class ListItem(object):

def \_\_init\_\_(self ,key,value):

self.key = key

self.value = value

class SortSequenceList(object):

def \_\_init\_\_(self):

self.SeqList=[]

def CreateSequenceListByInput(self,nElement):

self.SeqList.append(ListItem(int(0), 0))

print("请输入数据：")

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

a = input()

self.SeqList.append(ListItem(int(a), i))

def TraverseElementSet(self):

for i in range(1,len(self.SeqList)):

print(self.SeqList[i].key)

############################

#算法8-9 快速排序的一趟分区

############################

def AdjustPartition(self,low,high):

left=low

right=high

self.SeqList[0].key = self.SeqList[left].key

while left < right:

while left < right and self.SeqList[right].key >= self.SeqList[0].key:

right = right-1

self.SeqList[left].key = self.SeqList[right].key

while left < right and self.SeqList[left].key <= self.SeqList[0].key:

left = left+1

self.SeqList[right].key = self.SeqList[left].key

self.SeqList[left].key= self.SeqList[0].key

return left

############################

#算法8-10 快速排序

############################

def QuickSort(self,low,high):

if low<high:

pivot = self.AdjustPartition(low,high)

self.QuickSort(low,pivot-1)

self.QuickSort(pivot+1,high)

if \_\_name\_\_ =='\_\_main\_\_':

SL=SortSequenceList()

SL.CreateSequenceListByInput(10)

SL.QuickSort(1,10)

print('排序算法结果为:')

SL.TraverseElementSet()