36.（共12.5分）任务要求：分别计算给定字符串中数字、大写和小写字母的数量。

########代码开始########

num\_digit = 0 # 记录数字字符个数

num\_capital = 0 # 记录大写字符个数

num\_lowercase = 0 # 记录小写字符个数

str = 'Hello Python from 1990!'

for c \_\_\_\_【1】\_\_\_\_ str:

if c.\_\_\_\_【2】\_\_\_\_():

num\_digit += 1

elif c.\_\_\_\_【3】\_\_\_\_():

num\_lowercase += 1

elif c.\_\_\_\_【4】\_\_\_\_():

num\_capital += 1

# 输出数字、大写字符、小写字符的个数

print("数字个数%d, 大写字符个数%d, 小写字符个数%d" \_\_\_\_【5】\_\_\_\_ (num\_digit, num\_capital, num\_lowercase))

######## 代码结束########

\_\_\_\_【1】\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_【2】\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_【3】\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_【4】\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_【5】\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

37.（共12.5分）任务要求：完成OrderItem类的定义。

完成后的OrderItem类，在通过下列方式进行调用时：

item = OrderItem(1001, 'RedmiBook Pro 14增强版', 5500.0, 2)

print(item)

输出的结果应该为：订单项：1001, 名称：RedmiBook Pro 14增强版, 单价：5500.0, 数量：2, 总价：11000.00

########代码开始########

class OrderItem:

# 构造函数

def \_\_\_\_【1】\_\_\_\_(self, itemid, productname, price, quantity=1):

self.itemid = itemid # 订单项编号

self.productname = productname # 商品名称

self.price = price # 商品单价

self.quantity = quantity # 该商品购买数量

# 定义只读的订单项总价属性

@\_\_\_\_【2】\_\_\_\_

def totalprice(self):

return self.price \* self.quantity

# 以指定的文本格式输出订单项的信息，以便print函数能直接输出该信息

def \_\_str\_\_(\_\_\_\_【3】\_\_\_\_):

return "订单项：{}, 名称：{}, 单价：{}, 数量：{}, 总价：{: .2f}".\_\_\_\_【4】\_\_\_\_(

self.itemid, self.productname, self.price, self.quantity, self.\_\_\_\_【5】\_\_\_\_

)

########代码结束########

\_\_\_\_【1】\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_【2】\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_【3】\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_【4】\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_【5】\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

38.（共12.5分）

任务要求：统计某个Python代码文件中有效代码行数。

注意：空行或者只包含空格字符的行不计入有效行；只包含注释语句的行不计入有效行。

########代码开始######### 定义计算代码文件有效行数的函数

def calc\_code\_lines(file\_path):

row\_count = 0

\_\_\_\_【1】\_\_\_\_ open(file\_path, 'r', encoding='utf-8') as f:

# 一次性读取所有文本行

for line in f.\_\_\_\_【2】\_\_\_\_():

# 去除该行开头和结尾的空格

line = line.\_\_\_\_【3】\_\_\_\_()

# (1)如果该行长度为0，则不予计算

if \_\_\_\_【4】\_\_\_\_(line) == 0:

continue

# (2)如果该行以#字符开头，也不予计算

if line.\_\_\_\_【5】\_\_\_\_('#'):

continue

row\_count += 1

return row\_count