**作业1**

**题目 冰雹序列图的绘制**

**源代码**

import turtle

# 定义 3 个输入值相同的变量：

# begin 不会再改变并用于确定初始和结束的位置

# test 用于试运行并知晓行走步数

# number 用于实际运行

begin = test = number = int(input('Give me the number you want: '))

counter = 0

if number <= 1:

    print('Sorry. We cannot calculate the number you give.')

    exit()

while test != 1:

    if not test % 2:

        test /= 2

    elif test % 2:

        test = test\*3 + 1

    counter += 1  # 记录步数

stepX = 500//counter  # 优化显示效果，使无论输入什么值，图像在左右边界上都处在合适的位置

turtle.speed(9)  # 提升运行速度

turtle.penup()

turtle.setposition(-(counter//2)\*stepX, number - begin//2)

turtle.pendown()

turtle.write(number)

for i in range(-(counter//2)\*stepX + stepX, -(counter//2)\*stepX + counter\*stepX+1, stepX):

    if number % 2 == 0:

        number /= 2

    elif number % 2:

        number = number\*3 + 1

    turtle.goto(i, number - begin//2)

    turtle.write(number)

    print(number)  # 方便后台查看运行情况

turtle.done()

**运行结果截屏**

图表

低可信度描述已自动生成

**作业2**

**题目 密码验证**

You are creating a new account and need to provide a password. The password has the

following requirements:

(a) The password must be at least 6 characters and at most 20 characters.

(b) It must contain at least one lowercase letter, one uppercase letter, and one number.

Write a program that prompts the user to input a password and checks if the password is valid. If the password is valid, print a confirmation statement. If it is not, print a statement that the password is not valid.

**源代码**

# 判断是否含有数字

def checkNumber(str):

    for x in str:

        if x.isnumeric():

            return True

    return False

# 判断是否同时含有大小写字母

def checkLetter(str):

    tag1 = tag2 = False

    for x in str:

        if x.isupper():

            tag1 = True

        if x.islower():

            tag2 = True

    return tag1 and tag2

password = input("请输入密码：")

secure1 = secure2 = secure3 = False

# 规则1：密码长度

secure1 = len(password) in range(6, 21)

# 规则2：包含数字

secure2 = checkNumber(password)

# 规则3：包含大小写字母

secure3 = checkLetter(password)

if secure1 and secure2 and secure3:

    confirm = input("您的密码强度合格！是否确认使用该密码(Y/N): ")

    if confirm == 'Y':

        print('已成功使用该密码！')

    elif confirm == 'N':

        print('密码设置已取消！')

    else:

        print('未输入 Y/N 。已取消设置密码。')

else:

    print("您设置的密码强度不合格。请遵从以下规范：")

    if not secure1:

        print(" - 密码长度应不小于6位且不大于20位")

    if not secure2:

        print(" - 密码中应包含数字")

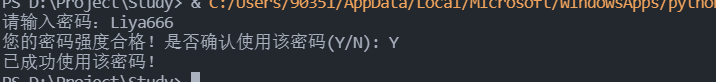
    if not secure3:

        print(" - 密码中应同时包含大小写字母")

**运行结果截屏**

文本

描述已自动生成



**作业3**

**题目 重组后的单词**

Using the incredible power of the human mind, according to research at Cambridge University, it doesn't matter in what order the letters in a word are, the only important thing is that the first and last letter be in the right place. The rest can be total mess and you can still read it without a problem. This is because the human mind does not read every letter by itself, but the word as a whole. Amazing, huh? Yeah and I always thought spelling was important! Write a program that prompts for a word and prints the scrambled version (with first and last fixed).

Hint: The random module helps scramble words.

**源代码**

import random

word = input('Please input a word: ')

if word.isalpha():

    # 创建单词序列并改变顺序

    order = list(word[1:-1])

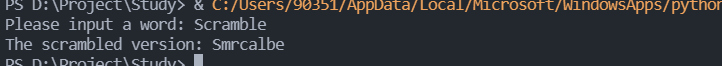
    random.shuffle(order)

    print('The scrambled version: ' + word[0] + ''.join(order) + word[-1])

else:

    print('Sorry. What you input is incorrect.')

**运行结果截屏**



**作业4**

**题目 元音检查**

Develop a Python program that will:

(1) prompt for input,

(2) extract all the vowels in the input string and make them lowercase,

(3) add all the nonduplicate vowels you found into a temporary string, and

(4) stop the code when you have collected all five vowels (aeiou) and print the number of trials that you did to collect all five vowel characters.

**源代码**

inputStr = str.lower(input('Please input a string: '))

# 逐个字符验证是否符合条件并提取到临时字符串上

vowels = ['a', 'e', 'i', 'o', 'u']

str = ''

step = 0

for letter in inputStr:

    step += 1

    if letter in vowels:

        str += letter

    if vowels[0] in str and vowels[1] in str and vowels[2] in str and vowels[3] in str and vowels[4] in str:

        break

# 对临时字符串去重处理

strList = list(str)

dealedList = list(set(strList))

dealedList.sort(key=strList.index)

if dealedList == []:

    print('Sorry. No vowel was found.')

else:

    print('These vowels were found: ' + ''.join(dealedList))

    print('Times we checked: ', step)

**运行结果截屏**

