**任务活动1:**用Python编辑器打开文件“三角形面积.py”进行以下操作并保存结果。

(1)海伦公式是利用三角形的三条边的边长直接求三角形面积的公式，表达式为: 其中p是三角形的周长的一半，p=(a+b+c)/2。

(2)请填空完善该程序,输入三条边长，判断能否构成三角形。如果能构成三角形就计算该三角形的面积并输

出，否则提示不能构成三角形。

(3)编写完成后原名保存并关闭应用软件。

注:程序修改时，请把下划线及序号删除；不能删除注释语句。

**试题：**

#请删除序号及下划线，填写正确的代码，使程序完善。

import math

a=float(input("请输入第一条边长："))

b=float(input("请输入第二条边长："))

c=float(input("请输入第三条边长："))

if(a+b>c)and(a+c>b)and(\_\_\_①\_\_\_):

p=\_\_\_②\_\_\_

s=math.sqrt(p\*(p-a)\*(p-b)\*(p-c))

print(a,b,c,"能构成三角形！","三角形的面积为:",\_\_\_③\_\_\_)

\_\_\_④\_\_\_:

print(a,b,c,"不能构成三角形！")

**任务活动2:**用Python编辑器打开文件“猜数.py”，进行以下操作并保存结果。

(1)请填空完善该程序，实现功能:随机产生一个0-99内的整数，在给定的次数内进行猜数。如果猜的数大了，提示“大了”，如果猜的数小了，提示“小了”，如果猜对了，提示“恭喜你，答对了!”,并给出猜的次数。如果在给出猜测的次数内没有猜对，提示“机会用完了，再来一次!”。

(2)编写完成后原名保存并关闭应用软件。

注：程序修改时，请把下划线及序号删除；不能删除注释语句。

**试题：**

# 程序初始化，并随机产生一个0-99的整数作为秘密数，

import \_\_①\_\_ # 导入随机数模块

import math

secret=random.randint(0,99)

guess=0

tries = \_\_②\_\_ # 尝试次数赋初值

#给定猜测次数n

n=6

# 打招呼

print('嗨，你好！今天我们来玩一个猜数游戏。')

print('这个数字在0－99之间，我给你6次机会。')

#得到玩家猜的数，并判断大小，最多允许n次

while tries<n and guess!=secret:

\_\_③\_\_ # 尝试次数加1

guess = int(input("请输入你猜的数："))

if guess > secret:

print("大了")

elif guess < secret:

print("小了")

else:

print("恭喜你,答对了！猜了", \_\_④\_\_, "次，")

#如果给出猜测的次数内没有猜对，则提示机会用完了，再来一次。

if tries == n and guess != \_\_⑤\_\_:

print("机会用完了，再来一次！")

**任务活动3:**打开文件“打的计费.py”进行以下操作并保存。

(1)请填空完善该程序，实现功能: 2公里(包括2公里)内，起步价为5元，超过2公里部分(不足整公里的，向上取整) ,每公里1.6元。

(2)编写完成后原名保存并关闭应用软件。

注:程序修改时，请把下划线及序号删除;不能删除注释语句。

**试题**

import \_\_①\_\_ # 引用math库

s = \_\_②\_\_(input("请输入你要去目的地距离出发点的公里数:"))

if \_\_③\_\_: # 2公里（包括2公里）内

cost = 5

else:

cost = 5+math.ceil(s-2)\*1.6

print("你的打的费用是", \_\_④\_\_, "元")

**\*\*\*任务活动4:**用Python编辑器文件“凯撒密码.py”，进行以下操作并保存结果。

(1）凯撒密码是一种移位密码，明文中的所有字母都在字母表上向后按照一个固定数目（密钥）进行偏移后被替换为密文。

(2)请填空完善该程序，实现功能:输入一串字符串(message)和一个数字(key),对这串字符进行加密（向后移位key位，只对26个英文字母加密)。

(3）编写完成后原名保存并关闭应用软件。

**试题源代码：**

import math

import random

import os

#定义加密函数，对字母进行加密，即向后移动key位，其他字符不加密。

def cipher(befmessage, key):

aftmessage = ''

for char in befmessage:

if char.isupper(): #对大写字母进行加密

code = ord('A')+(ord(char)-ord('A')+key) % 26

aftmessage = aftmessage+chr(code)

elif char.islower(): #对小写字母进行加密

code = \_\_①\_\_+(ord(char) - ord('a') + key) % 26

aftmessage = \_\_②\_\_+chr(code)

else:

aftmessage = aftmessage+char#字母以外的其他字符不进行加密

return aftmessage

#主程序

message = input('请输入明文：')

key = \_\_③\_\_(input('请输入密钥（整数）：')) # 输入数字密钥

secret = cipher(message, \_\_④\_\_)

print('加密后的密文是：', \_\_⑤\_\_)

# 结束