**Python实验报告4**

**实验四**：程序的控制结构

**实验目的**：了解程序的基本结构并绘制流程图

掌握程序的分支结构并运用if语句实现分支结构

掌握并应用for语句和while语句实现循环结构

掌握随即库使用方法

了解程序的异常处理方法及用法

**实验要求**：Python程序控制语法及结构

学会编写带有条件判断及循环的程序

**实验过程**：1. 程序练习题4.2 统计不同字符个数。用户从键盘输入一行字符，编写一个程序，统计并输出其中英文字符、数字、空格和其他字符的个数。

编写代码

n\_alphabet = 0

n\_number = 0

n\_space = 0

n\_other = 0

s = input('请输入一行字符')

for c in s:

if 'A' <= c <= 'Z' or 'a' <= c <= 'z':

n\_alphabet += 1

elif '0' <= c <= '9':

n\_number += 1

elif c == ' ':

n\_space += 1

else:

n\_other += 1

print('有{0}个英文字符，{1}个数字，{2}个空格和{3}个其它字符'.format(n\_alphabet, n\_number, n\_space, n\_other)

2. 程序练习题4.3 最大公约数计算。从键盘接受两个整数，编写程序求出这两个整数的最大公约数和最小公倍数

编写代码

n1, n2 = eval(input('请输入两个整数，中间用逗号隔开：'))

x, y = n1, n2

r = x % y

while r > 0:

x = y

y = r

r = x % y

print('{}和{}最大公约数是：{}， 最小公倍数是：{}'.format(n1, n2, y, n1 \* n2 / y))

3. 程序练习题2.8（教材57页）正方形螺旋线的绘制。利用turtle库绘制一个正方形螺旋线。如图所示。

编写代码

import turtle

edge = 200

turtle.setup(500,500,-200,-200)

while edge > 5:

theta = 90

turtle.seth(theta)

turtle.fd(edge)

theta = theta - 90

turtle.seth(theta)

turtle.fd(edge)

edge = edge - 5

theta = theta - 90

turtle.seth(theta)

turtle.fd(edge)

theta = theta - 90

turtle.seth(theta)

turtle.fd(edge)

edge = edge - 5

4. 程序练习题3.5（教材94页）田字格的输出。使用print（）函数输出如图所示样式的田字格。

编写代码

for i in range(11):

for j in range(11):

if i%5 == 0:

if j%5 == 0:

print("+ ", end ='')

else:

print("- ", end='')

else:

if j%5 == 0:

print("| ", end='')

else:

print(" ", end='')

print("\n")

5. 程序练习题4.1 猜数游戏。在程序中预设一个0~9之间的整数，让用户通过键盘输入所猜的数，如果大于预设的数，显示“遗憾，太大了”；小于预设的数，显示“遗憾，太小了”，如此循环，直至猜中该数，显示“预测N次，你猜中了！”，其中N是用户输入数字的次数。

编写代码

p = 7

count = 0;

while True:

n = eval(input('请输入一个0-9之间的整数：'))

count += 1

if n > p:

print('遗憾，太大了')

elif n == p:

print('预测{}次，你猜中了！'.format(count))

break

else:

print('遗憾，太小了')

5. 程序练习题4.4 猜数游戏续。改编程序练习题4.1，让计算机能够随机产生一个预设数字，范围在0~100之间，其他游戏规则不变。

编写代码

from random import \*

p = randint(0,100)

count = 0;

while True:

n = eval(input('请输入一个0-100之间的整数：'))

count += 1

if n > p:

print('遗憾，太大了')

elif n == p:

print('预测{}次，你猜中了！'.format(count))

break

else:

print('遗憾，太小了')

6. 程序练习题4.5 猜数游戏续。对于程序练习题4.4程序，当用户输入的不是整数（如字母、浮点数等）时，程序会终止执行退出。改编该程序，当用户输入出错时给出“输入内容必须为整数！”的提示，并让用户重新输入。

编写代码

from random import \*

seed(0)

p = randint(0,100)

count = 0;

while True:

n = eval(input('请输入一个0-100之间的整数：'))

if not isinstance(n, int):

print('你输入的不是整数。')

break

count += 1

if n > p:

print('遗憾，太大了')

elif n == p:

print('预测{}次，你猜中了！'.format(count))

break

else:

print('遗憾，太小了')

7. 程序练习题4.6 羊车门问题。有3扇关闭的门，一扇门后面停着汽车，其余门后是山羊，只有主持人知道每扇门后面是什么。参赛者可以选择一扇门，在开启它之前，主持人会开启另一扇门，露出门后的山羊，然后允许参赛者更换自己的选择。请问：参赛者更换选择后能否增加猜中汽车的机会？——这是一个经典的问题。

编写代码

from random import \*

N = 1000000

sequence = ['车','羊1','羊2']

first\_guess\_count = 0

change\_guess\_count = 0

for i in range(N):

guess = choice(sequence)

if guess == '车':

first\_guess\_count += 1

else:

change\_guess\_count += 1

print("改变选择获胜的概率：{:.4f}；坚持选择获胜的概率：{:.4f}".format(first\_guess\_count / N, change\_guess\_count / N))

8.类似于羊车门的问题：有三个盒子，一个盒子装红球，另外俩个盒子装的白球。

老师知道每个盒子装的什么球。并且当你选择一个盒子时，老师会给你指明一个错误的答案，这时你可以选择是否更换自己的选择。求更换与不更换选择，哪种方式正确的概率大。

**实验思考：**

程序的基本结构，有分支结构、循环结构等，对于遍历循环（for 语句）和无限循环（while 语句），程序的异常处理（try和except保留字），循环结构中有保留字:break 和 continue。对于random库，其引用方法与math库一样。