**第十一章 算法与数据结构基础**

**姓名**：胡浩龙，**学号**：2018050576

**第一题**：参照例11.3实现在列表中顺序查找特定数值的程序。修改程序，设法从命令行参数中获取要查询的结果。

**源代码**：

def search(alist,item):

pos=0

found=False

while pos<len(alist) and not found:

if(alist[pos])==item:

found=True

else:

pos=pos+1

return found

#test

import sys

a=[int(i) for i in sys.argv[1:]]

b=int(input("please input the number you wanna search:"))

print(search(a,b))

**输出结果：**

****

**第二题**：参考例11.6实现二分查找法的非递归程序。修改程序，设法从命令行参数列表中获取测试列表的各个元素以及所要查找的关键字。

**源代码**：

def search(key,a):

low=0

high=len(a)-1

while low<=high:

mid=(low+high)//2

if a[mid]<key:

low=mid+1

elif a[mid]>key:

high=mid-1

else:

return mid

return -1

import sys

b=[int(i) for i in sys.argv[1:len(sys.argv)-1]]

keys=int(sys.argv[len(sys.argv)-1])

temp=search(keys,b)

if temp!=-1:

print("I have found it in the %d place"%(temp+1))

else:

print("I can not found!")

**输出结果：**



**第三题**：参考例11.9实现选择排序算法程序，修改程序，设法从命令行参数中获取测试列表的各元素。

**源代码**：

def selectionSort(a):

for i in range(0,len(a)):

m=i

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

if(a[j]<a[m]):

m=j

a[i],a[m]=a[m],a[i]

def main():

import sys

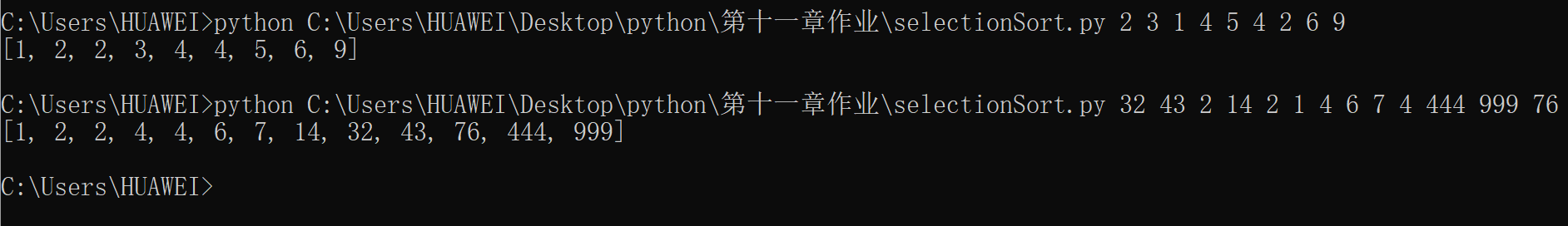
a=[int(i) for i in sys.argv[1:]]

selectionSort(a)

print(a)

if \_\_name\_\_=='\_\_main\_\_':main()

**输出结果：**



**第四题**：参考例11.11实现归并排序算法程序，修改程序，设法从命令行参数中获取测试列表的各元素。

**源代码**：

def merge(left,right):

merged=[]

i,j=0,0

while i<len(left) and j< len(right):

if left[i]<=right[j]:

merged.append(left[i])

i+=1

else:

merged.append(right[j])

j+=1

merged.extend(left[i:])

merged.extend(right[j:])

return merged

def mergeSort(a):

if len(a)<=1:

return a

mid=len(a)//2

left=mergeSort(a[:mid])

right=mergeSort(a[mid:])

return merge(left,right)

def main():

import sys

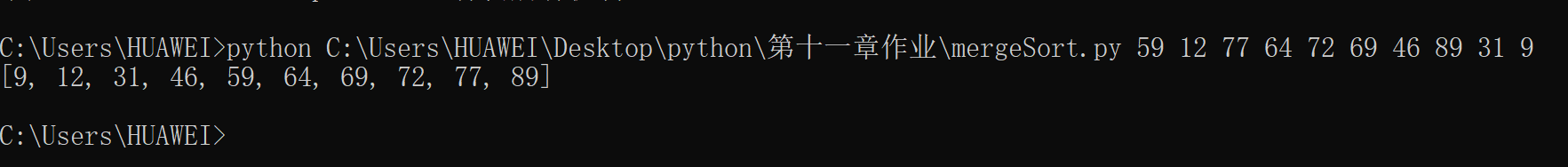
a=[int(i) for i in sys.argv[1:]]

a1=mergeSort(a)

print(a1)

if \_\_name\_\_=='\_\_main\_\_':main()

**输出结果：**



**第五题**：参照例11.3实现快速排序算法。修改程序，设法从命令行参数中获取要查询的结果。

**源代码**：

def quickSort(a,low,high):

i=low

j=high

if i>=j:

return a

key=a[i]

while i<j:

while i<j and a[j]>key:

j=j-1

a[i]=a[j]

while i<j and a[i]<=key:

i=i+1

a[j]=a[i]

a[i]=key

quickSort(a,low,i-1)

quickSort(a,j+1,high)

def main():

import sys

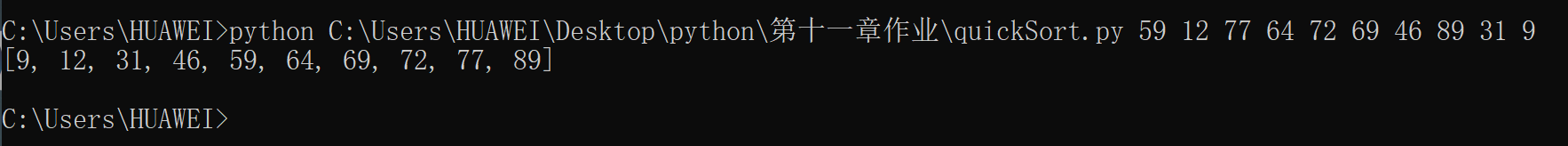
a=[int(i) for i in sys.argv[1:]]

quickSort(a,0,len(a)-1)

print(a)

if \_\_name\_\_=='\_\_main\_\_':main()

**输出结果：**



**第六题**：参考例11.16实现栈的程序。创造一个1~10的栈，展示入栈和出栈的操作，以及打印栈的内容。

**源代码**：

class Stack:

def \_\_init\_\_(self,size=10):

self.stack=[]

def push(self,obj):

self.stack.append(obj)

def pop(self):

try:

return self.stack.pop()

except IndexError as e:

print("Stack is Empty")

def \_\_str\_\_(self):

return str(self.stack)

def main():

stack=Stack()

print("1~10入栈:")

for i in range(1,11):

stack.push(i)

print(stack)

print("1~10出栈:")

for i in range(1,11):

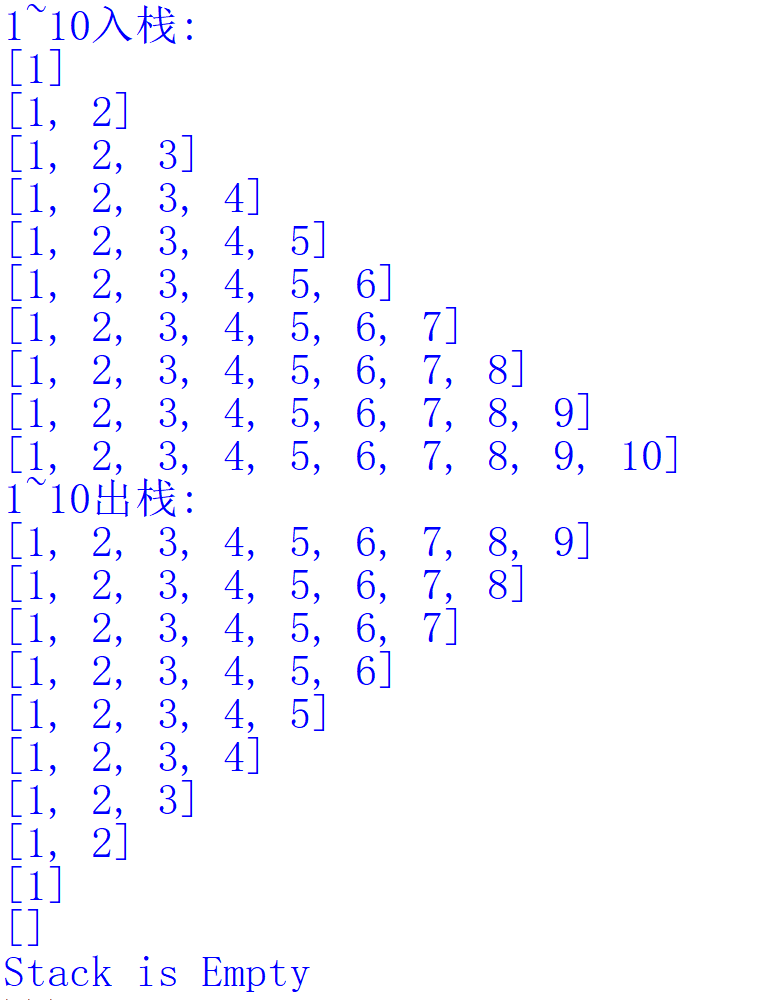
stack.pop()

print(stack)

stack.pop()

if \_\_name\_\_=='\_\_main\_\_':main()

**输出结果：**



**第七题**：创建由’Monday’~’Sunday’7个值组成的字典，输出键列表、值列表和键值列表。

**源代码**：

d={1:'Mon',2:'Tues',3:'Wed',4:'Thur',5:'Fri',6:'Sat',7:'Sun'}

for i in d.keys():

print(i,end=" ")

print("\n")

for i in d.values():

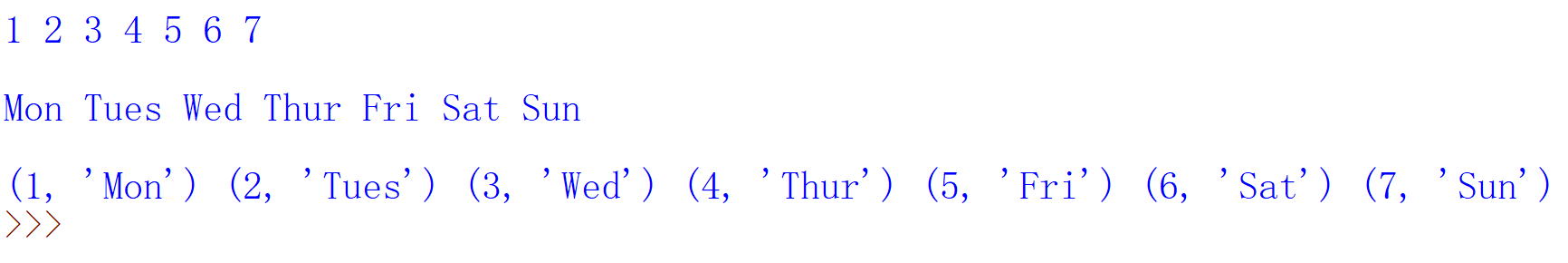
print(i,end=" ")

print("\n")

for i in d.items():

print(i,end=" ")

**输出结果：**



**第七题**：随机生成10个0到10的整数，分别组成集合A和B，输出A和B的内容、长度、最大值和最小值以及它们的并集、交集和差集。

**源代码**：

import random

a=set();b=set()

temp=random.randrange(0,11)

while len(a)!=temp:

m=random.randrange(0,11)

a.add(m)

while len(b)!=10-temp:

m=random.randrange(0,11)

b.add(m)

print("集合的内容，长度，最大值和最小值分别为：")

print(a,len(a),max(a),min(a))

print(b,len(b),max(b),min(b))

print("A和B的并集，交集和差集分别为：")

print(a|b,a&b,a-b)

**输出结果：**

