

2023-2024 学年 02 学期

潍坊学院计算机工程学院

课程报告（B）

课程名称: B022211 《Python 程序设计》

课程类型: 专业选修（考查）

学生姓名: 樊宗豪

学 号: 22022402430

专业班级: 数据科学与大数据技术 2 班

课程教师: 张辉辉

2024 年 05 月

一、任务与具体要求

依据教学大纲相关要求，本课程报告设计一系列题目实现对 Python 教学内容的考察，学生针对预设问题给出自己的设计作答，并通过撰写报告的形式进行展示。

二、任务设计与教材对应关系

本课程报告的设计题目涵盖 Python 程序设计的基础内容（如 Python 数据类型、函数设计、模块设计、异常处理等），具体的考察内容和章节对应关系见下表。

表 1 设计题目与章节对应关系

题目设计	考察内容	对应章节
题目 1	Python 语言基础	第一、二章
题目 2	Python 流程控制	第三章
题目 3	Python 函数设计	第四章
题目 4	Python 内置组合数据	第五章
题目 5	Python 异常与文件处理	第六章
题目 6	Python 数值处理	第七章
题目 7	Python 字符串和文本处理	第八章
题目 8	Python 面向对象	第九章
题目 9	Python 模块设计	第十章
题目 10	课程学习心得体会	全课程

三、内容解答及提交要求

1. Python 源代码;
2. 课程报告的电子版 (pdf 格式);
3. 课程报告纸质打印版;
4. 针对每一个设计问题, 给出解答明细, 例如源代码、测试结果。

三、成绩评定

指导教师 (签字) _____

2024 年 06 月 12 日

目录

1 Python 语言基础	1
1.1 程序阅读分析	1
1.2 程序阅读分析	2
1.3 程序阅读分析	2
2 Python 流程控制	3
2.1 程序分析	3
2.2 程序设计	5
2.3 程序设计	6
3 Python 函数设计	8
3.1 问题求解一	8
3.2 问题求解二	8
3.3 问题求解三	9
4 Python 内置组合数据	12
4.1 问题求解一	12
4.2 问题求解二	12
5 Python 文件处理	14
6 Python 数值处理	16
7 Python 字符串和文本处理	18
7.1 问题求解	18
7.2 程序分析	19
8 Python 面向对象设计	20
9 Python 模块设计	22
心得体会	23

1 Python 语言基础

本题共计 10 分：第一小题 5 分、第二小题 2 分、第三小题 3 分。

1.1 程序阅读分析

阅读如下 Python 程序并给出分析结果。

```
#阅读下面Python程序，请问输出结果是什么？
print("T",end=' ') if not 0 else print('F',end=' ')
print("T",end=' ') if 6 else print('F',end=' ')
print("T",end=' ') if "" else print('F',end=' ')
print("T",end=' ') if "abc" else print('F',end=' ')
print("T",end=' ') if () else print('F',end=' ')
print("T",end=' ') if (1,2) else print('F',end=' ')
print("T",end=' ') if [] else print('F',end=' ')
print("T",end=' ') if [1,2] else print('F',end=' ')
print("T",end=' ') if {} else print('F',end=' ')
print("T",end=' ') if {1,2} else print('F',end=' ')
```

(1) 上述代码的执行结果 (3 分)

T T F T F T F T F T

(2) 原因分析 (2 分)

1. not 0 的结果是 True，因为 0 在 Python 中被视为假，所以 not 0 是真。
2. 6 是一个非零数值，所以它被视为真。
3. "" 是一个空字符串，所以它被视为假。
4. "abc" 是一个非空字符串，所以它被视为真。
5. () 是一个空元组，所以它被视为假。
6. (1,2) 是一个非空元组，所以它被视为真。
7. [] 是一个空列表，所以它被视为假。
8. [1,2] 是一个非空列表，所以它被视为真。
9. {} 是一个空字典，所以它被视为假。
10. {1,2} 是一个非空集合，所以它被视为真。

因此，根据每个条件表达式的真假，程序会相应地打印出 “T” 或 “F”。

1.2 程序阅读分析

阅读如下的程序代码，给出程序执行结果。

```
# 下列Python语句的程序运行结果为。  
x = True; y = False; z = True;  
if not x or y: print(1)  
elif not x or not y and z: print(2)  
elif not x or y or not y and x: print(3)  
else: print(4)
```

程序执行结果：

2

1.3 程序阅读分析

假设有 $a=10$ ，写出下面表达式运算后 a 的值：

- (1) $a += a$
- (2) $a -= 2$
- (3) $a *= 2 + 3$
- (4) $a /= 2 + 3$
- (5) $a \% = a - a \% 4$
- (6) $a //= a - 3$

答案：

1	2	3	4	5	6
20	8	50	2.0	2	1

2 Python 流程控制

本题共计 17 分：第一小题 9 分，第二小题 3 分，第三小题 5 分。

2.1 程序分析

说明以下 3 个 if 语句的区别：给出每个 if 对应的分段函数表达式，给出相应的程序执行流程图。

```
(1) if (i > 0):
    if (j > 0):
        n = 1
    else:
        n = 2
```

```
(2) if (i > 0):
    if (j > 0):
        n = 1
    else:
        n = 2
```

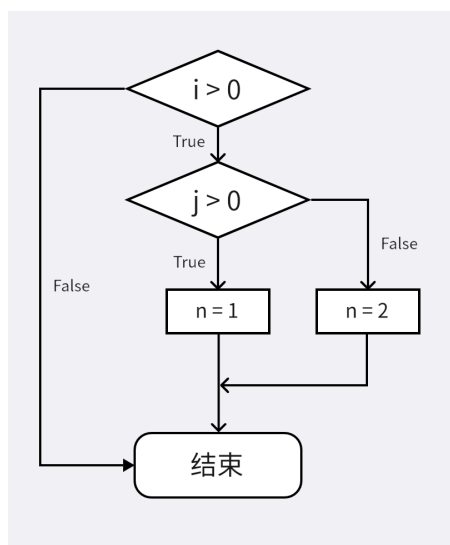
```
(3) if (i > 0):
    n = 1
else:
    if (j > 0):
        n = 2
```

程序分析解答：

(1) if 逻辑的计算公式 (1 分)：

$$n = \begin{cases} 1, & i > 0 \text{ and } j > 0 \\ 2, & i > 0 \text{ and } j \leq 0 \end{cases}$$

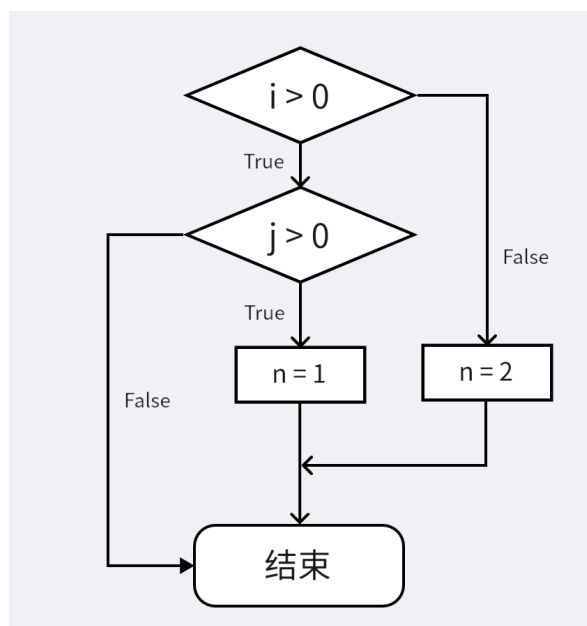
程序执行流程图 (2 分):



(2) if 逻辑的计算公式 (1 分):

$$n = \begin{cases} 1, & i > 0 \text{ and } j > 0 \\ 2, & i \leq 0 \end{cases}$$

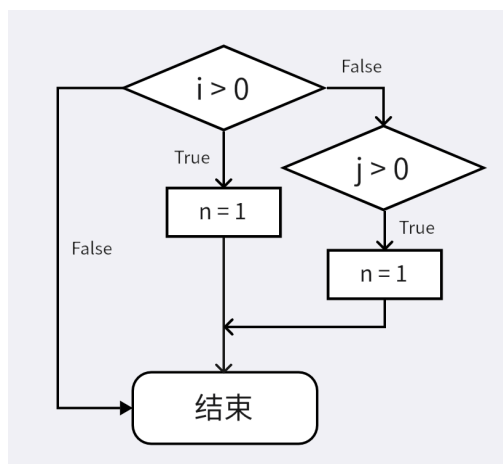
程序执行流程图 (2 分):



(3) if 逻辑计算公式 (1 分):

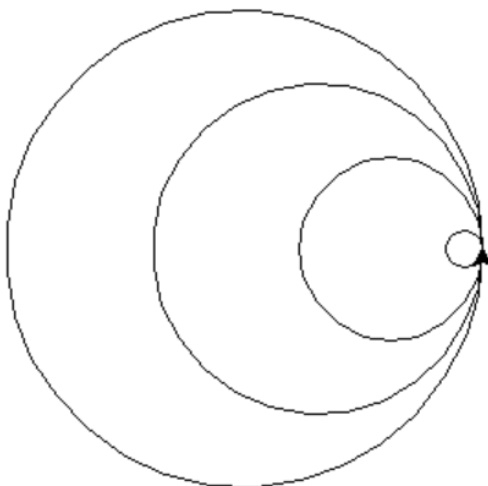
$$n = \begin{cases} 1, & i > 0 \\ 2, & i \leq 0 \text{ and } j > 0 \end{cases}$$

程序执行流程图 (2 分):



2.2 程序设计

利用循环结构, 使用 turtle 库的 `turtle.circle()` 函数和 `turtle.seth()` 函数绘制套圈, 最小的圆圈半径为 10 像素, 不同圆圈之间的半径差是 40 像素。程序运行效果如图所示。



实现代码 (3 分):

```
import turtle as t
```

```
r = 10
head = 90
t.seth(head)
for i in range(4):
    t.circle(r + i * 40)
t.done()
```

2.3 程序设计

(5 分)

编写程序，输入两个整数 a 和 b ，利用“更相减损法”求这两个整数的最大公约数和最小公倍数。运行结果如下图所示。

```
请输入整数a: 12
请输入整数b: 3
最大公约数: 3
最小公倍数: 12
```

(1) “更相减损法”出自于中国古代的《九章算术》，也是一种求最大公约数的算法，具体算法如下：

①先判断两个正整数的大小，如果两数相等，则这个数本身就是所求的最大公约数。

②如果两个正整数不相等，则用大数减去小数，然后用这个较小数与它们相减的结果相比较，如果相等，则这个差就是两个正整数的最大公约数，而如果不相等，则继续执行②操作。

(2) 求得了最大公约数后，最小公倍数就是已知的两个正整数之积除以最大公约数的商。

代码实现 (3 分):

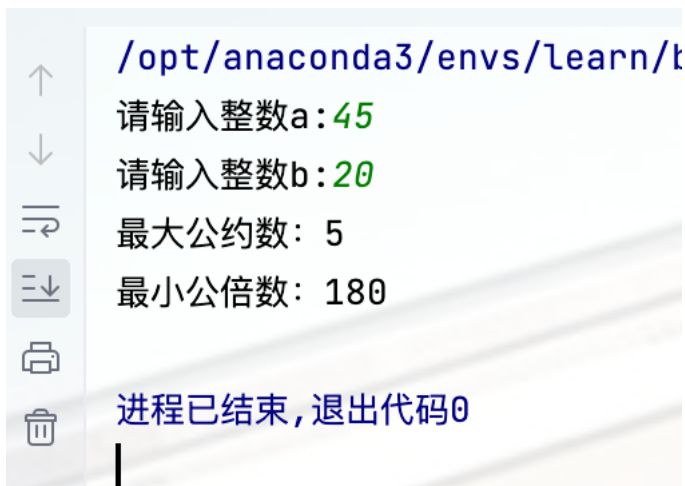
```
def gcd(a, b):
    if a == b: return a
    sub = max(a, b) - min(a, b)
    while sub != min(a, b):
```

```
a, b = min(a, b), sub
sub = max(a, b) - min(a, b)
return sub

def lcm(a, b):
    return a * b // gcd(a, b)

a = int(input('请输入整数 a:'))
b = int(input('请输入整数 b:'))
print(f"最大公约数: {gcd(a, b)}")
print(f"最小公倍数: {lcm(a, b)}")
```

测试结果 (2 分):



```
/opt/anaconda3/envs/learn/t
请输入整数a:45
请输入整数b:20
最大公约数: 5
最小公倍数: 180
进程已结束,退出代码0
```

3 Python 函数设计

本题共计 16 分：第一小题 5 分，第二小题 5 分，第三小题 6 分。

3.1 问题求解一

编写程序，分别定义求两个整数最大公约数的函数 GCD 和求最小公倍数的函数 LCM，并编写测试代码，要求从键盘接收两个整数进行测试，程序运行结果如下图所示。请使用递归方式实现。

```
请输入第一个整数: 12
请输入第二个整数: 15
3 60
```

解答代码:

```
def gcd(a, b):
    return b if a % b == 0 else gcd(b, a % b)

def lcm(a, b):
    return a * b // gcd(a, b)

a = int(input('请输入第一个整数: '))
b = int(input('请输入第二个整数: '))
print(gcd(a, b), lcm(a, b))
```

3.2 问题求解二

(5 分)

按照下列要求，设计完成一个 Python 程序。具体要求如下：

1. 编写函数 `get_birthday`, 参数为身份证号码，根据身份证号码获得这个人的生日，并返回列表 `[year, month, day]`。

2. 主程序中调用函数 `get_birthday`，给出列表对象['310101199005052115', '310104199607076128', '310117199309235133']中的身份证号码的生日。并在主程序中显示并输出。

3. 程序运行结果如下图所示：

```
310101199005052115 的生日为: 1990-5-5
310104199607076128 的生日为: 1996-7-7
310117199309235133 的生日为: 1993-9-23
```

问题求解代码：

```
def get_birthday(id_num: str) -> list[int]:
    year = int(id_num[6:10])
    month = int(id_num[10:12])
    day = int(id_num[12:14])
    return [year, month, day]

id_nums = ['310101199005052115', '310104199607076128',
            '310117199309235133']
for id_num in id_nums:
    print("{}的生日为: {}-{}-{}".format(id_num,
        *get_birthday(id_num)))
```

3.3 问题求解三

(6 分)

问题描述：设计完成一个 Python 程序，具体要求如下。

(1) 程序功能：计算某地区月最大降水量和对应降水月份。

(2) 文本文件 `rainfall.txt` 存放着某些地区某年降水量的数据，每行为一个地区 12 个月的降水量（单位：mm），数据间的分隔符为制表符。内容如下：

地区	1月	2月	3月	4月	5月	6月	7月	8月	9月	10月	11月	12月
上海	52	20	104	60	199	167	158	211	14	92	2	14
北京	0	2	7	5	46	69	196	120	116	10	0	3
天津	0	0	4	13	60	115	216	199	51	44	4	0
喀什	1	0	2	41	3	4	6	1	3	5	0	3
西安	4	1	43	32	22	20	71	24	24	64	8	0
重庆	30	21	21	27	118	225	167	51	77	101	46	39
广州	42	71	78	104	71	219	275	316	168	305	6	5
韶关	67	140	115	136	134	470	128	120	17	128	22	30
海口	36	14	63	37	198	273	252	272	190	313	125	19

(3) 编写函数 `getMaxPos(L)`, 形参 `L` 为一组含有数值型数据的列表 (某地区各月份的降水量), 计算列表中数据的最大值和对应下标, 并将最大值和下标以元组形式 (作为元组的两个元素) 返回。

(4) 主程序读取 `rainfall.txt` 内的数据, 将每行数据中的月降水量转为数值型数据, 利用函数 `getMaxPos(L)` 计算并屏幕输出各地区月最大降水量 (显示宽度 8 位) 和对应降水月份 (显示宽度 6 位), 各数据间以制表符相隔。

(5) 程序运行的结果如下图所示:

地区	最大月降水量(mm)	对应降水月份↵
上海	211	8 月↵
北京	196	7 月↵
天津	216	7 月↵
喀什	41	4 月↵
西安	71	7 月↵
重庆	225	6 月↵
广州	316	8 月↵
韶关	470	6 月↵
海口	313	10 月↵


实现方案代码 (5 分):

```
def getMaxPos(L: list[int]) -> tuple[int, int]:
    return max(L), L.index(max(L)) + 1

def main():
    citys: list
    with open("rainfall.txt", mode='r') as f:
        citys = [[tokens[0]] + [int(i) for i in tokens[1:]] for tokens
in (line.split() for line in f.readlines()[1:])]
    print("地区", "最大月降水量 (mm)", "对应降水月份", sep="\t")
    print(*("{}\t{:8}\t{:6}月".format(city[0], *getMaxPos(city[1:])))
for city in citys), sep="\n")

if __name__ == '__main__':
    main()
```

测试结果 (1 分):



```
运行 main x
/opt/anaconda3/envs/learn/bin/python
地区 最大月降水量 (mm) 对应降水月份
上海 211 8月
北京 196 7月
天津 216 7月
喀什 41 4月
西安 71 7月
重庆 225 6月
广州 316 8月
韶关 470 6月
海口 313 10月

进程已结束,退出代码0
```

4 Python 内置组合数据

本题共计 10 分：第一小题 5 分，第二小题 5 分。

4.1 问题求解一

(5 分)

主程序中已有一个排好序的列表，请编写函数 `insertList`，将从键盘接收的整数按原来从小到大的排序规律插入到该列表中。

```
def insertList(L1,x):
#函数代码

L1=[1,4,6,9,13,16,28,40,100]
x=int(input('请输入一个要插入的整数: '))
insertList(L1,x)
print(L1)
```

案例实现代码：

```
def insertList(L1, x):
    def __get_index(r, l=0):
        if r-l <= 1: return l
        return __get_index(l=(l+r)//2, r=r) if L1[(l+r)//2] < x else
__get_index(l=l, r=(l+r)//2)

    L1.insert(__get_index(len(L1))+1, x)
```

4.2 问题求解二

(1) 阅读下面 Python 语句，给出运行结果并分析原因 (2 分)

```
d = {1:'a', 2:'b', 3:'c'};
del d[1]
d[1] = 'x'
```



```
del d[2]
print(d)
```

运行结果 (1 分):

{3: 'c', 1: 'x'}

或 {1: 'x', 3: 'c'}

(结果由解释器行为决定)

原因分析 (1 分):

`d = {1:'a', 2:'b', 3:'c'}` 创建了一个包含三个键值对的字典。

`del d[1]` 删除了键为 1 的键值对。

`d[1] = 'x'` 向字典中添加了一个新的键值对, 键为 1, 值为 'x'。

`del d[2]` 删除了键为 2 的键值对。

最后, `print(d)` 打印出字典 `d` 的当前状态, 此时字典中只包含两个键值对:

{3: 'c', 1: 'x'}

(2) 一棵二叉树共有 25 个节点, 其中 5 个是叶子结点, 则度为 1 的节点数有多少个? (3 分)

答案 (1 分):

度为 1 的节点数有 16 个

分析 (2 分):

根据二叉树的基本性质, 在任意一棵二叉树中, 度为 0 的结点(叶子结点)总是比度为 2 的结点多一个。本题中度为 2 的结点即为 4 个。二叉树只包含度为 0 的结点、度为 1 的结点和度为 2 的结点。度为 1 的结点个数等于总结点数减去度为 0 和 2 的结点数。25-5-4=16

5 Python 文件处理

本题共计 10 分：代码实现 7 分，代码测试 3 分。

案例编程要求：假设当前目录下有一个文件名为 score1.txt 的文本文件，存放着某班学生的计算机课程成绩，共有学生学号、平时成绩和期末成绩三列，每一行数据之间采用空格分隔。请编写程序完成下列要求：

(1) 根据平时成绩占 40%、期末成绩占 60% 的比例计算总评成绩(取整数)，并将每个学生的学号和总评成绩写入另一个文件 score2.txt 中，每个学生的数据占据一行。

(2) 在屏幕上输出本班学生总人数。

(3) 根据总评成绩，统计并输出 90 以上、80~89、70~79、60~69、60 分以下各成绩档学生的人数，并输出班级总平均分（取整数）。

程序运行效果如下图所示。

学生总人数为50。按总评成绩统计各成绩档学生的人数：
90以上9人、80~89有11人、70~79有14人、60~69有10人、60分以下6人。
班级总平均分为75.0分。

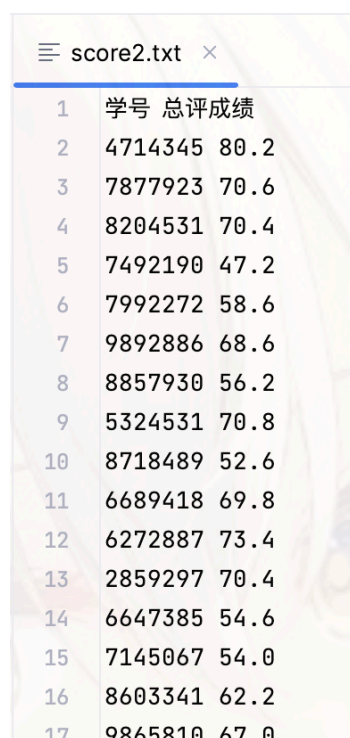
问题求解代码（7 分）：

```
import sys

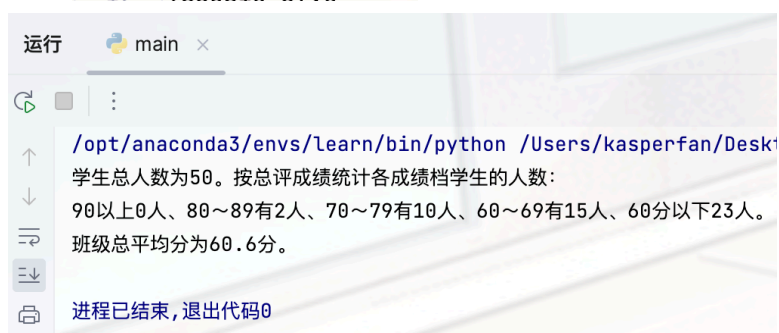
score2: list
with open("score1.txt", mode='r') as f:
    score2 = [[tokens[0]] + [int(tokens[1]) * 0.4 + int(tokens[2]) * 0.6] for tokens in
                (line.split() for line in f.readlines()[1:])]
with open("score2.txt", mode='w') as f:
    sys.stdout = f
    print("学号", "总评成绩", sep="\t")
    print(*("{ }\t{:.1f}".format(i[0], i[1]) for i in score2),
          sep="\n")
sys.stdout = sys.__stdout__
seq = [0 for _ in range(5)]
total = 0
for i in score2:
    if i[1] < 60: total += i[1]; seq[0] += 1
    elif 60 <= i[1] < 70: total += i[1]; seq[1] += 1
    elif 70 <= i[1] < 80: total += i[1]; seq[2] += 1
    elif 80 <= i[1] < 90: total += i[1]; seq[3] += 1
    elif 90 <= i[1]: total += i[1]; seq[4] += 1
```

```
print(f"学生总人数为{len(score2)}。按总评成绩统计各成绩档学生的人数：",  
      f"90 以上{seq[4]}人、80~89 有{seq[3]}人、70~79 有{seq[2]}人、  
60~69 有{seq[1]}人、60 分以下{seq[0]}人。",  
      "班级总平均分为{:.1f}分。".format(total / len(score2)),  
      sep="\n")
```

测试结果展示 (3 分):



1	学号	总评成绩
2	4714345	80.2
3	7877923	70.6
4	8204531	70.4
5	7492190	47.2
6	7992272	58.6
7	9892886	68.6
8	8857930	56.2
9	5324531	70.8
10	8718489	52.6
11	6689418	69.8
12	6272887	73.4
13	2859297	70.4
14	6647385	54.6
15	7145067	54.0
16	8603341	62.2
17	9865810	67.0



```
运行 main ×  
/opt/anaconda3/envs/learn/bin/python /Users/kasperfan/Desktop/  
学生总人数为50。按总评成绩统计各成绩档学生的人数：  
90以上0人、80~89有2人、70~79有10人、60~69有15人、60分以下23人。  
班级总平均分为60.6分。  
进程已结束,退出代码0
```

6 Python 数值处理

本题共计 8 分：代码实现 6 分，代码测试 2 分。

问题描述：请编写程序，生成随机密码。具体要求如下：

- (1)使用 random 库，采用 0x1010 作为随机数种子。
- (2)密码由 26 个字母大小写、10 个数字字符和!@#\$%^&*等 8 个特殊符号组成。
- (3)每个密码长度固定为 10 个字符。
- (4)程序运行每次产生 10 个密码，每个密码一行。
- (5)每次产生的 10 个密码首字符不能一样。
- (6)程序运行后产生的密码保存在“随机密码.txt”文件中。

编程实现（6 分）：

```
import random, string, sys

random.seed(0x1010)
chars = string.ascii_letters+string.digits+"!@#$%^&*"
pwds = []
exclude = set()
while len(pwds) < 10:
    pwd = ""
    while len(pwd) < 10:
        pwd+=random.choice(chars)
        if pwd[0] in exclude: continue
        exclude.add(pwd[0])
    pwds.append(pwd)
with open("随机密码.txt", mode="w") as f:
    sys.stdout = f
    print(*pwds, sep="\n")
sys.stdout = sys.__stdout__
```

测试结果（2 分）：

三 随机密码.txt ×

1	So1WpkoC6i
2	armJ75eUG8
3	B*GcqsYC^B
4	wQ2bcfcAJy
5	Xdyg7pQTIS
6	Y0!0YH0AP2
7	cuhZUk@s4&
8	D@3d8\$TBfp
9	TBm#WfYNHr
10	Ue64y\$E8Cv
11	

7 Python 字符串和文本处理

本题共计 10 分：第一小题 8 分，第二小题 2 分。

7.1 问题求解

问题描述。比较字符串：提示用户输入两个字符串，将其进行比较，输出较小的字符串，要求只能使用单字符比较操作。

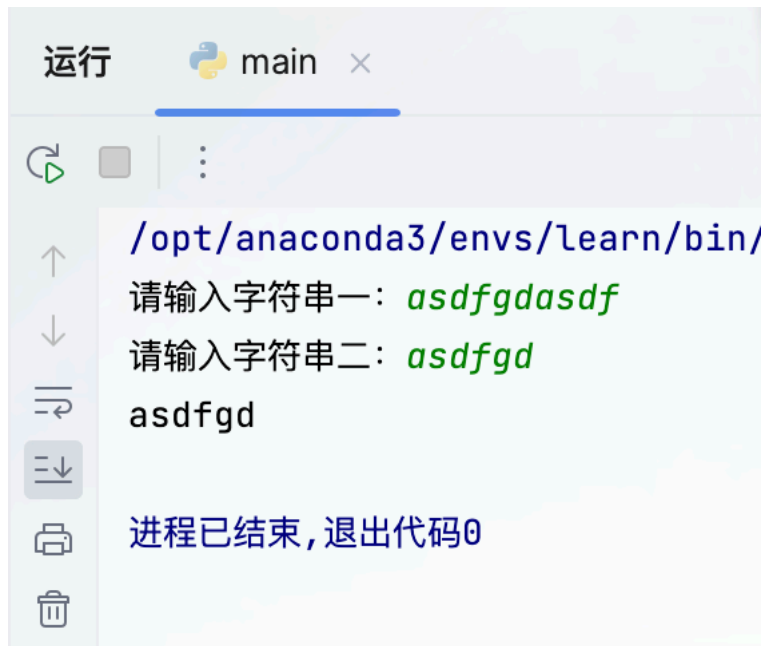
实现代码（6 分）：

```
str1 = input('请输入字符串一：')
str2 = input('请输入字符串二：')
if len(str1) == len(str2):
    for i in range(len(str1)):
        if str1[i] == str2[i]: continue
        print(str1 if str1[i] < str2[i] else str2)
        break
else: print(str1 if len(str1) < len(str2) else str2)
```

测试结果（2 分）：



```
运行 main x
/opt/anaconda3/envs/learn/bin/pyt
请输入字符串一： asdfgdasdf
请输入字符串二： asdfgdl;kn
asdfgdasdf
进程已结束,退出代码0
```



7.2 程序分析

阅读下面 Python 语句, 请问输出结果是什么?

```
s1="QQ"  
s2="Wechat"  
print("{:*<10} {:=>10}".format(s1,s2))
```

程序运行结果 (2 分):

QQ*****====Wechat

8 Python 面向对象设计

本题共计 10 分：代码实现 8 分，代码测试 2 分。

问题描述：设计一个 Course（课程）类，该类中包括 number（编号）、name（名称）、teacher（任课教师）、location（上课地点）共 4 个属性，其中 location 是私有属性；还包括__init__()、show_info()（显示课程信息）共两个方法。设计完成后，创建 Course 类的对象显示课程的信息。

实现代码（8 分）：

```
class Course:
    def __init__(self, number, name, teacher, location):
        self.number, self.name, self.teacher, self.__location =
number, name, teacher, location

    def show_info(self):
        print(f"课程编号: {self.number}",
              f"课程名称: {self.name}",
              f"任课教师: {self.teacher}",
              f"上课地点: {self.__location}", sep="\n")

course1 = Course("CS101", "计算机科学导论", "张磊", "天一楼 7212")
course1.show_info()
print(course1.name)
print(course1.number)
print(course1.teacher)
print(course1.__location)
```

测试结果（2 分）：

1 个用法

`class Course:`

```
def __init__(self, number, name, teacher, location):  
    self.number, self.name, self.teacher, self.__location = number, name, teacher, location
```

1 个用法

```
def show_info(self):  
    print(f"课程编号: {self.number}",  
          f"课程名称: {self.name}",  
          f"任课教师: {self.teacher}",  
          f"上课地点: {self.__location}", sep="\n")
```

```
course1 = Course("CS101", "计算机科学导论", "张磊", "天一楼7212")  
course1.show_info()  
print(course1.name)  
print(course1.number)  
print(course1.teacher)  
print(course1.__location)
```

main ×

```
/opt/anaconda3/envs/learn/bin/python /Users/kasperfan/Desktop/.Code/Learn/Python/PythonLearn/main.py
```

课程编号: CS101

课程名称: 计算机科学导论

任课教师: 张磊

上课地点: 天一楼7212

计算机科学导论

CS101

张磊

Traceback (most recent call last):

```
File "/Users/kasperfan/Desktop/.Code/Learn/Python/PythonLearn/main.py", line 277, in <module>  
    print(course1.__location)
```

AttributeError: 'Course' object has no attribute '__location'

进程已结束,退出代码1

9 Python 模块设计

本题共计 5 分，第一小题 3 分，第二小题 2 分。

(1) 简述 Python 模块的导入操作 (3 分)。

导入模块的基本操作是使用 `import` 语句。当导入一个模块，Python 会在模块路径列表中搜索模块，并编译为字节代码，如果成功，模块代码将被执行。可以通过以下几种方式导入模块：

- 使用 `import` 语句直接导入一个模块：`import module_name`
- 使用 `from` 关键字导入特定的变量、函数或类：`from module_name import some_function`
- 使用 `as` 关键字给模块指定别名：`import module_name as alias`
- 使用 `from module_name import *` 导入模块中的所有公有成员。

(2) 简述 Python 模块名称和定义成员的查询操作 (2 分)。

块名称查询：每个模块都有一个内置属性 `__name__`，当模块被直接运行时，`__name__` 的值为 `'__main__'`，当模块被导入时，`__name__` 的值为模块的名字。

定义成员查询：可以使用 `dir()` 函数列出模块定义的名称。这返回一个排序过的字符串列表：`dir(module_name)`。如果没有提供参数，`dir()` 会列出当前定义的名称。

心得体会

(4 分)

我对这门 Python 课程的学习体会是深刻且充满挑战的。从最初的 Python 语言基础到最后的模块设计，每一个章节都像是攀登一座新的高峰，每一次成功的编程尝试都让我感到无比的成就感。

在完成各个设计题目时，我不仅复习了课堂上学到的知识，还通过实践加深了理解。例如，在处理 Python 流程控制的题目时，我真正体会到了编程逻辑的严谨性；而在进行函数设计时，我学会了如何将复杂的问题分解成简单的子问题，这对于提高编程效率至关重要。

特别是在进行异常处理和文件操作的学习时，我意识到了编程中的健壮性和用户体验的重要性。通过学习 Python 的内置数据结构和面向对象的概念，我对编程有了更全面的认识。而模块设计的学习，则让我明白了代码复用和程序结构的重要性。

总的来说，这门课程不仅教会了我 Python 编程的技能，更重要的是培养了我解决问题的能力。我相信这些宝贵的经验将在我的未来学习和工作中发挥重要作用。感谢这门课程，也感谢尊敬的张辉辉老师，是辉辉老师对原理和流程的详尽指导使我受益匪浅。我期待着将所学应用到更多实际项目中，继续在编程的道路上探索和前进。