

深圳大学实验报告

课程名称： Python 程序设计

实验项目名称： 实验 1：函数与模块

学院： 计算机与软件学院

专业： 软件工程

指导教师： 潘浩源

报告人： 郑彦薇 学号： 2020151022

实验时间： 2022/03/18

实验报告提交时间： 2022/03/22

教务部制

一、实验目的：

用 python 语言编写解决 6 个问题的函数或模块，在主函数重调用并运行。

通过利用 python 编写解决一些简单数学问题的函数和模块，进一步熟悉使用 python 编程。

二、实验方法步骤

1、读题，对每个问题提出解决问题的思路

2、按照得到的思路，利用 python 语言编写解决问题的函数或模块

3、主函数正确调用函数或模块，运行代码，调试程序，直至程序可以正确输入输出

三、实验过程及内容：

（一）emirp 数问题：

1、解决问题的思路与方法：

根据问题要求编写解决问题的 emirp 函数：

①规定 emirp 质数的范围从 13 开始，因为根据题意可以容易知道整数里最小的 emirp 数为 13

②判断当前的数是不是质数，如果是质数，再进一步判断

③利用取余得到最后一位、除以 10 删去已处理最后一位实现数据的反转，再对反转数据做是否是质数的判断

④若反转得到的数是质数，判断该数与原数据是否相同，不同则写入列表 list（即用 list 存储 emirp 数），然后 N 进行-1 操作，再重复上述操作知道已经得到 N 个 emirp 数

⑤将最终得到的 list 写入文件” FirstN_Emirp_Numbers”

代码及具体解释如下：

```
1 def emirp(N):
2     temp = 13
3     list = []
4     while N != 0:
5         cnt = 0
6         i = temp
7         num = temp
8         for k in range(2, temp):
9             if temp%k == 0:
10                 cnt = 1 #表示是合数，不用进一步判断
11                 break
12         if cnt == 0:
13             wei = 0
14             while i != 0:
15                 wei += 1
16                 i = i//10
17             sum = 0
18             while wei != 0:
19                 sum += (temp%10)*(10**(wei-1))
20                 wei -= 1
21                 temp = temp//10
22             for j in range(2, sum):
23                 if sum%j == 0:
24                     cnt = 1 #如果反转是合数，不用输出
25                     break
26             if cnt == 0 and sum != num:
27                 N -= 1
28                 list.append(num)
29                 temp = num+1
```

设置 cnt 标志为，若待判断数据不是质数，cnt 赋值为 1；若待判断数据反转后的数据不是质数，cnt 复制为 1；最终判断时，cnt 仍为 0 说明满足质数的反转型数仍是质数，最终加入列表 list 前再加上是否相同的判断

统计位数

质数反转并判断反转数字是否是质数

最终判断满足条件则写入列表 list

```
30
31
32     count = 0
33     with open("FirstN_Emipr_Numbers.txt", 'w') as txt:
34         for p in list:
35             txt.write(str(p) + ' ')
36             count += 1
37             if count % 10 == 0:
38                 txt.write('\n')
39
40 x = input("Enter the n: ")
41 N = int(x)
42 emirp(N)
```

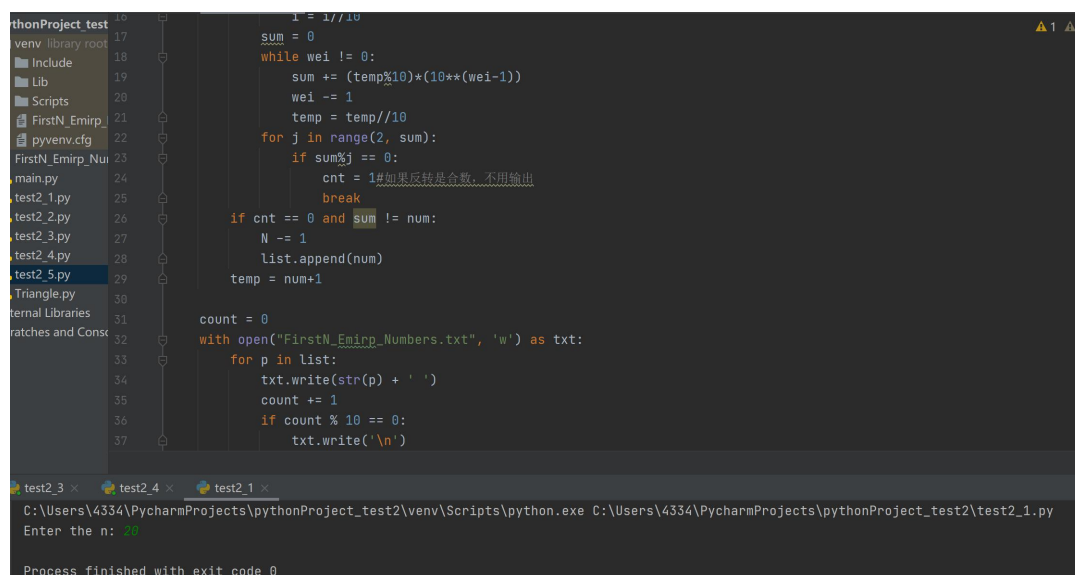
将列表 list 写入文件

2、遇到的问题和收获:

在解决 emirp 数问题时,首先是学习了函数的编写,在 python 中函数可以返回一个或多个值,也可以像这道题一样不返回值;另外是在进行这个问题的解决时,我从三个角度入手:原数据是否是质数→数据反转后是不是质数→两个数据是否相同。在数据反转时采用了比较直接的思路是对整数类型的数据进行每一位的“反转”,但进一步对这道题进行学习收获了另一种更简单的方法,即是待判断的整数视为字符串,再用一个变量存储字符串的倒序输出即可。

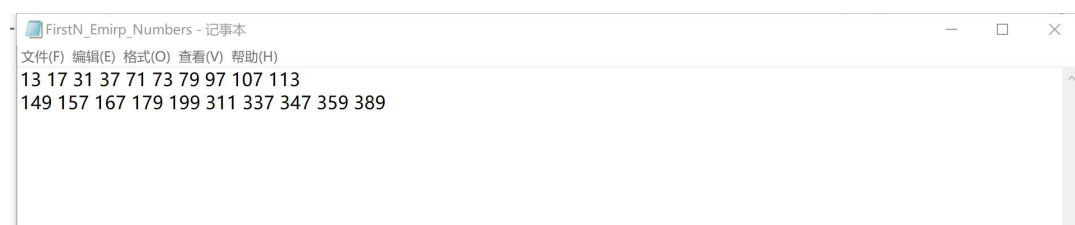
3、代码运行结果展示:

输入 20:



```
16
17
18     sum = 0
19     while wei != 0:
20         sum += (temp%10)*(10**(wei-1))
21         wei -= 1
22         temp = temp//10
23     for j in range(2, sum):
24         if sum%j == 0:
25             cnt = 1#如果反转是合数,不用输出
26             break
27     if cnt == 0 and sum != num:
28         N -= 1
29         list.append(num)
30         temp = num+1
31
32 count = 0
33 with open("FirstN_Emipr_Numbers.txt", 'w') as txt:
34     for p in list:
35         txt.write(str(p) + ' ')
36         count += 1
37         if count % 10 == 0:
38             txt.write('\n')
```

在文本文档中的结果显示为:



```
FirstN_Emipr_Numbers - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
13 17 31 37 71 73 79 97 107 113
149 157 167 179 199 311 337 347 359 389
```

(二) 平方数之和问题:

1、解决问题的思路与方法:

根据问题要求编写解决问题的 checkSumOfSquareNumbers 函数:

- ①根据输入的数据 c 规定整数 i、j 的范围
- ②在双层循环里判断 i、j 的平方和是否与 c 的平方值相同,若相同则直接返回 True
- ③循环结束后仍未返回值,则说明找不到这样的 i、j,返回 False

代码及细节解释如下:

```
1 def checkSumOfSquareNumbers(c):
2     for i in range(1, c):
3         for j in range(i, c):
4             if i**2+j**2 == c**2:
5                 return True
6     return False
7
8 c = int(input("Enter the number: "))
9 if checkSumOfSquareNumbers(c):
10     print("Ture!")
11 else:
12     print("False!")
```

首先根据勾股数性质，勾股一定小于弦，故对 i 规定了 $1 \sim c-1$ ；又因为如(1,3)和(3,1)这样的数据在这个判断里属于同一组数，因此，若以 i 为行, j 为列得到一个矩阵，只需取上三角或下三角，故 j 的范围规定为 $i \sim c-1$ ，减少循环次数

2、遇到的问题和收获:

在解决平方数问题时，这个问题思路比较简单，但是引入矩阵的概念再根据矩阵对称性去限定循环查找满足条件的 a、b 值的范围，可以提高效率。

3、代码运行结果展示:

输入 4:

```
Enter the number: 4
False!
```

输入 5:

```
Enter the number: 5
Ture!
```

(三) 两数和的最接近值问题:

1、解决问题的思路与方法:

根据问题要求编写解决问题的 `twoSumCloesest(nums,target)` 函数:

- ①根据输入的数组 `nums` 得到数组的长度 `l`
- ②设置 `min`，是用来更新 `target` 减去两数和后的最小值，因此 `min` 的初始值可以取一个大数，这里我将其值设置为 1000
- ③得到最小值 `min`: 设置双重循环，数组的第 `i` 位与后面的 `l-i+1` 位数两两组合，与 `target` 做差，将得到的更小值赋给 `min`，直至循环结束，返回最小值 `min`

代码及具体解释如下:

```

1 def twoSumClosest(nums, target):
2     l = len(nums)
3     min = 1000
4     for i in range(1, l):
5         for j in range(i+1, l):
6             if (target-nums[i]-nums[j])<min:
7                 min = target-nums[i]-nums[j]
8     return min
9
10 num = input("Enter the array: ")
11 nums = [int(n) for n in num.split()]
12 target = int(input("Enter the target: "))
13 print(twoSumClosest(nums, target))

```

设置双重循环更新 min 的值

将输入的被默认为字符串的数据转换位整型数组

2、遇到的问题收获:

寻找两数和与 target 的差值最小可以直接用 target 对数组中两两组合的数进行相减, 不断更新 min 值来得到最终的正确结果。

3、代码运行结果展示:

输入 nums=[-1,2,1,-4], target=4:

```

Enter the array: -1 2 1 -4
Enter the target: 4
1

```

输入 nums=[-1,-1,-1,-4], target=4:

```

Enter the array: -1 -1 -1 -4
Enter the target: 4
6

```

(四) 自定义 Triangle 问题:

1、解决问题的思路与方法:

根据问题要求编写解决问题的 Triangle 模块:

①isValid 函数: 根据三角形三边关系, 即任意两边之和大于第三边, 任意两边之差小于第三边设置两个判断语句, 不符合条件则返回 "False", 满足条件返回 "True"

②perimeter 函数: 根据传入函数的三个值, 返回三个值相加的结果即可

③area 函数: 根据输入函数的三个值, 利用海伦公式和 sqrt 函数求出面积, 返回所得值, 实验中我设置返回结果三位小数

perimeter 函数和 area 函数在这里我默认传入的数据满足三角形三边关系

代码及具体解释如下:

```

1 from math import fabs as f
2 from math import sqrt
3
4 def isValid(side1,side2,side3):
5     if side1+side2<=side3 or side1+side3<=side2 or side2+side3<=side1:
6         return False
7     if f(side1-side2)>=side3 or f(side1-side3)>=side2 or f(side2-side3)>=side1:
8         return False
9     return True
10
11 def perimeter(side1,side2,side3):
12     return side1+side2+side3
13
14 def area(side1,side2,side3):
15     p = (side1+side2+side3)/2
16     s = sqrt(p*(p-side1)*(p-side2)*(p-side3))
17     return round(s, 3)

```

判断是否满足三角形三边关系

海伦公式求面积

下面是设置的主函数, 调用模块 Triangle 并实现三个函数, 这里默认三个函数的数据是不同

的输入:

```
1 import Triangle
2
3 side1 = int(input())
4 side2 = int(input())
5 side3 = int(input())
6 if Triangle.isValid(side1, side2, side3):
7     print("True!")
8 else:
9     print("False!")
10
11 side1 = int(input())
12 side2 = int(input())
13 side3 = int(input())
14 print(Triangle.perimeter(side1, side2, side3))
15
16 side1 = int(input())
17 side2 = int(input())
18 side3 = int(input())
19 print(Triangle.area(side1, side2, side3))
```

2、遇到的问题 and 收获:

在计算面积时, 可以借助海伦公式, 注意在取 q 的值时, 除法符号不要误采用整除

3、代码运行结果展示:

输入三组数: 1 2 3, 2 3 4, 2 3 4:

```
1
2
3
False!
2
3
4
9
2
3
4
2.905
```

输入三组数: 都输入 3 4 5:

```
3
4
5
True!
3
4
5
12
3
4
5
6.0
```

(五) 猫头鹰问题:

1、解决问题的思路与方法:

根据问题要求编写解决问题的 owlRow(num)函数:

- ①根据输入的 num 值确定要进行输出的猫头鹰数, 即根据 num 值设置循环
- ②在循环里放入形成猫头鹰的语句

代码及具体解释如下:

```
1 def owlRow(num):
2     empty = ' '
3     lineA = '    {o,o}'
4     lineB = '    /)_)'
5     lineC = '    "  "'
6     for i in range(1, num):
7         print(lineA, end='')
8         print(lineA)
9         for j in range(1, num):
10            print(lineB, end='')
11            print(lineB)
12            for k in range(1, num):
13                print(lineC, end='')
14            print(lineC)
15
16 num = int(input("Enter the num: "))
17 owlRow(num)
```

根据样例输出格式, 对组成猫头鹰的三行逐行进行循环输出

2、遇到的问题和收获:

在进行输出时是逐行输出而不是以一只完整猫头鹰为单位进行输出

3、代码运行结果展示:

输入 5:

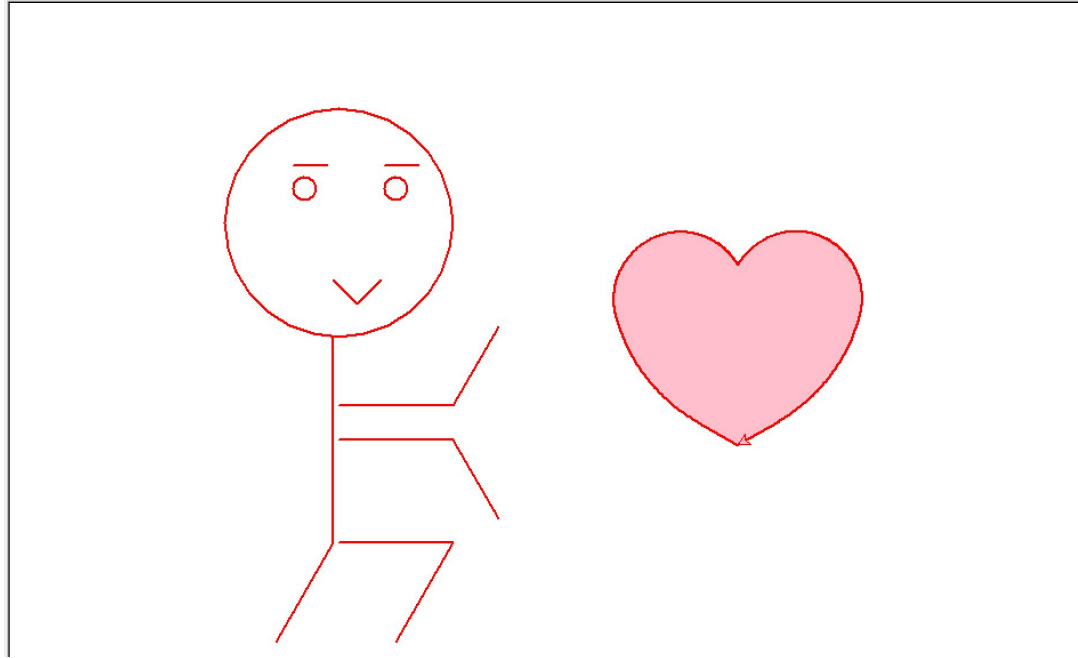
```
Enter the num: 5
    {o,o}    {o,o}    {o,o}    {o,o}    {o,o}
    /)_)    /)_)    /)_)    /)_)    /)_)
    "  "    "  "    "  "    "  "    "  "

Process finished with exit code 0
```

(六) 海龟绘图问题:

1、解决问题的思路与方法:

- (1) ①运行代码后, 得到如下图像:



②解释函数的含义：

up():抬笔，且没有痕迹

down():停笔，会留下痕迹

goto():将画笔移动到规定坐标为(x,y)的位置

circle():以传入该函数的数据为半径画圆，数据为正表示圆心在画笔左边画圆，数据为负表示圆心在画笔的右边画圆

forward():表示向当前方向移动指定的像素长度

right():表示画笔顺时针移动指定度数

left():表示画笔逆时针移动指定度数

speed():设置画笔的移动速度，数字越大越快

2、问题与收获：

对 python 中的 turtle 库进行了学习，并借其功能实现作画。

3、运行结果展示：

根据给出的例子以及 turtle 中部分函数的应用，在（1）的基础上简单作了一个火柴人：

火柴人组成元素代码如下：

```
173 person1(270, 95, 90)
```

```
18 def person1(x, y, r):
19     head1(x, y, r)
20     eye1(x, y)
21     mouth1(x, y)
22     leg1(x, y)
23     hand1(x, y)
```

```
32 def head1(x, y, r):
33     go_to(x, y)
34     speed(6)
35     circle(r)
```



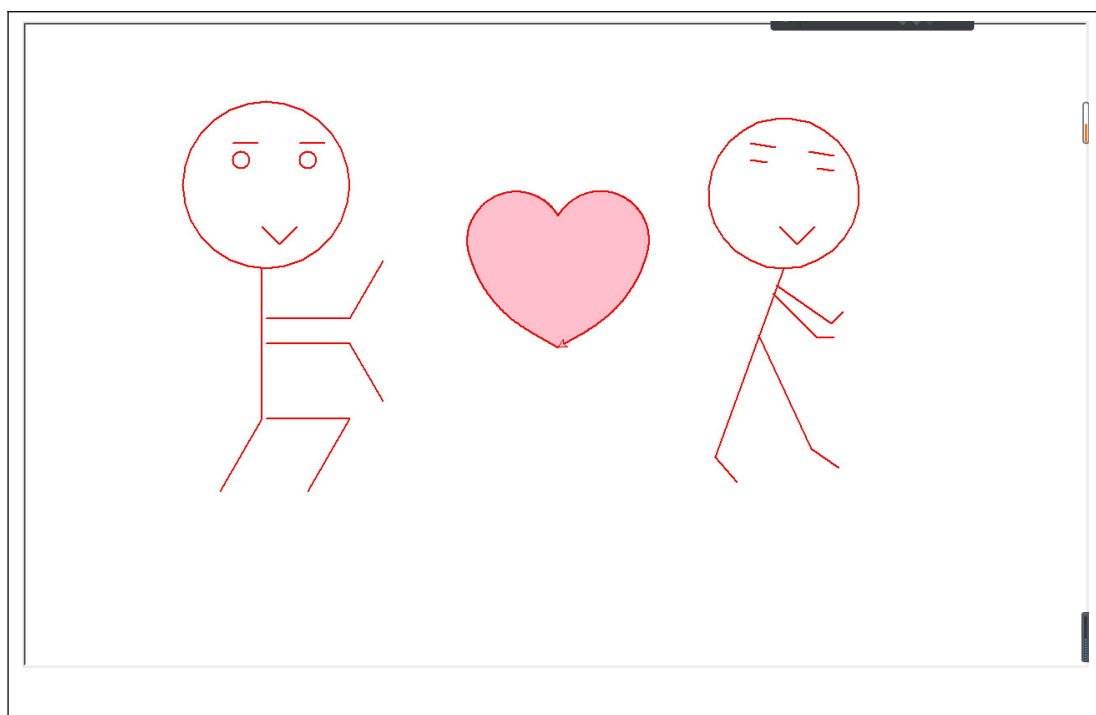
```
53 def eye1(x, y):
54     go_to(x-40, y+150)
55     right(10)
56     forward(30)
57     go_to(x-40, y+130)
58     forward(20)
59     go_to(x+30, y+140)
60     forward(30)
61     go_to(x+40, y+120)
62     forward(20)
63     left(10)
```

```
74 def mouth1(x, y):
75     go_to(x-5, y+50)
76     right(45)
77     forward(30)
78     left(90)
79     forward(30)
80     right(45)
```

```
97 def leg1(x, y):
98     go_to(x, y)
99     right(110)
100    forward(240)
101    left(60)
102    forward(40)
103    go_to(x-30, y-80)
104    right(15)
105    forward(150)
106    left(30)
107    forward(40)
```

```
123 def hand1(x, y):
124     go_to(x-9, y-20)
125     forward(80)
126     left(80)
127     forward(20)
128     go_to(x-13, y-30)
129     right(90)
130     forward(75)
131     left(45)
132     forward(20)
```

运行结果如下:



四、实验总结：

- 1、该实验是利用 `python` 编写解决问题的函数或模块，再编写主函数对自定义函数或模块进行调用。
- 2、对于实验中对函数或者模块进行调用时，注意调用语句或格式。

指导教师批阅意见：

成绩评定：

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。