课程编号：C0800000012

程序设计基础实验报告



|  |  |  |  |
| --- | --- | --- | --- |
| 姓 名 | 贾敬哲 | 学号 | 20175276 |
| 班 级 | 计算机类Ⅱ1701 | 指 导 教 师 | 李丹程 |
| 实 验 名 称 | 程序设计基础实验 | | |
| 开 设 学 期 | 2017-2018 第 一 学 期 | | |
| 开 设 时 间 | 第 8 周 — — 第 17 周 | | |
| 报 告 日 期 | 2017/11/ | | |
| 评 定 成 绩 |  | 评 定 人 |  |
| 评 定 日 期 |  |

东北大学软件学院

实验五 查找与排序程序设计

1. 实验目的

掌握查找与排序算法；编辑和运行 Python 程序。实验要求采用查找与排序算法进行程序设计，给出程序设计流程图或伪代码，并使用 Python 代码编程实现。

1. 问题分析与程序设计

第一题：

问题分析：将学生的姓名，三门课的成绩，总分和平均分设为6个平行数组。用排序算法按总分进行排序。更改索引值时，同时更改6个平行数组的索引值。完成对学生成绩的排序。最后遍历数组，打印结果。

伪代码：

Main

Write "The program is used to count the three grades of five students and to rank them according to the total score"

Declare count As Integer

Declare name[6] As String

Declare course1[6] As Float

Declare course2[6] As Float

Declare course3[6] As Float

Declare Total\_grade[6] As Float

Declare average\_grade[6] As Float

For (count=1;count<=5;count++)

Write "Please enter the studen"\t"+count+" information"

Write "Please enter student's name."

Input name[count]

Write "Please enter the results of the course1."

Input course1[count]

Write "Please enter the results of the course2."

Input course2[count]

Write "Please enter the results of the course3."

Input course3[count]

Set Total\_grade[count]=course1[count]+course2[count]+course3[count]

Set average\_grade[count]=Total\_grade[count]/3

End For(count)

Call bubble(name,course1,course2,course3,Total\_grade,average\_grade)

Call output(name,course1,course2,course3,Total\_grade,average\_grade)

End program

Subprogrammer bubble(name,course1,course2,course3,Total\_grade,average\_grade)

Declare temp As String

Declare temp\_temp As Float

Declare i As Integer

Declare listlength As Integer

listlength = Length(Total\_grade)-1

While (listlength>0)

For(i=1;i<=listlength-1;i++)

If (Total\_grade[i+1]<Total\_grade[i]) Then

Set temp=name[i]

Set name[i]=name[i+1]

Set name[i+1]=temp

//change name`s index

Set temp\_temp =Total\_grade[i]

Set Total\_grade[i]=Total\_grade[i+1]

Set Total\_grade[i+1]=temp\_temp

//change Total\_grade`s index

Set temp\_temp =course1[i]

Set course1[i]=course1[i+1]

Set course1[i+1]=temp\_temp

//change course1 index

Set temp\_temp =course2[i]

Set course2[i]=course2[i+1]

Set course2[i+1]=temp\_temp

//change course2 index

Set temp\_temp=course3[i]

Set course3[i]=course3[i+1]

Set course3[i+1]=temp\_temp

// change course3 index

Set temp\_temp =average\_grade[i]

Set average\_grade[i]=average\_grade[i+1]

Set average\_grade[i+1]=temp\_temp

//change average\_grade index

listlength=listlength-1

End Subprogrammer

Subprogrammer output(name,course1,course2,course3,Total\_grade,average\_grade)

Print "name\tcourse1\tcourse2\tcourse3\tTotal\_grade\taverage\_grade"

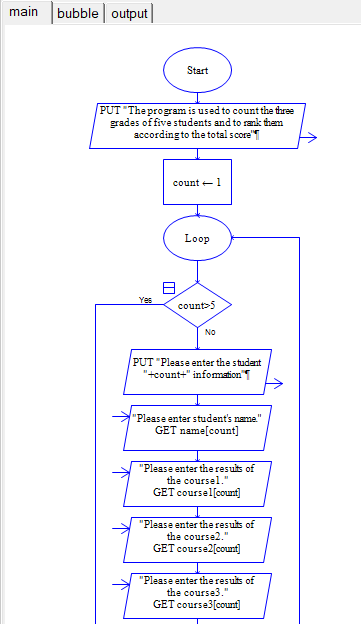
For (i=1;i<6;i++)

Print(name[i]+"\t"+course1[i]+"\t"+course2[i]+"\t"+course3[i]+"\t"+Total\_grade[i]+"\t"+average\_grade[i])

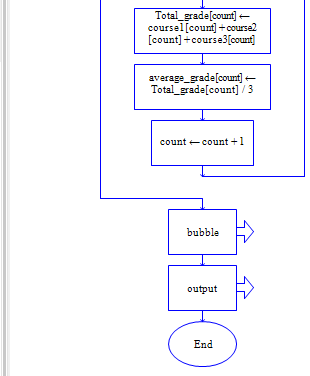
End Subprogrammer

流程图：

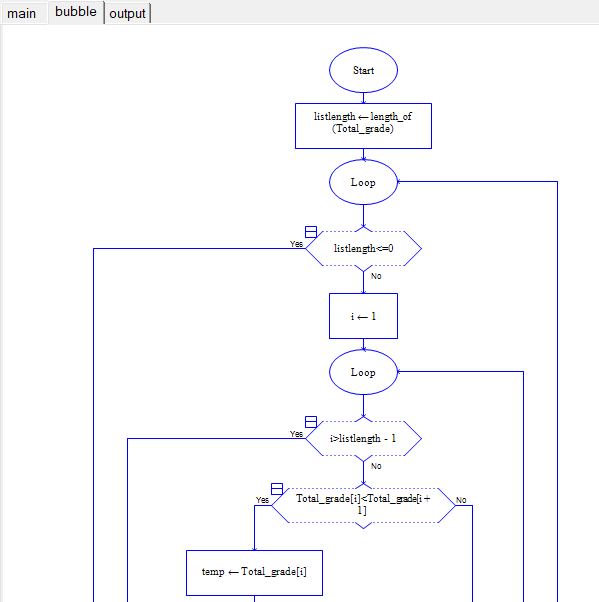
主程序：

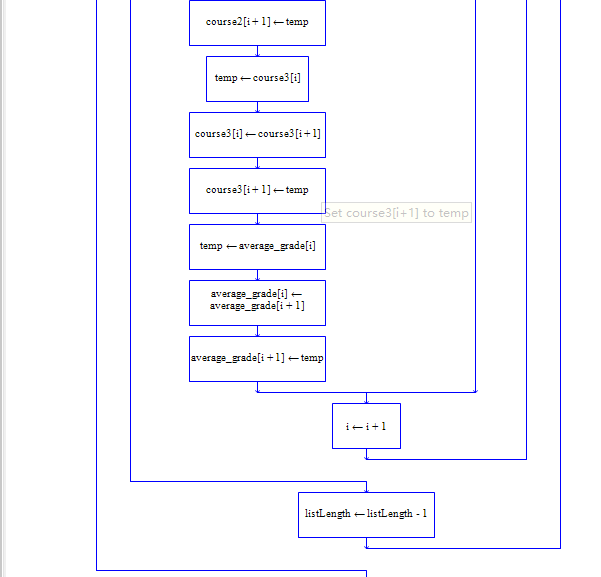
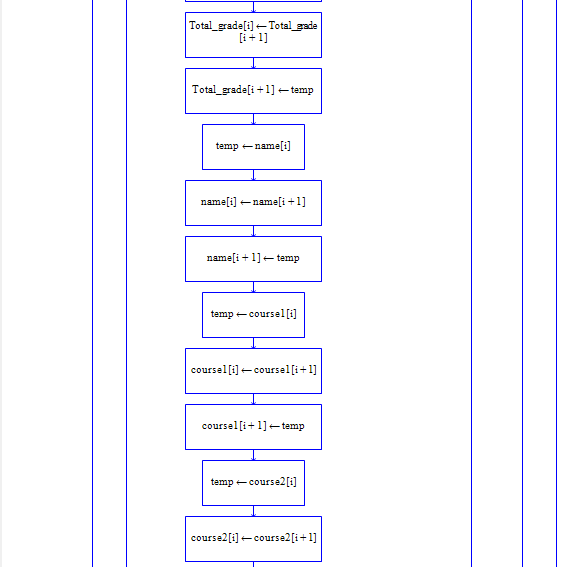


冒泡排序：



输出模块：







第二题：

问题分析：通过input函数输入15个数字，放入数组中储存。用排序算法对数组排序，然后利用折半查找法找出数组的索引值。

伪代码：

Main

Declare list[15] As Float

Declare i As Integer

Write "The program is used to find the value of the number of elements in the array"

For (i=0;i<=15;i++)

Write "Please enter number "+i

Input list[i]

End For(i)

Write "Please enter the number that you want to find"

Input num

Call bubble(list As Ref)

Call binary\_chop(num,list As Ref)

End Program

Subprogrammer bubble (list As Ref)

Declare temp As Integer

Declare listlength As Integer

Declare i As Integer

Set listlength = Length(list)-1

While (listlength>0)

For(i=1;i<=listlength-1;i++)

If (list[i]>list[i+1]) Then

Set temp=list[i]

Set list[i]=list[i+1]

Set list[i+1]=temp

Set listlength=listlength-1

End While

End Subprogrammer

Subprogrammer binary\_chop(num,list As Ref)

Declare Min As Integer

Declare Max As Integer

Declare found As Boolean

Declare flag As Integer

Set Min =0

Set Min =14

Set found=Flase

While (Max>Min And found=Flase)

Set flag=Floor((Max+Min)/2) //向下取整

If num>list[flag] Then

Set Min=flag+1

Else

If num=list[flag] Then

found=1

Else

Max=flag-1

End If

End If

End While

If found==1 Then

Call chachong(list,flag)

Else

Write "No number"

End Subprogrammer

Subprogrammer chachong(list,flag)

Declare left As Integer

Declare right As Integer

Declare temp As Integer

Set temp=flag

While (list[flag]==list[flag+1] And flag<14)

Set flag=flag+1

End While

Set right= flag

While (list[temp]==list[temp-1] And temp>0)

Set temp=temp-1

End While

Set left = temp

Write "The index value of the number in the array is"

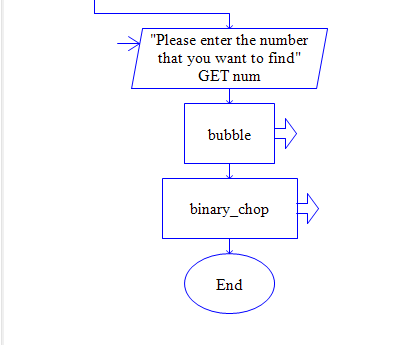
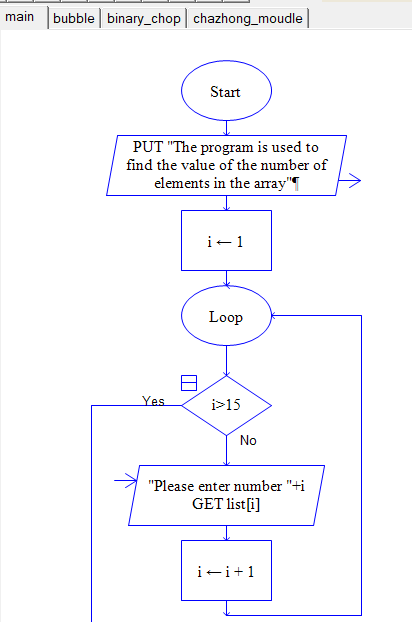
For (left<=ringt;left++)

Write left

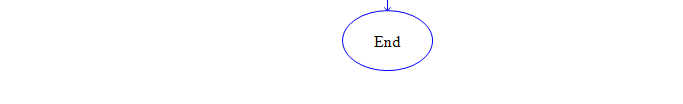
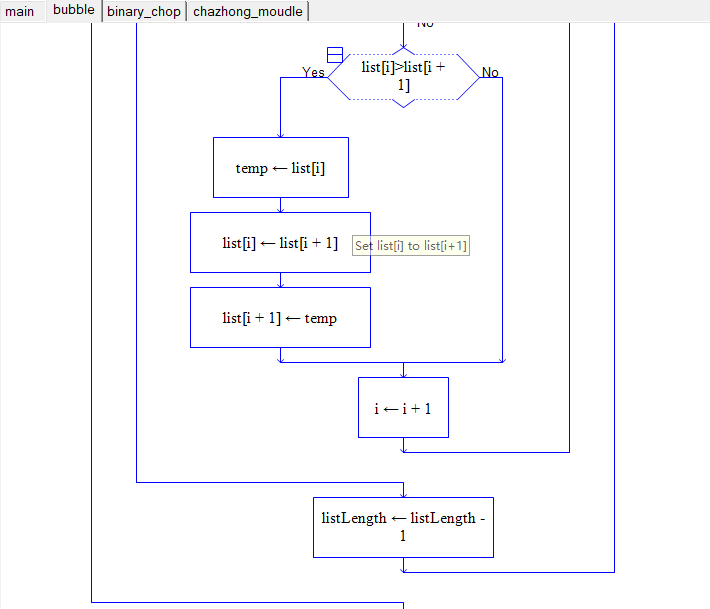
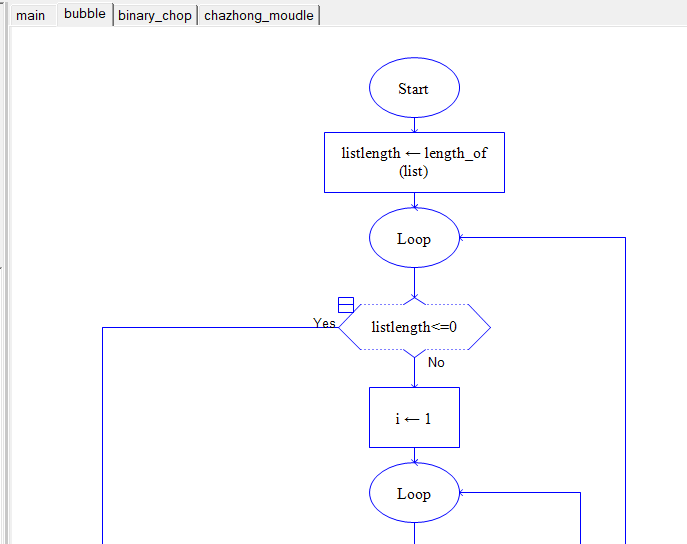
End For(left)

End Subprogrammer

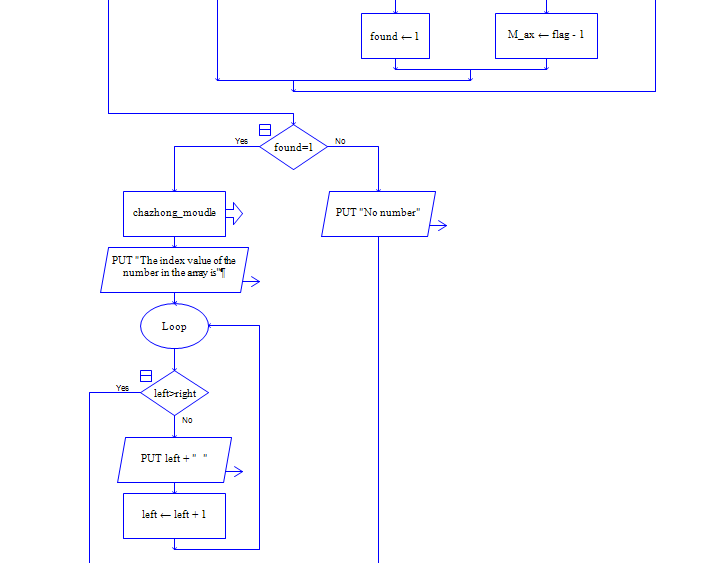
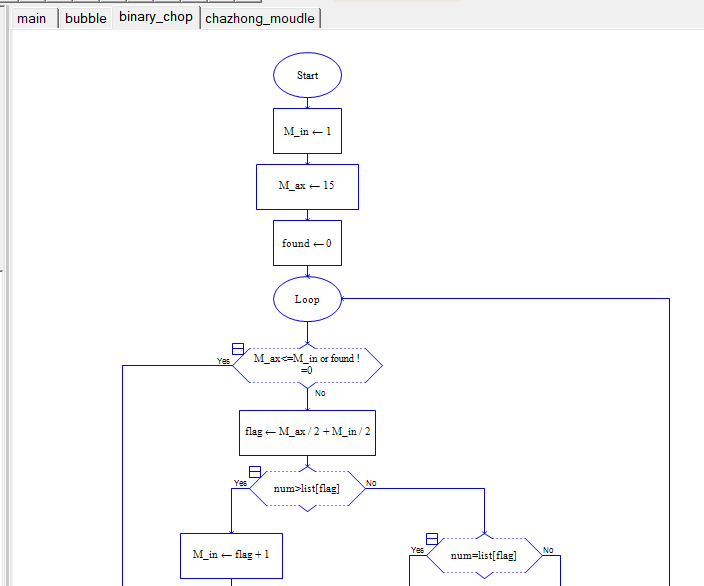
流程图：主程序：

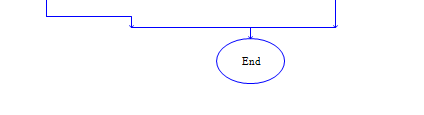


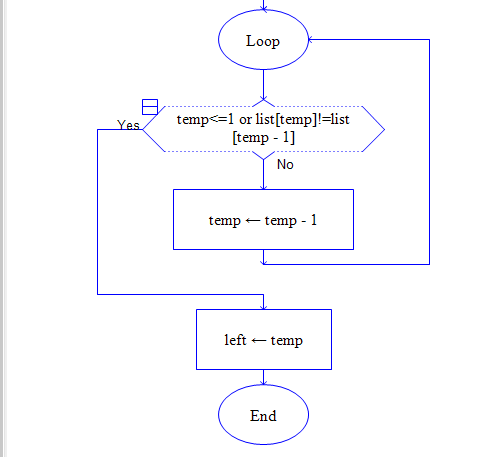
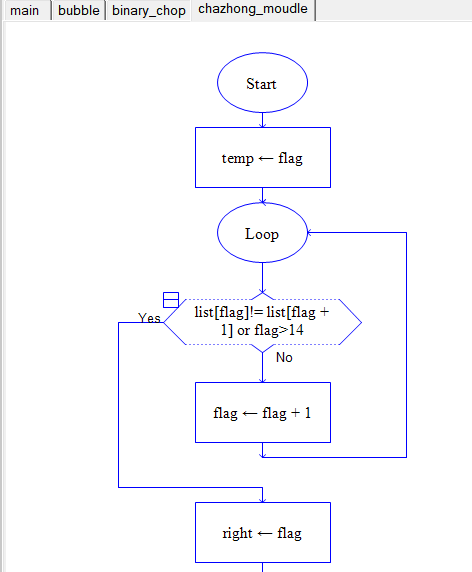
冒泡排序：



二分查找：





查重模块：

1. 实现过程与测试结果分析

第一题：

Python代码：

def change\_index(list,index1):#交换索引

list[index1],list[index1+1]=list[index1+1],list[index1]

return list

def bubble():#冒泡排序

name,course1,course2,course3,Total\_grade,average\_grade=input\_moudle()

listLength = len(Total\_grade)

while listLength > 0:

for i in range(listLength - 1):

if Total\_grade[i]<Total\_grade [i+1]:

Total\_grade=change\_index(Total\_grade,i)

name=change\_index(name,i)

course1=change\_index(course1,i)

course2=change\_index(course2,i)

course3=change\_index(course3,i)

average\_grade=change\_index(average\_grade,i)

listLength -= 1

return name,course1,course2,course3,Total\_grade,average\_grade

def input\_moudle():

name,course1,course2,course3,Total\_grade,average\_grade=[],[],[],[],[],[]

print("该程序用来统计五名学生的三门成绩，并根据总分进行排序")

for i in range (5):

print("请输入第",i+1,"个学生的信息")

name.append(input("请输入学生的姓名"))

course1.append(eval(input("请输入课程一的成绩")))

course2.append(eval(input("请输入课程二的成绩")))

course3.append(eval(input("请输入课程三的成绩")))

Total\_grade.append(course1[i]+course2[i]+course3[i])

average\_grade.append(Total\_grade[i]/3)

return name,course1,course2,course3,Total\_grade,average\_grade

def output\_moudle():

name,course1,course2,course3,Total\_grade,average\_grade=bubble()

print("名字\t课程一 课程二 课程三 总成绩 平均成绩")

for i in range(5):

print(name[i]," \t",course1[i],"\t",course2[i],"\t",course3[i],"\t",Total\_grade[i],"\t",average\_grade[i])

output\_moudle()

第二题：

Python代码：

#quick sort

def quickSort(L, low, high):

i = low

j = high

if i >= j:

return L

key = L[i]

while i < j:

while i < j and L[j] >= key:

j = j-1

L[i] = L[j]

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

i = i+1

L[j] = L[i]

L[i] = key

quickSort(L, low, i-1)

quickSort(L, j+1, high)

return L

List=[]

for i in range(15):

print("请输入数组中的第",i+1,"个数:")

List.append(eval(input()))

quickSort(List,0,14)

result=[]

print("退出请直接按回车键")

while True:

try:

x=eval(input("请输入需要查找的数:"))

Min,Max,flag,found=0,14,0,0

while (Max>=Min) and (found==0):

flag=int((Max+Min)/2)

if x>List[flag]:

Min=flag+1

elif x == List[flag]:

found=1

else:

Max=flag-1

if found== 1:

temp=flag#设立中间值

while (List[flag]==List[flag+1])and (flag<=13):#对列表的右方检索相同的数

flag=flag+1

if flag==14:

break

ringht=flag

while(List[temp]==List[temp-1]) and (temp>=1):#对列表的左方检索

temp-=1

if temp==0:

break

left=temp

result=[x for x in range(left,ringht+1) ]

print("该数在数组的索引值为 ",result)

else:

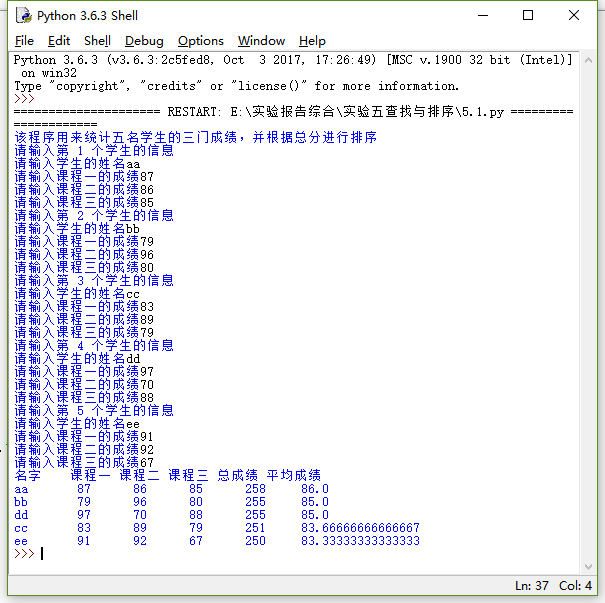
print("无此数")

except:

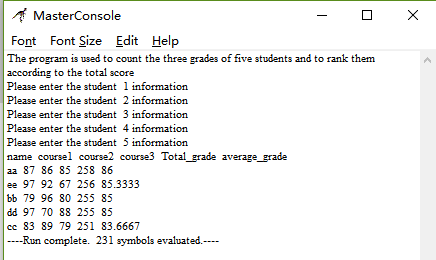
break

1. 实验结果总结

第一题：

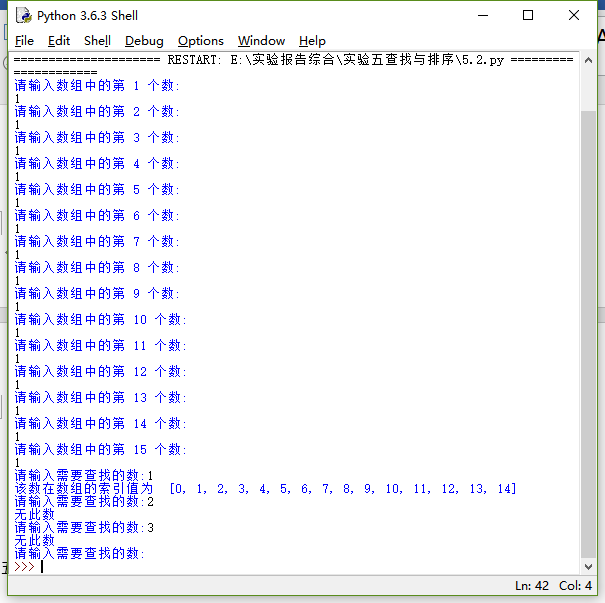
Python代码结果：

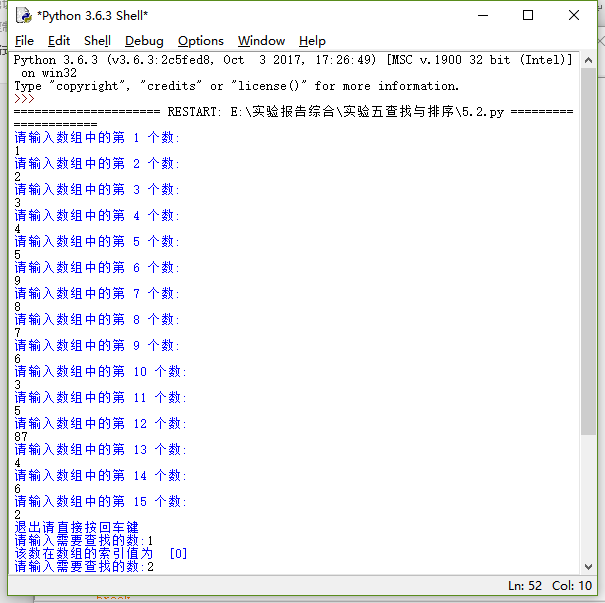
流程图结果：

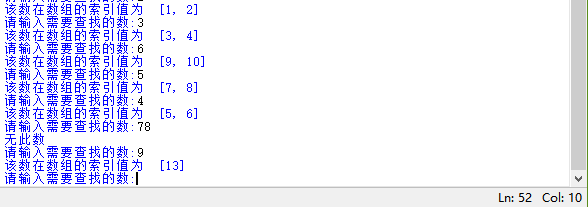


第二题：

Python代码结果：

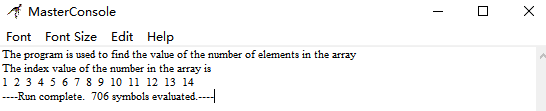






流程图结果：

1. 数字全部为1



1. 随机数字



1. 创新的部分

第一题中：Python代码应用了效率更高的快速排序，并且使用了制表符，使结果更加美观。

第二题中：Python代码和Raptor都使用了查重算法，使得结果更加精确。

1. 对实验的意见与建议

通过本次实验，我对查找和排序算法更加熟练，自己的能力也得到了很大的提高。尤其是额外练习了其他的排序和查找算法。

七、实验思考题

1. 如何提高查找与排序的执行效率？

换用效率更高的算法，在执行算法钱，对数据进行一定的处理，使得需要处理的数据数量减少，从而提高算法的执行速度。

2. 排序算法通常使用哪种数据结构？

通常使用数组，以及栈、队列等等

实验六 模块化程序设计

1. 实验目的

掌握模块化程序设计方法；编辑和运行 Python 程序。实验要求采用模块化程序设计方法，给出程序设计流程图或伪代码，并使用 Python 代码编程实现

1. 问题分析与程序设计

第一题：

问题分析：输入一组正数，储存在数组中，对数组中的元素进行求和，然后计算平均值，对数组的元素遍历，得到符合条件的数量，输出结果

伪代码：

Declare i As Integer

Declare l List[] As Integer

Declare sum As Integer

Declare average As Float

Declare num As Integer

Write "Please enter the number of the set of positive numbers"

Input i

Set sum=0

For (;i>=1;i--)

Write "Please enter the "+ i+"th number"

Input List[i]

Set sum=sum+List[i]

Call average\_moudle(i,sum,average As ref)

Call judgement(List,average,i,num As ref)

Call output\_moudle(average,num)

End progremmer

Subprogrammer average\_moudle(i,sum,average As ref)

Set average=sum/i

End Subprogrammer

Subprogrammer judgement(List,average,i,num As ref)

Set num =0

For (;i>=1;i--)

If List[i]>=average Then

num=num+1

End If

End Subprogrammer

Subprogrammer output\_moudle(average,num)

Write "The average is "+average

Write "The number of more than or equal to the average is "+num

End Subprogrammer

第二题：

问题分析：对两个整数进行判断，确定大小顺序，然后用循环结构，得出最大公约数的结果。

伪代码：

Main

Declare num1 As Integer

Declare num2 As Integer

Write "Please enter the first integer"

Input num1

Write "Please enter the second integer"

Input num2

Call change\_moudle(num1 As Ref,num2 As Ref)

Call Common\_divisor(num1.num2)

End programmer

Subprogremmer change\_moudle(num1 As Ref,num2 As Ref)

Declare temp As Integer

If num1<num2 Then

Set temp=num2

Set num1 = num2

Set num2=temp

End If

End Subprogremmer

Subprogremmer Common\_divisor(num1,num2)

Declare res As Integer

Set res=1

While res!=0

Set res=num1%num2

Set num1=num2

Set num2 =res

Write "最大公约数是 "+res

End Subprogremmer

第三题：

问题分析：设置菜单程序，输入不同的数字输出不同的结果，编辑三个模块，分别对应三个Area值。

伪代码：

Main

Declare x As Float

Declare flag As Integer

Write "请输入数值X："

Input x

Write "操作\t按钮\n正方形面积:\t1\n圆面积:\t2\n等边三角形面积:\t3\n"

Input flag

If flag==1 Then

Call square(x)

Else

If flag==2

Call circle(x)

Else

Call equilateral\_triangle(x)

End If

End If

End programmer

Subprogrammer square(x)

Declare res As Float

Set res=x\*x

Write "正方形面积为 "+res

End Subprogrammer

Subprogrammer circle(x)

Declare res As Float

Set res=3.14\*x\*x

Write "圆的面积为 "+res

End Subprogrammer

Subprogrammer equilateral\_triangle(x)

Declare res As Float

Set res=sqrt(3)\*x\*x

Write "正三角方形面积为 "+res

End Subprogrammer

1. 实现过程与测试结果分析

第一题Python代码：

def average\_moudle(List):

sum\_num=sum(List)

average=sum\_num/len(List)

return average

def judgement(list):

count=0

average=average\_moudle(List)

for i in List:

if i >= average:

count+=1

return count

def output\_moudle(List):

average=average\_moudle(List)

count=judgement(List)

print("平均数是 ",average)

print("大于等于平均值的个数是 ",count)

num=input("输入一组数据，用空格隔开：\n")

List=num.split(' ')

List=list(map(int,List))

output\_moudle(List)

第二题Pythondef change\_num(a,b):

if a <b:

a,b=b,a

num1=eval(input("请输入一个整数： "))

num2=eval(input("请输入第二个整数： "))

change\_num(num1,num2)

a,b=num1,num2

r=1

while r !=0:

r=a%b

a=b

b=r

print("最大公约数为：",a)代码：

第三题Python代码：

def square(x):

return x\*x

def circle(x):

return 3.14\*x\*x

def equilateral\_triangle(x):

import math

return math.sqrt(3)\*x\*x

x=eval(input("请输入数值X： "))

print("操作\t按钮\n正方形面积:\t1\n圆面积:\t2\n等边三角形面积:\t3\n")

flag=eval(input("请输入需要的操作"))

if flag==1:

A=square(x)

elif flag==2:

A=circle(x)

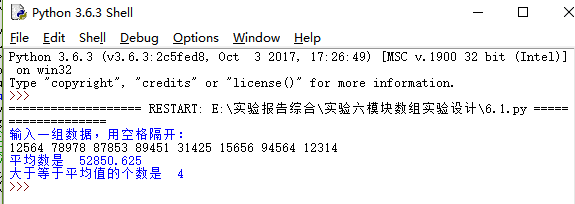
else :

A=equilateral\_triangle(x)

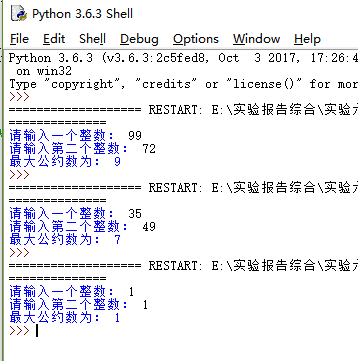
print("面积为",A)

1. 实验结果总结

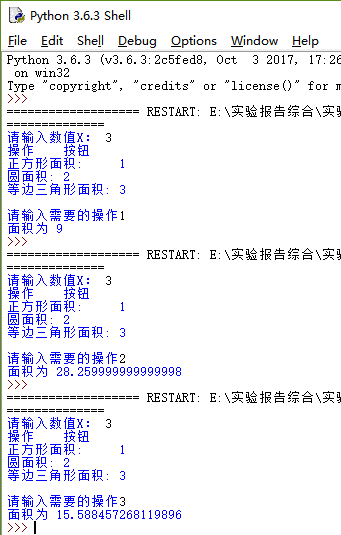
第一题：



第二题：



第三题：



1. 创新的部分

第一题中，输入数组时，可以同时输入多个数字

1. 对实验的意见与建议

通过本次实验，我熟练地掌握了模块化的程序设计，体会到模块化所带来的逻辑清晰和便捷，。

1. 思考题
2. 模块化程序设计的特点？

特点是模块之间相互独立，这样降低程序复杂度，使程序设计、调试和维护等操作简单化。

1. 模块是如何划分的？

在设计较复杂的程序时，一般采用自顶向下的方法，将问题划分为几个部分，各个部分再进行细化，直到分解为较好解决问题为止。

实验七 数据文件程序设计

1. 实验目的

了解文件操作的程序设计方法；编辑和运行 Python 程序。实验要求采用数据文件进行程序设计，学会使用文件打开、关闭、读、写等文件操作，给出程序设计流程图或伪代码，并使用 Python 代码编程实现

1. 问题分析与程序设计

通过对实验题目中给出的问题的详细分析，绘制程序设计的流程图或者写出伪代码，并

对流程图进行相关的注释。

第一题：

Declare i As Integer

Open "name" For Input As Datafile

Write "请输入总人数"

Input num

For (i=1;i<=num;i++)

Write "请输入客户的姓:"

Input first\_name

Set name[num]=first\_name

Write "请输入客户的名："

Input last\_name

Write Datafile,first\_name, last\_name

End For(i)

Close Datafile

Set remove same name//去除name数组中重复的部分

Open 'name' For Output As Datafile

While NOT EOF(Datafile)

Read Datafile, first\_name,last\_name

Write first\_name+last\_name

Add first\_name To name[]//将姓名添加name这个数组中

End While

Open 'name' For Output As Datafile

For (i=1;i<=num;i++)

res=0

While NOT EOF (Datafile)

Read Datafile, first\_name,last\_name

If first\_name==name Then

res=res+1

Write "姓为"+first\_name+"共有"+res+"人"

End While

End For(i)

1. 实现过程与测试结果分析

file=open('name.txt','w')

print("停止输入时，请按回车")

name=[]

while True:

first\_name=input("请输入用户的姓：")

last\_name=input("请输入用户的名：")

file.write(first\_name+last\_name+'\n')#写入文档

if first\_name=='':

break

file.close()

file=open('name.txt','r')

data=file.readlines()

for i in data:

print(i,end='')

name.append(i[0])

l2=list(set(name))

for i in l2:

if i !='\n':

res=0

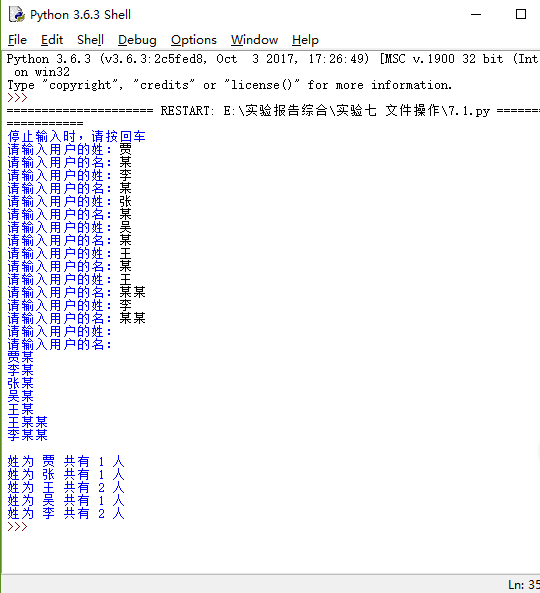
for j in data:

if i == j[0]:

res=res+1

print("姓为",i[0],"共有",res,"人")

1. 实验结果总结



1. 创新的部分

使用了Python代码中的数组去重函数

1. 对实验的意见与建议

通过这次实验，我熟练地掌握了关于数据文件的处理，这极大地增强了我的能力，让我可以通过程序保存数据，并对类似txt的文件进行处理，让我们可以做更多的事情。

实验八 面向对象程序设计

1. 实验目的

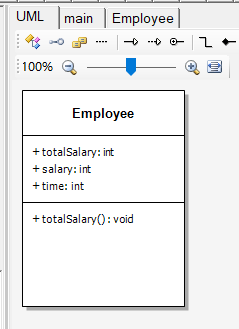
掌握面向对象程序设计思想，掌握类图的绘制方法，编辑和运行 Python 程序。实验要求采用面向对象程序设计方法，给出程序设计流程图或伪代码，并使用 Python 代码编程实现。

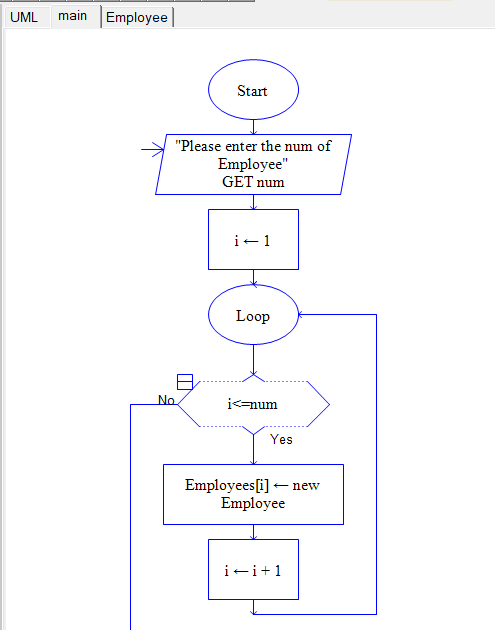
1. 问题分析与程序设计

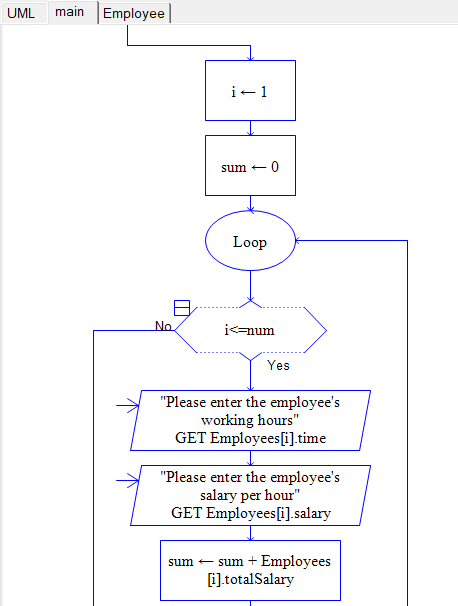
第一题：

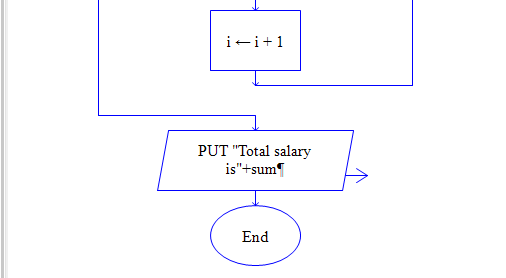
问题分析： 创造一个员工的类，每次输入时，都创造一个实例，通过方法得出totalSalary，并对totalSalary进行求和，最终输出公司应付的总钱数。

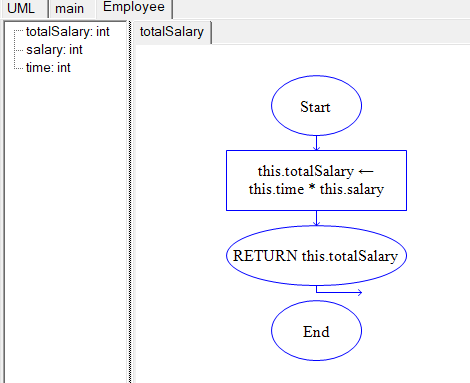
流程图：











1. 实现过程与测试结果分析

Python 代码：

class employee():

def \_\_init\_\_(self,time,salary):

self.time=time

self.salary=salary

def Total\_salary(self):#计算总工资

return self.time\*self.salary

def set\_time(self,time):

self.time=time

def set\_salary(self,salary):

self.salary=salary

def set\_award(self,award):

self.award=award

def main():

num\_of\_employee=eval(input("请输入员工的数量："))

salary=[]

for i in range(num\_of\_employee):

print("请输入第",i+1,'个员工')

Employee=employee(eval(input("请输入该员工工作时间：")),eval(input("请输入该员工每小时的工资：")))

result=Employee.Total\_salary()

salary.append(result)

print("信息统计完成")

sum\_of\_salary=sum(salary)#计算总工资

print('雇员的总工资为',sum\_of\_salary)

average\_of\_salary=sum\_of\_salary/num\_of\_employee

count=1

for i in salary:

if i >average\_of\_salary:

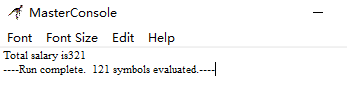
print("给予第",count,'个员工证书奖励')

count+=1

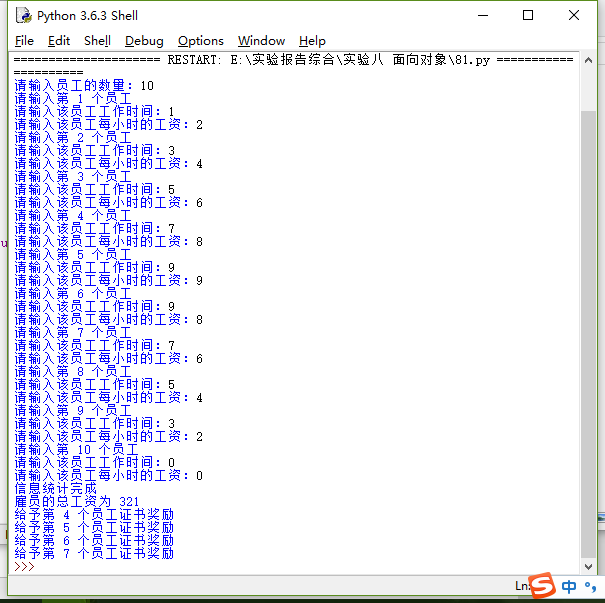
main()

1. 实验结果总结

流程图结果：



Python代码结果：



1. 创新的部分

添加了对员工的证书奖励 的部分

1. 对实验的意见与建议

通过本次实验，我简单地掌握了Python中对类的操作，和类的基本概念。

评价表格（每份实验报告只需一份评分表）

|  |  |
| --- | --- |
| 考核标准 | 得分 |
| （1）正确理解和掌握实验所涉及的概念和原理（20%）； |  |
| （2）按实验要求合理设计程序执行流程（20%）； |  |
| （3）能编程实现设计的程序流程，运行结果正确（20%）； |  |
| （4）认真记录实验数据，原理及实验结果分析准确（20%）； |  |
| （5）实验过程中，具有严谨的学习态度和认真、踏实、一丝不苟的科学作风（5%）； |  |
| （6）所做实验具有一定的创新性（5%）； |  |
| （7）实验报告规范（10%）。 |  |