [[1,2,3,4],[5,6,7,8],[9,10,11,12]]
36 for i in range(0, len(mat), 1):
37     print(mat[i])
38
39 ans = spiral_matrix(mat, 3, 4)
40 print(ans)

```



```

Run A
C:\Users\cvqty\AppData\Local\Programs\Python\Python35\python.exe H:/CharmPyProjects/Lesson3/A.py
[1, 2, 3, 4]
[5, 6, 7, 8]
[9, 10, 11, 12]
[1, 2, 3, 4, 8, 12, 11, 10, 9, 5, 6, 7]
Process finished with exit code 0

```

B. 用栈（使用 list）实现队列：支持 push(element), pop() 和 top() 方法。pop 和 top 方法都应该返回第一个元素的值。比如执行以下操作序列：push(1), pop(), push(2), push(3), top(), pop(), 你应该返回 1, 2 和 2。

思路：我是自己实现了一个 queue 的类，包含了 pop(), push(), top(), print_queue() 这四个方法，代码如下：

```

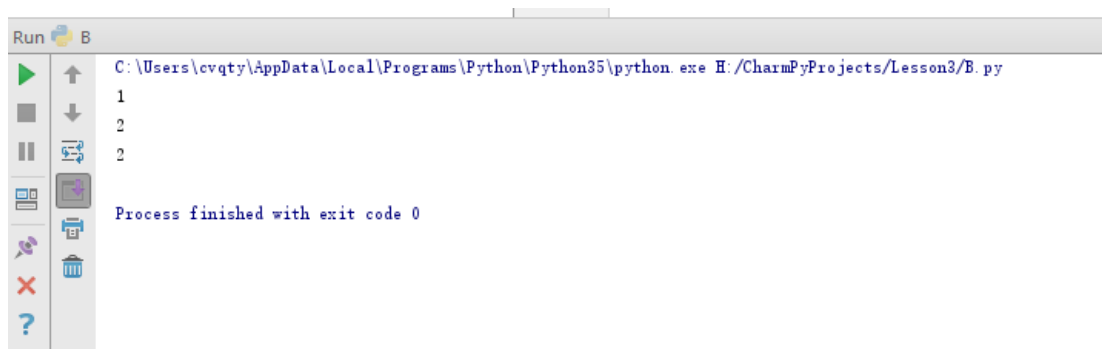
1 #Lesson3 Homework B
2 # use 2 stacks implement a queue
3 class queue:
4     def __init__(self):
5         self.stk1 = []
6         self.stk2 = []
7     def top(self):
8         while len(self.stk1) >= 2:
9             self.stk2.append(self.stk1.pop())
10        return self.stk1[0]
11    def pop(self):
12        while len(self.stk1) >= 2:
13            self.stk2.append(self.stk1.pop())
14        return self.stk1.pop()
15    def push(self, element):
16        while len(self.stk2):
17            self.stk1.append(self.stk2.pop())
18        self.stk1.append(element)

```

```

19     def print_queue(self):
20         while self.stk2:
21             self.stk1.append(self.stk2.pop())
22             for i in range(0, len(self.stk1), 1):
23                 print(self.stk1[i], ' ', end='')
24             print('')
25 #push(1), pop(), push(2), push(3), top(), pop()
26 q = queue()
27 q.push(1)
28 print(q.pop())
29 q.push(2)
30 q.push(3)
31 print(q.top())
32 print(q.pop())

```



C. 矩阵转换：给定矩阵 A，令矩阵 B 里每个元素 $B[i][j]$ 的值等于 $A[0][0]$ 到 $A[i][j]$ 子矩阵元素的和。

思路：

	0	1	2	3	4	5	6	7
0								
1								
2								
3								

不想每一个格子都算一遍，这样肯定不够好，所以，我采用了这种策略：

B 的每个元素都是 $A[0][0]$ 到 $A[i][j]$ 的子矩阵的和，所以，可以推导出如下公式：

$$B[i][j] = B[i][j-1] + B[i-1][j] - B[i-1][j-1] + A[i][j]$$

先计算出 $B[0][j]$ 和 $B[i][0]$ ，再套用以上公式。

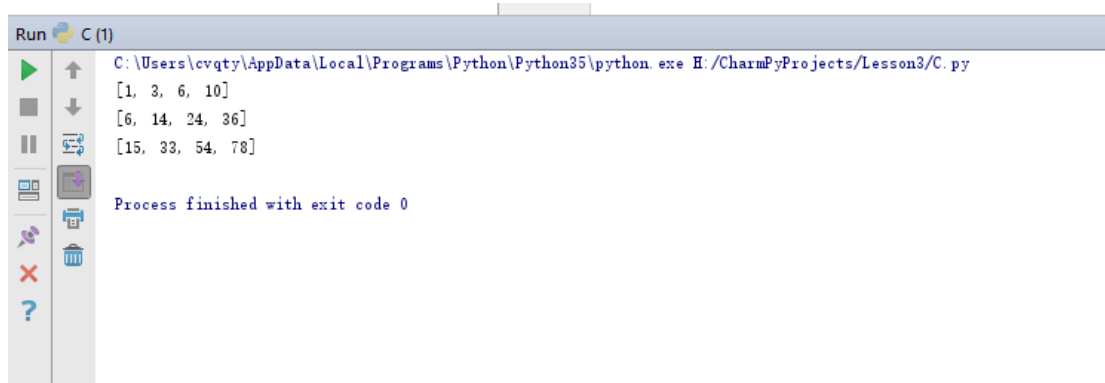
比如说计算 $B[2][4] = B[2][3] + B[1][4] - B[1][3] + A[2][4]$

代码如下：

```

1 #Lesson3 Homework C
2 #Given matrix A, calculate matrix B which makes B[i][j] =
3 A[0][0]+ ...+ A[i][j]
4 #PS:A[0][0] ...A[i][j] forms a sub-matrix
5 def calculate_B(A):
6     B = A
7     for i in range(1, len(A[0]), 1):
8         B[0][i] += B[0][i-1]
9     for i in range(1, len(A), 1):
10        B[i][0] += B[i-1][0]
11
12    for i in range(1, len(A), 1):
13        for j in range(1, len(A[0]), 1):
14            B[i][j] += B[i-1][j] + B[i][j-1] - B[i-1][j-1]
15    return B
16
17 A = [[1,2,3,4],[5,6,7,8],[9,10,11,12]]
18 B = calculate_B(A)
19 for i in range(0, len(B), 1):
20     print(B[i])

```



```

Run C (1)
C:\Users\cvqty\AppData\Local\Programs\Python\Python35\python.exe H:/CharmPyProjects/Lesson3/C.py
[1, 3, 6, 10]
[6, 14, 24, 36]
[15, 33, 54, 78]
Process finished with exit code 0

```

D. 翻转单向链表

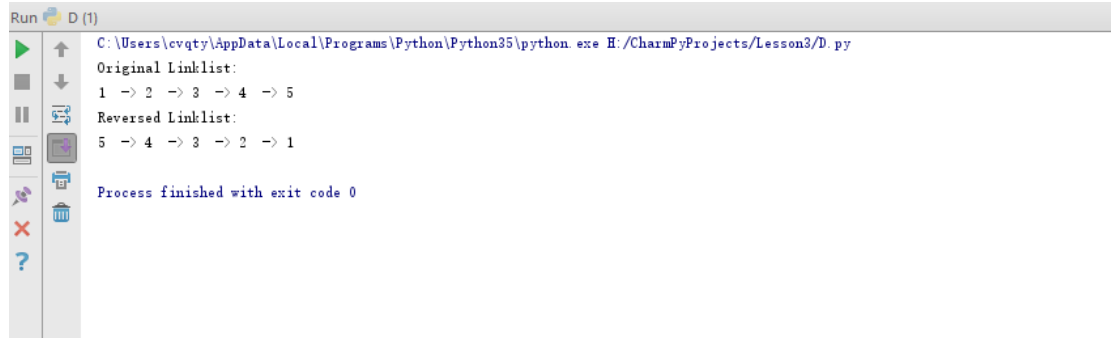
```

class ListNode:
    def __init__(self, val, next = None):
        self.val = val
        self.next = next

```

思路就不多说了，直接上代码：

```
1 #Lesson3 Homework D
2 #Reverse LinkedList
3 class ListNode:
4     def __init__(self, val, next = None):
5         self.val = val
6         self.next = next
7
8 def reverse_linklist(head):
9     if head==None:
10         return head
11     pre = None
12     cur = head
13     ne = None
14     while cur:
15         ne = cur.next
16         cur.next = pre
17         pre = cur
18         cur = ne
19     return pre
20
21 def print_linklist(head):
22     if head == None:
23         print('Empty linklist')
24     cur = head
25     #learn how to make print start with a new line
26     while cur.next:
27         print(cur.val, ' -> ', end='')
28         cur = cur.next
29     print(cur.val)
30
31 node_obj = []
32 for i in range(0, 5, 1):
33     node_obj.append(ListNode(i+1))
34 for i in range(0, len(node_obj)-1, 1):
35     node_obj[i].next = node_obj[i+1]
36 print('Original Linklist:')
37 print_linklist(node_obj[0])
38 head1 = reverse_linklist(node_obj[0])
39 print('Reversed Linklist:')
40 print_linklist(head1)
```



The screenshot shows the Run console of a Python IDE. The console title is "Run D (1)". The command executed is `C:\Users\cvqty\AppData\Local\Programs\Python\Python35\python.exe H:/CharmPyProjects/Lesson3/D.py`. The output consists of three lines: "Original Linklist:", "1 -> 2 -> 3 -> 4 -> 5", and "Reversed Linklist:". The next line shows the reversed list: "5 -> 4 -> 3 -> 2 -> 1". The final line indicates "Process finished with exit code 0". On the left side of the console, there is a vertical toolbar with icons for running, stepping through, and other debugging actions.

```
Run D (1)
C:\Users\cvqty\AppData\Local\Programs\Python\Python35\python.exe H:/CharmPyProjects/Lesson3/D.py
Original Linklist:
1 -> 2 -> 3 -> 4 -> 5
Reversed Linklist:
5 -> 4 -> 3 -> 2 -> 1
Process finished with exit code 0
```