python作业

注：python版本为python3

第1章

2

> 相同的值在内存中只有一份

3

> /返回一个浮点数
>
> 而//返回一个整数

4

```
import numpy as np
from random import randomint
from math import *
```

5

> pip是较常用的扩展库管理工具

6

> 每个 Python 脚本在运行时都有一个“name”属性。我们可以根据这个属性，控制python脚本是否作为模块被导入还是作为脚本程序独立运行

7

> %可以对浮点数进行求余操作

8

> 单个数字也是合法的python表达式

10

```
x = int(input("输入一个数字："))
print("百位以上的数字是：")
if x < 100:
 print("0")
else:
 print(x//100)
```

第2章

1

> 尾部进行增删操作的效率高，速度快

2

> 在python2中，返回一个包含整数的列表，在python3中，返回一个可迭代的range对象

3

```
from random import randint
x = [randint(0, 100) for i in range(1000)]

######################
dict = {}
for key in x:
 dict[key] = dict.get(key, 0) + 1
print(dict)


######################
from collections import Counter
result = Counter(x)
print(result)
```

4

> False

5

```
x = eval(input("请输入一个列表："))
m, n = eval(input("请输入两个下标："))
print(x[m:n+1])
```

6

> None

7

> remove()

8

> [6, 7, 9, 11]

9

```
dic = {1:"ni", 2:"hao", 3:"ma", 4:"?"}
key = int(input("请输入键："))
print(dic.get(key, "您输入的键不存在！"))
```

10

```
from random import randint
randomlist = [randint(0, 100) for i in range(20)]
print(randomlist)
randomlist.sort()
randomlist[10:] = randomlist[19:9:-1]
print(randomlist)

####
弄懂深浅拷贝
```

12

> items()返回键值对列表，使用values()返回字典的值列表

13

```
dict(zip(a, b))
```

14

```
b = a[::3]
```

15

```
[5 for i in range(10)]
```

16

> 不可以用del来删除元组中的部分元素

第3章

2

```
year = int(input("请输入一个年份:"))
if year % 400 == 0 or (year % 4 == 0 and year % 100 != 0):
 print("是闰年")
else:
 print("不是闰年")
```

4

```
from random import randint
ranlist = [randint(0, 100) for i in range(50)]
for i in range(len(ranlist) - 1, -1, -1):
 if ranlist[i] % 2 != 0:
     ranlist.pop(i)
     #del ranlist[i]
print(ranlist)
```

5

```
from random import randint
ranlist = [randint(0, 100) for i in range(20)]
templist = ranlist[::2]
templist.sort(reverse = True)
ranlist[::2] = templist
print(ranlist)
```

6

```
num = int(input("请输入一个整数："))
factor = []
while num > 1:
 for i in range(num - 1):
     j = i + 2
     if num % j == 0:
         factor.append(j)
         num //= j
         break
print(factor)
```

7

```
numlist = [i for i in range(100) if i%2 == 1]

##################
print(sum(numlist))


##################
result = 0
for i in numlist:
 result += i
print(result)
```

8

```
import math
def IsPrime(num):
 #由于最小为1234，不用判断是否为1， 或2
 for i in range(2, int(math.sqrt(num) + 1)):
     if num % i == 0:
         return False
 return True


result = []
listqian = [1, 2, 3, 4]
for qian in listqian:
 listbai = [i for i in listqian if i != qian]
 for bai in listbai:
     listshi = [i for i in listbai if i!= bai]
     for shi in listshi:
         listge = [i for i in listshi if i != shi]
         for ge in listge:
```

```
                num = qian * 1000 + bai * 100 + shi * 10 + ge
                if IsPrime(num):
                    result.append(num)
print(result)
```

9

```
x = eval(input("输入x: "))
if x < 0 or x >= 20:
 y = 0
elif 0 <= x < 5:
 y = x
elif 5 <= x < 10:
 y = 3 * x - 5
elif 10 <= x < 20:
 y = 0.5 * x - 2
print(y)
```

第4章

1

```
import re
s = "i have a dream, i have an apple!"
s_t = re.sub(r'i', 'I', s)
print(s_t)
```

2

```
import re
s = "implatIon englIsh ajiIeh"
s_t = re.sub(r'\BI\B', 'i', s)
print(s_t)
```

3

```
import re
s = "This is a desk desk."
pattern = re.compile(r'\b(\w+)(\s+\1){1,}\b')
y = pattern.search(s)
s_t = pattern.sub(y.group(1), s)
print(s_t)
```

4

> 字符串长度为0和1是，默认采用驻留机制
>
> 当字符串长度大于1时，如果只含有字母、数字、下划线时，也会驻留

5

```
import re
s = "jfie efji iee fjj feiifje eje."
pattern1 = re.compile(r'\b\w{3}\b')
s_t = pattern1.findall(s)
print(s_t)
```

第5章

1

2

> 在第3章的第8题已经编写过，此处不重复编写，在此，尝试用python写欧拉筛选法筛选1至n中的素数

3

```
def count(s):
    if isinstance(s, str):
        a = b = c = d = 0
        for i in s:
            if 'A' <= i <= 'Z':
                a += 1
            elif 'a' <= i <= 'z':
                b += 1
            elif '0' <= i <= '9':
                c +=1
            else:
                d +=1
        return (a, b, c, d)
    else:
        return "输入的不为字符串"
```

4

> global

5

> None

6

> 错

7

```
def func1():
    a = 11
    print(a)

def func2():
    print(a)

a = 4
func1()
func2()
print(a)
```

在这个程序中，func1中有与全局变量同名的局部变量，而func2中，没有，两者的输出分别是11和4，说明，在func1中，局部变量隐藏了同名的全局变量

8

对（注意lambda用法，现在还不熟练，同时，复习map函数）

9

```
def func(*num):
    print(num)
    print(max(num))
    print(sum(num))
```

10

```
def mysum(*v):
    res = 0
    for i in v:
        res += i
    return res
```

11

包含yield语句的函数可以用来创建生成器

注意熟悉用法，还不太熟练

12

```
def mysorted(tlist):
    v = tlist[::1]
    for i in range(len(v) - 1):
        for j in range(i + 1, len(v)):
            if v[i] > v[j]:
                v[i], v[j] = v[j], v[i]
    return v
```

第6章

1

```
import types

#我使用的是python3，默认继承object类
class Person():
    def __init__(self, name = '', age = 18, sex = 'man'):
        self.setName(name)
        self.setAge(age)
        self.setSex(sex)


    def setName(self, name):
        if not isinstance(name, str):
            print("name must be string.")
            return
        self.__name = name


    def setAge(self, age):
        if not isinstance(age, int):
            print("age must be integer.")
            return
        self.__age = age


    def setSex(self, sex):
        if sex != 'man' and sex != "woman":
            print("sex must be man or woman")
            return
        self.__sex = sex


    def show(self):
        print(self.__name, self.__age, self.__sex, sep='\n')


class Student(Person):
    def __init__(self, name='', age=30, sex='man', major='Computer'):
        super(Student, self).__init__(name, age, sex)
        self.setMajor(major)


    def setMajor(self, major):
        if not isinstance(major, str):
            print('major must be str.')
            return
        self.__major = major


    def show(self):
        super(Student, self).show()
        print(self.__major)


if __name__ == '__main__':
    s1 = Student('caieleven', 19, 'man', 'Computer')
```

```
    s1.show()
```

2

```
class Vector:
    def __init__(self, x, y, z):
        self.__x = x
        self.__y = y
        self.__z = z


    def __add__(self, rhs):
        rx = self.__x + rhs.__x
        ry = self.__y + rhs.__y
        rz = self.__z + rhs.__z
        result = Vector(rx, ry, rz)
        return result


    def __sub__(self, rhs):
        rx = self.__x - rhs.__x
        ry = self.__y - rhs.__y
        rz = self.__z - rhs.__z
        result = Vector(rx, ry, rz)
        return result


    def __mul__(self, rhs):
        rx = self.__x * rhs.__x
        ry = self.__y * rhs.__y
        rz = self.__z * rhs.__z
        result = Vector(rx, ry, rz)
        return result


    def __truediv__(self, rhs):
        rx = self.__x / rhs.__x
        ry = self.__y / rhs.__y
        rz = self.__z / rhs.__z
        result = Vector(rx, ry, rz)
        return result


    def show(self):
        print((self.__x, self.__y, self.__z))

if __name__ == '__main__':
    v1 = Vector(1, 1, 1)
    v1.show()
    v2 = Vector(2, 2, 2)
    v2.show()
    v3 = v1 + v2
    v3.show()
    v4 = v1 - v2
    v4.show()
    v5 = v1 * v2
    v5.show()
```

```
    v6 = v1 / v2
    v6.show()
```

3

oop三要素：封装、继承、多态

4

前面两个下划线（没记错的话，叫dunder），与c++中的私有成员类似，之不过，在python中，通过特殊手段可以访问，是类的伪私有成员

前面一个下划线，没有什么特殊含义，只不过可以影响模块带入测试代码见E2中的6.4temp.py

前后两个下划线，是系统定义的

后面带下划线，没有特殊含义，可以方便命名

5

```
__pow__()
__floordiv__()
```

6

`a._A__value`