**《Python高级应用》实验报告**

实验名称 ： 实验1 基础实验

实验日期 ： 　 2024/9/9

姓 名 ： 　　 高心阳

学 号 ： 　 084623237

班 级 ： 　 计算机232

成 绩 ：

**人工智能与信息技术学院**

**南京中医药大学**

|  |
| --- |
| **实验目的：** |
| 1. 配置并熟悉Python开发环境 2. 掌握Python基本语法的使用 3. 掌握标准库和第三方库的使用 |
| **实验内容和要求** |
| 1、运行或补充编写下列程序，熟悉Python语言的开发和运行环境。如果代码报错，请仔细修改。  1）整数序列求和。用户输入一个正整数N，计算从1到N（包含1到N）相加之后的结果。  n=input(“请输入整数N:”)  sum=0  for i in range(int(n))  sum+=i+1  print(“1到N求和结果：”,sum)  2）字符串拼接。接受用户输入的两个字符串，将它们组合后输出。  str1=input(“请输入一个人的名字：”)  ( )//str2是通过input输入的国家名，请补充代码  print(“世界那么大，{ }想去{ }看看。”.format(str1,str2))  3）健康食谱输出。列出五种不同食材，输出它们可能组成的所有菜式名称。  diet=[‘西红柿’,’花椰菜’,’黄瓜’,’牛排’,’虾仁’]  for x in range(0,5):  for y in range(0,5):  if not(x=y):  print(“{ }{ }”.format(diet[x],diet[y]))  4）九九乘法表输出。工整打印输出常用的九九乘法表，格式不限。  for i in range(1,9):  for j in range(1,i+1):  print(“{ }\*{ }={:2}”.format(j,i,i\*j),end=’’)  print(‘ ’)  5）编程实现计算2!+3!+…+10!的结果。  6）Hello World的条件输出  描述：获得用户输入的一个整数，参考该整数值，打印输出"Hello World"，要求：‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬  （1）如果输入值是0，直接输出"Hello World"‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬  （2）如果输入值大于0，以两个字符一行方式输出"Hello World"（空格也是字符）‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬  （3）如果输入值小于0，输出"Hello World"的第1和最后一个字符。  2、编程实现：输入水的重量（以kg为单位）以及水的初始温度和最终温度，计算并输出将水从初始温度加热到最终温度所需的能量。  计算能量的公式为：  Q=M\*(最终温度-初始温度)\*4184  这里M是以kg为单位的水的重量，温度以℃为单位，能量Q以J为单位。要求输出的结果保留两位小数。   1. 标准库turtle的使用：查阅资料，使用该库画出自己喜欢的图片。   4、1）安装第三方库Numpy；  2）找一张南中医的图片变换风格。如显示图片底片色、改成灰度图等，并保存相应的图片（\*手绘风格选做）。 |
| **运行结果（写清题号）** |
| 1、   |  | | --- | | 1-1代码 | | n=input("请输入整数N:") sum=0 for i in range(int(n)+1):  sum+=i print("1到N求和结果：",sum) |  |  | | --- | | 1-2代码 | | str1=input("请输入一个人的名字：")  str2=input("请输入想去的国家名：")#str2是通过input输入的国家名，请补充代码  print(f"世界那么大，{str1}想去{str2}看看。".format(str1,str2)) |  |  | | --- | | 1-3代码 | | # 相同食材改变顺序视作不同 print("case1:") diet=['西红柿','花椰菜','黄瓜','牛排','虾仁'] for x in range(0,5):  for y in range(0,5):  if not(x==y):  print(f"{diet[x] }{diet[y] }".format(diet[x],diet[y]))  # 相同食材改变顺序视作相同 print("-"\*20) print("case2:") diet=['西红柿','花椰菜','黄瓜','牛排','虾仁'] for x in range(0,5):  for y in range(x+1,5):  print(f"{diet[x] }{diet[y] }".format(diet[x],diet[y]))   # Output: # case1: # 西红柿花椰菜 # 西红柿黄瓜 # 西红柿牛排 # 西红柿虾仁 # 花椰菜西红柿 # 花椰菜黄瓜 # 花椰菜牛排 # 花椰菜虾仁 # 黄瓜西红柿 # 黄瓜花椰菜 # 黄瓜牛排 # 黄瓜虾仁 # 牛排西红柿 # 牛排花椰菜 # 牛排黄瓜 # 牛排虾仁 # 虾仁西红柿 # 虾仁花椰菜 # 虾仁黄瓜 # 虾仁牛排 # -------------------- # case2: # 西红柿花椰菜 # 西红柿黄瓜 # 西红柿牛排 # 西红柿虾仁 # 花椰菜黄瓜 # 花椰菜牛排 # 花椰菜虾仁 # 黄瓜牛排 # 黄瓜虾仁 # 牛排虾仁 |  |  | | --- | | 1-4代码 | | for i in range(1,9):  for j in range(1,i+1):  print("{0}\*{1}={2:2}".format(j,i,i\*j),end=' ')  print('') # Output: # 1\*1= 1  # 1\*2= 2 2\*2= 4  # 1\*3= 3 2\*3= 6 3\*3= 9  # 1\*4= 4 2\*4= 8 3\*4=12 4\*4=16  # 1\*5= 5 2\*5=10 3\*5=15 4\*5=20 5\*5=25  # 1\*6= 6 2\*6=12 3\*6=18 4\*6=24 5\*6=30 6\*6=36  # 1\*7= 7 2\*7=14 3\*7=21 4\*7=28 5\*7=35 6\*7=42 7\*7=49  # 1\*8= 8 2\*8=16 3\*8=24 4\*8=32 5\*8=40 6\*8=48 7\*8=56 8\*8=64 |  |  | | --- | | 1-5代码 | | def f(x):  i = 1  sum = 0  while i <= x:  sum += i  i += 1  return sum def g(n):  i = 1  sum = 0  while i <= n:  sum += f(i)  i += 1  return sum  n = 10 print("1到10的阶乘求和结果：", g(int(n))) # Output: # 1到10的阶乘求和结果： 220 |  |  | | --- | | 1-6代码 | | str="Hello World" x=input("请输入一个数字：") x=int(x) if x==0:  print(str) elif x>0:  i=0  while i+1< len(str):  print(str[i:i+2])  i+=2  if i<len(str):  print(str[i:]) else:  print(str[0],str[-1]) |  |  | | --- | | 2代码 | | M=float(input("请输入水的重量（以kg为单位）：")) T1=float(input("请输入水的初始温度（以℃为单位）：")) T2=float(input("请输入水的最终温度（以℃为单位）：")) Q=M\*(T2-T1)\*4184 print("将水从",T1,"℃加热到",T2,"℃所需的能量为：",round(Q,2),"J") |  |  | | --- | | 3代码 | | from turtle import \*  for steps in range(1, 500, 3):  for c in ('red', 'orange', 'yellow', 'green', 'blue', 'purple'):  color(c)  forward(steps)  right(30) |      |  | | --- | | 4代码 | | import numpy as np import PIL.Image as img  img\_njucm = img.open('njucm.jpg') img\_array = np.array(img\_njucm)  # Red\_Channel red\_channel = img\_array.copy() red\_channel[:, :, 1] = 0 red\_channel[:, :, 2] = 0  # Green\_Channel green\_channel = img\_array.copy() green\_channel[:, :, 0] = 0 green\_channel[:, :, 2] = 0  # Blue\_Channel blue\_channel = img\_array.copy() blue\_channel[:, :, 0] = 0 blue\_channel[:, :, 1] = 0  # 转换图像 red\_image = img.fromarray(red\_channel) green\_image = img.fromarray(green\_channel) blue\_image = img.fromarray(blue\_channel) gray\_image = img\_njucm.convert('L')  # 保存修改后的图像 red\_image.save('red\_channel\_njucm.jpg') green\_image.save('green\_channel\_njucm.jpg') blue\_image.save('blue\_channel\_njucm.jpg') gray\_image.save('gray\_njucm.jpg') |   red_channel_njucmgreen_channel_njucmgray_njucmblue_channel_njucm |
| **实验的体会与建议** |
|  |