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。

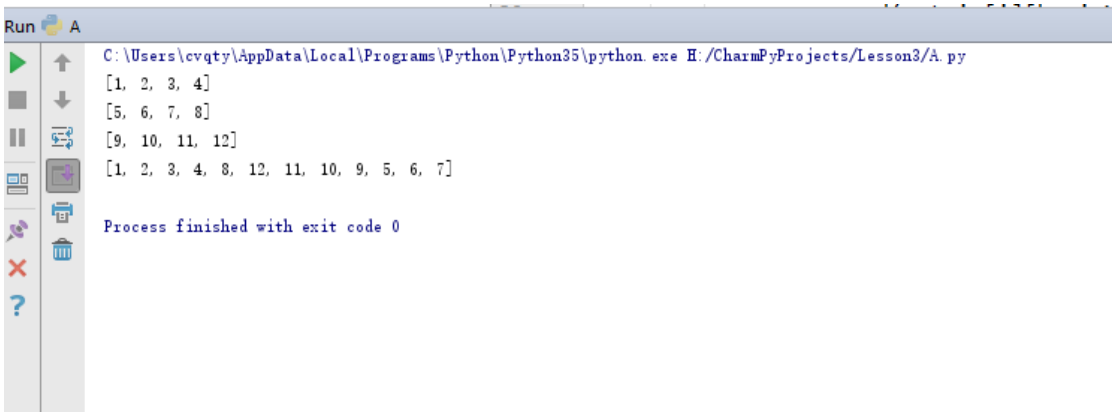
[[1, 2, 3],
 [4, 5, 6],
 [7, 8, 9]]

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：

```
#Lesson3 Homework A
def spiral_matrix(matrix, m ,n):
    ans = []
    beginX = 0
    endX = n-1
    beginY = 0
    endY = m-1
    while True:
        #left -> right
        for j in range(beginX, endX+1, 1):
            ans.append(matrix[beginY][j])
        beginY += 1
        if beginY > endY:
            break
        #top -> buttom
        for i in range(beginY, endY+1, 1):
            ans.append(matrix[i][endX])
        endX -= 1
        if beginX > endX:
            break
        #right -> left
        for j in range(endX, beginX-1, -1):
            ans.append(matrix[endY][j])
        endY -= 1
        if beginY > endY:
            break
        #buttom -> top
        for i in range(endY, beginY-1, -1):
            ans.append(matrix[i][beginX])
        beginX += 1
        if beginX > endX:
            break
    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)
```

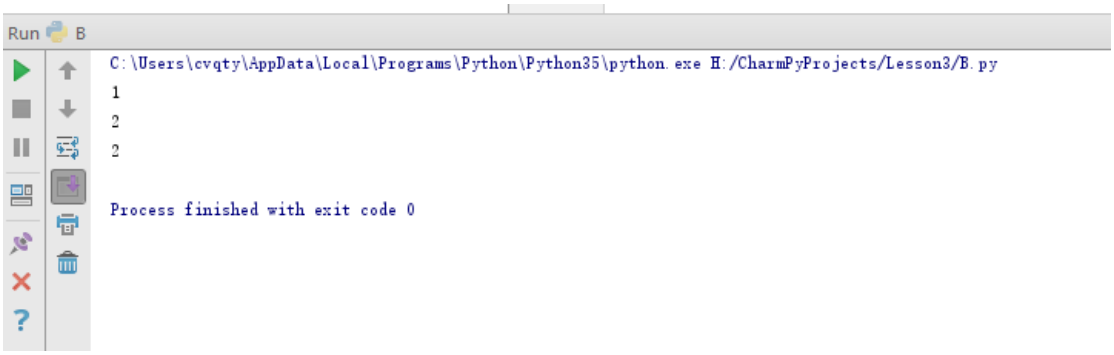


B. 用栈（使用 list）实现队列：支持 push(element)，pop()和 top()方法。pop 和 top 方法都应该返回第一个元素的值。比如执行以下操作序列：push(1)，pop()，push(2)，push(3)，top()，pop()，你应该返回 1，2 和 2。

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：

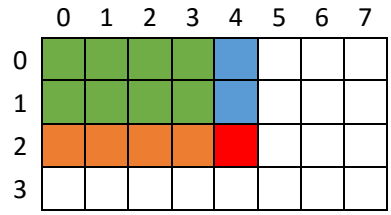
```
1  #Lesson3 Homework B
2  # use 2 stacks implement a queue
3  class queue:
4      def __init__(self):
5          self.stk1 = []
6          self.stk2 = []
7      def top(self):
8          while len(self.stk1) >= 2:
9              self.stk2.append(self.stk1.pop())
10         return self.stk1[0]
11     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：



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.

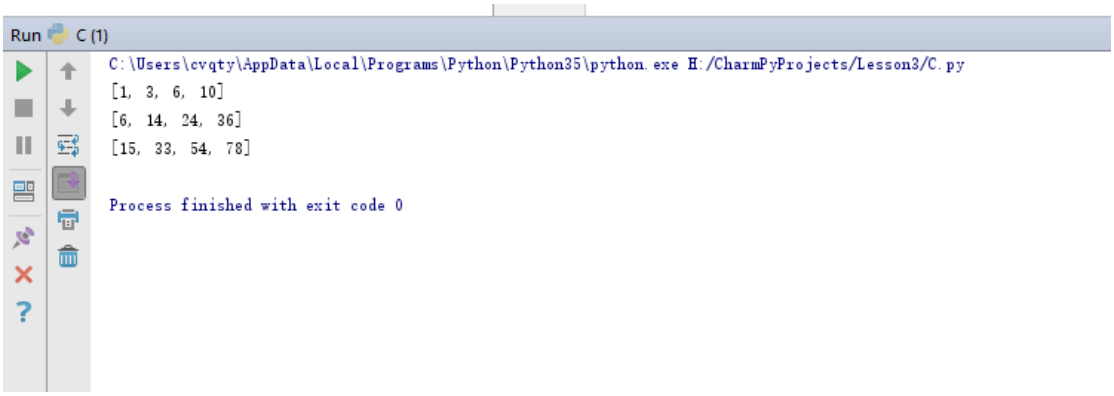
比如说计算 B[2][4] = B[2][3] + B[1][4] – B[1][3] + A[2][4]

代码如下：

```python
#Lesson3 Homework C
#Given matrix A, calculate matrix B which makes B[i][j] =
A[0][0]+ ...+ A[i][j]
#PS:A[0][0] ...A[i][j] forms a sub-matrix
def calculate_B(A):
    B = A
    for i in range(1, len(A[0]), 1):
        B[0][i] += B[0][i-1]
    for i in range(1, len(A), 1):
        B[i][0] += B[i-1][0]

    for i in range(1, len(A), 1):
        for j in range(1, len(A[0]), 1):
            B[i][j] += B[i-1][j] + B[i][j-1] - B[i-1][j-1]
    return B

A = [[1,2,3,4],[5,6,7,8],[9,10,11,12]]
B = calculate_B(A)
for i in range(0, len(B), 1):
    print(B[i])
```



D. 翻转单向链表

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

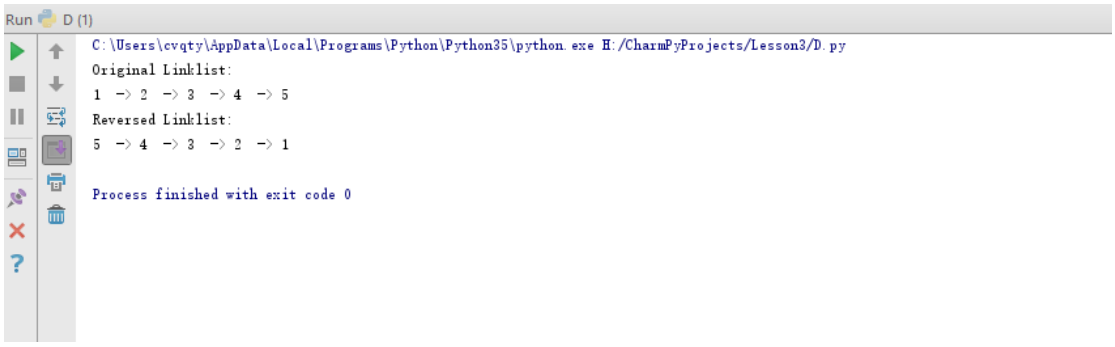
思路就不多说了，直接上代码：

```
#Lesson3 Homework D
#Reverse LinkedList
class ListNode:
    def __init__(self, val, next = None):
        self.val = val
        self.next = next

def reverse_linklist(head):
    if head==None:
        return head
    pre = None
    cur = head
    ne = None
    while cur:
        ne = cur.next
        cur.next = pre
        pre = cur
        cur = ne
    return pre

def print_linklist(head):
    if head == None:
        print('Empty linklist')
    cur = head
    #learn how to make print start with a new line
    while cur.next:
        print(cur.val, ' -> ', end='')
        cur = cur.next
    print(cur.val)

node_obj = []
for i in range(0, 5, 1):
    node_obj.append(ListNode(i+1))
for i in range(0, len(node_obj)-1, 1):
    node_obj[i].next = node_obj[i+1]
print('Original Linklist:')
print_linklist(node_obj[0])
head1 = reverse_linklist(node_obj[0])
print('Reversed Linklist:')
print_linklist(head1)
```


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