1.

2.

（1）散列表

代码：

# 散列表：字典

# 初始化

dict = {'key1': 'value1'}

# 增加新的键值对或更新已有键值对

dict['key2'] = 'newValue'

dict['key1'] = 'newValue'

# 取出键值对

print(dict.pop('key1'))

print('key1' in dict) # 观察键key1是否在dict中

运行结果：

newValue

False

（2）队列：使用python内置库queue

代码：

import queue

# 初始化队列（长度为maxsize）

q = queue.Queue(maxsize=4)

# 顺序入队1，2，3，4

for i in range(4):

q.put(i)

# 观察出队顺序

for i in range(4):

print("出队列元素{}".format(q.get()))

运行结果：

出队列元素0

出队列元素1

出队列元素2

出队列元素3

（3）栈：只需要满足“后入先出”即LIFO原则，因此仍可使用queue库

代码：

# 初始化栈（后入先出“队列”）

stack = queue.LifoQueue(maxsize=4)

# 顺序入队1，2，3，4

for i in range(4):

stack.put(i)

# 观察出栈顺序

for i in range(4):

print("出栈元素{}".format(stack.get()))

运行结果：

出栈元素3

出栈元素2

出栈元素1

出栈元素0

3. 运行结果

17

23

-60

-6.666666666666667

-1

0.000125

-7

3

10

3.14

(20-3j)

(20+3j)

(-7, -1)

4. 代码

n = 9

for i in range(1, n+1): # range(start, end), 区间左开右闭

for j in range(1, i+1):

# 直接print

# print(str(j)+"\*"+str(i)+"="+str(i\*j), end=" ")

# .format写法

print("{}\*{}={}".format(j,i,j\*i), end=' ')

print()

运行结果

1\*1=1

1\*2=2 2\*2=4

1\*3=3 2\*3=6 3\*3=9

1\*4=4 2\*4=8 3\*4=12 4\*4=16

1\*5=5 2\*5=10 3\*5=15 4\*5=20 5\*5=25

1\*6=6 2\*6=12 3\*6=18 4\*6=24 5\*6=30 6\*6=36

1\*7=7 2\*7=14 3\*7=21 4\*7=28 5\*7=35 6\*7=42 7\*7=49

1\*8=8 2\*8=16 3\*8=24 4\*8=32 5\*8=40 6\*8=48 7\*8=56 8\*8=64

1\*9=9 2\*9=18 3\*9=27 4\*9=36 5\*9=45 6\*9=54 7\*9=63 8\*9=72 9\*9=81