

深圳大学实验报告

课程名称： 现代程序设计

实验项目名称： Python 语法基础

学院： 电子与信息工程学院

专业：

指导教师： 邹文斌

报告人： 学号： 班级：

实验时间： 2023 年 9 月 12 日

实验报告提交时间： 2023 年 9 月 13 日

教务部制

一、 实验要求

a) 基本数据类型

- b) 变量的引用与内存机制
- c) 格式化输出
- d) 需上交实验报告、py 源程序。

二、 实验环境

Python IDLE, Pycharm

三、 实验内容

（一）抹零结账

某超市为吸引顾客特设置抹零结账，即抹去小数点后面的金额。编程实现抹零结账。

1. 输入商品名称、金额。
2. 将各个商品累加。
3. 打印输出商品总金额。
4. 将总金额转换为整数实现抹零，打印输出实收金额。

（二）简易计算器

编写一个简易计算器程序，一次性计算出两个数加、减、乘、除的结果。

输入：数 1，数 2。

输出：两个数分别加、减、乘、除的结果。

（三）恺撒密码

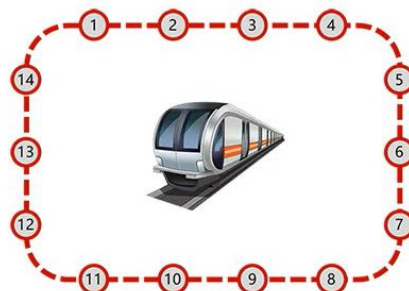
恺撒密码是一种替换加密技术，以罗马共和时期恺撒大帝的名字命名为了保密需要，重要军事情报传递过程中，需要将明文用密文替换，以防被敌人截获后掌握情报。

恺撒密码是将明文中的所有字母，都根据字母表，向后偏移一个固定数量，成为密文。例如，当偏移量是 3 的时候，所有的字母 A 被替换成 D，B 被替换成 E，以此类推。

输入一个字母(大写 A~W)，代表明文，当偏移量是 3 时，输出密文对应的字母。

（四）环形地铁

环线地铁的轨道首尾相连，共有 14 座车站，用数字 1~14 表示车站编号，如下图所示。



地铁双向发车，如果从 2 号车站乘车到 10 号车站，可以按 2→3→4→5→6→7→8→9→10 的路线，乘 8 站到达目的地；也可以乘坐反方向的车，按 2-1-14-13-12-11-10 的路线，乘 6 站到达目的地。假设各站间距相同，显然 6 站比 8 站要快。

输入起始车站和目的地车站的编号（1~14），请你算一算，从起始车站到达目的地车站最少乘坐几站。

四、实验过程

思路：

（涉及到算法实现的实验需阐述算法的逻辑关系）

（一）抹零结账

- 1、先输入各商品名称以及金额，在输入金额时设置为浮点型，便于下一步计算、抹零等操作。
- 2、将各个商品的金额累加，计算结果保留三位小数打印输出商品总金额。
- 3、将计算结果转换为整型输出，实现抹零。

（二）简易计算器

- 1、输入数字 1 以及数字 2，并在输入时将其设置为浮点型，便于计算处理。
- 2、分别对两个数进行加、减、乘、除法计算。
- 3、输出计算结果，得到两个数分别加、减、乘、除的结果。

（三）凯撒密码

- 1、题目要求输入一个字母(大写 A~W)进行凯撒加密，故打算设置条件判断，只对 A~W 范围内字母进行该操作。
- 2、用户输入字母。
- 3、通过 ASCII 码判别用户输入的字母是否在大写 A~W 范围内，如果是，则对字母进行偏移量+3 的凯撒加密并输出加密结果；如果不是，则要求用户重启程序进行正确输入。

（四）环形地铁

- 1、输入起始车站编号和目的地车站编号
- 2、考虑车站编号需要在 1~14 范围内，加入条件判断语句对输入的车站编号进行

分析：若起始车站及目的地编号均在 1~14 范围内，执行计算；若起始车站编号和目的地车站编号不全在 1~14 范围内，要求用户重启程序进行正确输入。

3、分别判断起始车站编号比目的地车站编号大和小的时候两种乘车方向的计算流程，设置计算公式并得出结果，打印输出两种乘车方向需搭乘的车站数。

4、比较两种乘车方向需要搭乘的车站数，计算较快的方向比较慢的方向少搭乘的车站数并输出打印。

完整代码：

（必须有详细的注释）

（一）抹零结账

#输入商品名称和金额

```
A = input ('请输入商品 A 名称：')
Price_A = float (input('请输入商品 A 金额：'))
B = input ('请输入商品 B 名称：')
Price_B = float (input('请输入商品 B 金额：'))
C = input ('请输入商品 C 名称：')
Price_C = float (input('请输入商品 C 金额：'))
```

#对结果进行计算和输出

```
Price = Price_A + Price_B + Price_C
print ("商品总金额为%.3f 元"%Price)
print ("实收金额为%d 元"%int(Price))
```

（二）简易计算器

#输入两个数

```
a = float ( input ('请输入数 a :'))
b = float ( input ('请输入数 b :'))
```

#分别进行加减乘除法计算

```
plus = a + b
subl= a - b
```

```
sub2= b - a

mul = a *b

div1= a / b

div2= b / a

#输出各结果（保留三位小数）

print (' a + b =%.3f'%plus)

print (' a - b =%.3f'%sub1)

print (' b - a =%.3f'%sub2)

print (' a * b =%.3f'%mul)

print (' a / b =%.3f'%div1)

print (' b / a =%.3f'%div2)
```

（三）凯撒密码

```
#输入字母

Gra = input (' 请输入明文大写字母(A`W): ')

#判断输入是否在范围内，如果是则进行操作，否则要求重启程序进行正确输入

if ord('A')<=ord(Gra)<=ord('W') :

    encrypt = chr ( ord (Gra)+3)

    print (' 加密后的密文字母为%s'%encrypt)

else:

    print(' 您输入的字母不在 A`W 范围内，请重启程序再次输入')
```

（四）环形地铁

```
#获得起始和目的地车站编号

Start = int(input(' 请输入起始车站编号（1-14):'))

Target = int(input(' 请输入目的地车站编号(1-14):'))

if 1<=Start<=14 and 1<=Target<=14:

    #当目的地车站编号比起始车站编号大时搭车站数的计算方法
```

```

if Target > Start :

    Dis_F = Target - Start

    Dis_R =14- Target + Start

    print ('正向乘车(1-14 --> 14-1)需要搭乘%d 站'%Dis_F)

    print ('反向乘车(14-1 --> 1-14)需要搭乘%d 站'%Dis_R)

    #比较计算两种方向搭乘少乘坐几站

        if Dis_F > Dis_R :

            print('反向乘车(14-1 --> 1-14)比正向乘车(14-1 --> 1-14)要
少乘坐%d 站'%(Dis_F-Dis_R))

            if Dis_R > Dis_F :

                print ('正向乘车 (14-1-->1-14) 比反向乘车 (14-1-->1-14)
要少乘坐%d 站'%(Dis_R - Dis_F))

                if Dis_F == Dis_R :

                    print ('两个方向乘车需要的站数一样')

if Target == Start :

    print ('您不需要乘车')

#当目的地车站编号比起始车站编号小时搭车站数的计算方法

if Target < Start :

    Dis_F =14- Start + Target

    Dis_R = Start - Target

    print ('正向乘车(1-14 --> 14-1)需要搭乘%d 站'%Dis_F)

    print ('反向乘车(14-1 --> 1-14)需要搭乘%d 站'%Dis_R)

    #比较计算两种方向搭乘少乘坐几站

        if Dis_F > Dis_R :

            print('反向乘车(14-1 --> 1-14)比正向乘车(14-1 --> 1-14)要
少乘坐%d 站'%(Dis_F-Dis_R))

            if Dis_R > Dis_F:

```

```

        print('正向乘车（14-1-->1-14）比反向乘车（14-1-->1-14）要少乘坐%d站' % (Dis_R - Dis_F))

        if Dis_F == Dis_R:

            print('两个方向乘车需要的站数一样')

        else:

            print("请确保输入的起始车站编号和目的地编号均在(1~14)范围内，请重启程序")

```

五、 实验结果

（运行结果，截图）

（一）抹零结账

```

>>>
= RESTART: D:\MineP\Program\Python.py\Modern_Program\Zeroing_closing_accounts.py
请输入商品 A 名称: A
请输入商品 A 金额: 0.762
请输入商品 B 名称: B
请输入商品 B 金额: 13
请输入商品 C 名称: C
请输入商品 C 金额: 1.000
商品总金额为14.762元
实收金额为14元
>>>

```

（二）简易计算器

```

>>>
==== RESTART: D:\MineP\Program\Python.py\Modern_Program\Simple_calculator.py
请输入数 a :0.5
请输入数 b :14
a + b =14.500
a - b =-13.500
b - a =13.500
a * b =7.000
a / b =0.036
b / a =28.000
>>>

```

（三）凯撒密码

```

>>>
===== RESTART: D:\MineP\Program\Python.py\Modern_Program\Encryption.py ==
请输入明文大写字母(A~W): Z
您输入的字母不在A~W范围内, 请重启程序再次输入
>>>
===== RESTART: D:\MineP\Program\Python.py\Modern_Program\Encryption.py ==
请输入明文大写字母(A~W): W
加密后的密文字母为Z
>>>

```

(四) 环形地铁

```

>>>
===== RESTART: D:\MineP\Program\Python.py\Modern_Program\Circular_subway.py
请输入起始车站编号 (1-14):20
请输入目的地车站编号(1-14):1
请确保输入的起始车站编号和目的地编号均在(1~14)范围内, 请重启程序
>>>
===== RESTART: D:\MineP\Program\Python.py\Modern_Program\Circular_subway.py
请输入起始车站编号 (1-14):1
请输入目的地车站编号(1-14):1
您不需要乘车
>>>
===== RESTART: D:\MineP\Program\Python.py\Modern_Program\Circular_subway.py
请输入起始车站编号 (1-14):6
请输入目的地车站编号(1-14):9
正向乘车(1-14 --> 14-1)需要搭乘3站
反向乘车(14-1 --> 1-14)需要搭乘11站
正向乘车 (14-1-->1-14) 比反向乘车 (14-1-->1-14) 要少乘坐8站
>>>

```

六、 实验心得

(本次实验遇到的问题, 解决过程, 有什么收获等)

- 1、 熟悉了 **Python** 的输入、输出、类型转换语法使用。
- 2、 实验过程中查阅资料学会了 **Python** 的 **if** 语句使用和 **and** 的使用。
- 3、 过程中发现格式化输出时的占位符%在中英文打字法(全角半角)时也会不同, 容易造成报错。

深圳大学学生实验报告用纸

成绩评定：			
实验过程（60 分）	实验结果（30 分）	心得体会（10 分）	总分（100 分）
指导教师签字： 年 月 日			
备注：			

- 注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。
- 2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。