Python

10.28

* 函数

Id（）函数输出地址

Input（）函数默认输出为字符型 split（）为分隔符

Int（）函数强制转换为数字符 e. g a=int (input ())

print ()换行输出 用end=’’参数不换行

abs() 绝对值运算

* #用于注释

0b或0B代表二进制 e. g 0b10 ——>2

0o或0O代表八进制

0x或0X代表十六进制

* 运算符：

X\*\*Y表示幂运算 /为浮点数除法（一般保留一位小数） //为整除

* 数据：

浮点数（小数）、科学计数法（1.23\*10^9—>1.23e9、0.000012—>1.2e-5其中e前后不能为空，e后面要整数）、

浮点数有误差eg2.1-2.0==0.1—>False 、round（1.23456,2）—>1.23

* 数学模块：

a=5,b=9

print(a,b) 输出5 9（含空格）

e.g1

m, n =input (). split () #默认为字符串类型

print(int(m)+int(n)) #输出m+n的数值

不能用m, n =int(input (). split ())

# int() argument must be a string, a bytes-like object or a real number, not 'list'

e.g2

>>> b,c=3,4

>>> print(b,c,5)

3 4 5

e.g3

import math #引用函数

a=int(input())

b=int(input())

c=int(input())

s=(a+b+c)/2

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

print("边长",a,b,c,end='')

print("面积",area)

python的print自动换行

11.4

* 字符型

换行可以用‘‘‘并且输入时换行；也可以用\n换行

转义字符\a响铃 \b退格 \t横向制表符 \r回车 \f换页

字符串运算 ‘2’\*3 输出‘222’ ‘abcd’+ ‘efg’输出‘abcdefg’

* Type函数是内置函数 查询对象类型信息

>>>type(1)

<class ‘int’>

>>>type(python)

<class ‘str’>

* 布尔类型 只有True和False

布尔值可以通过逻辑运算符and or not和关系运算符< <= > >= == !=计算出来

字符串之间用比较符实际上就是比较第一个字母的[ASCII码](https://so.csdn.net/so/search?q=ASCII%E7%A0%81&spm=1001.2101.3001.7020" \t "_blank)大小 不同类型不能比较

e.g. ‘123’ <= ‘23’ 输出True

e.g. >>>1<3<5 符合语法 #等价于1<3and3<5 （关系运算先于逻辑运算）

e.g. >>> 1<5>2 True >>>1<6>8 False >>> ‘Hello’> ‘world’False

* 优先级

a\*\*b幂指函数(从右向左运算)>正负号>乘除>加减>关系符号>not>and>or

11.11

* 多变量赋值：x，y=4,8
* 交换变量：a=int(input())

b=int(input())

a, b=b, a

* If语句（用缩进的方式6个空格）

If 表达式1：

语句1

语句2

else：

语句3

print(“水费={0:.2f}”.format(y))

只有一个输出则零可以省略 小数点后两位

* For语句

for variable in 列表：

语句块

E.g. for i in[1,2,3,4]:或者for i in range(1,5):

Print(i)

(输出的时候每个数字有换行)

* range函数

range（start，stop，step）不包括stop ； step步长默认为1（即增量）

range（5）等价于range（0,5）等价于[0,1,2,3,4]等价于range（0,5,1）

* sum函数

>>sum([1,2,3,4,5])或者sum(range(1,6))

15

e.g.求阶乘:

n=int(input())

fac=1

for i in range(1,n+1):

fac=fac\*i

print(fac)

* for和列表的结合

lst=[i+5 for i in [1,2,3,4,] if i%2==1]等价于

lst=[]

for i in[1,2,3,4]:

ifi%2==1:

lst=lst+[i+5]

(只有1,3保留并更改为6,8 ; 将2,4舍弃)

e.g.1

n=int(input())

sum([1/i if i%2==1 else -1/i for i in range(1,21)])

求1-1/2+1/3…前n项的和

e.g.2

sum([1/i if i%4==1 else -1/i for i in range(1,50) if i%2==1]))

求和1-1/3+1/5-1/7…+1/49

e.g.3

n=(input())

sum([int(‘6’\*i) for i in range(1,n+1)])

求6+66+666…的和

* 格式化输出“需格式化字符串”.format(参数表)

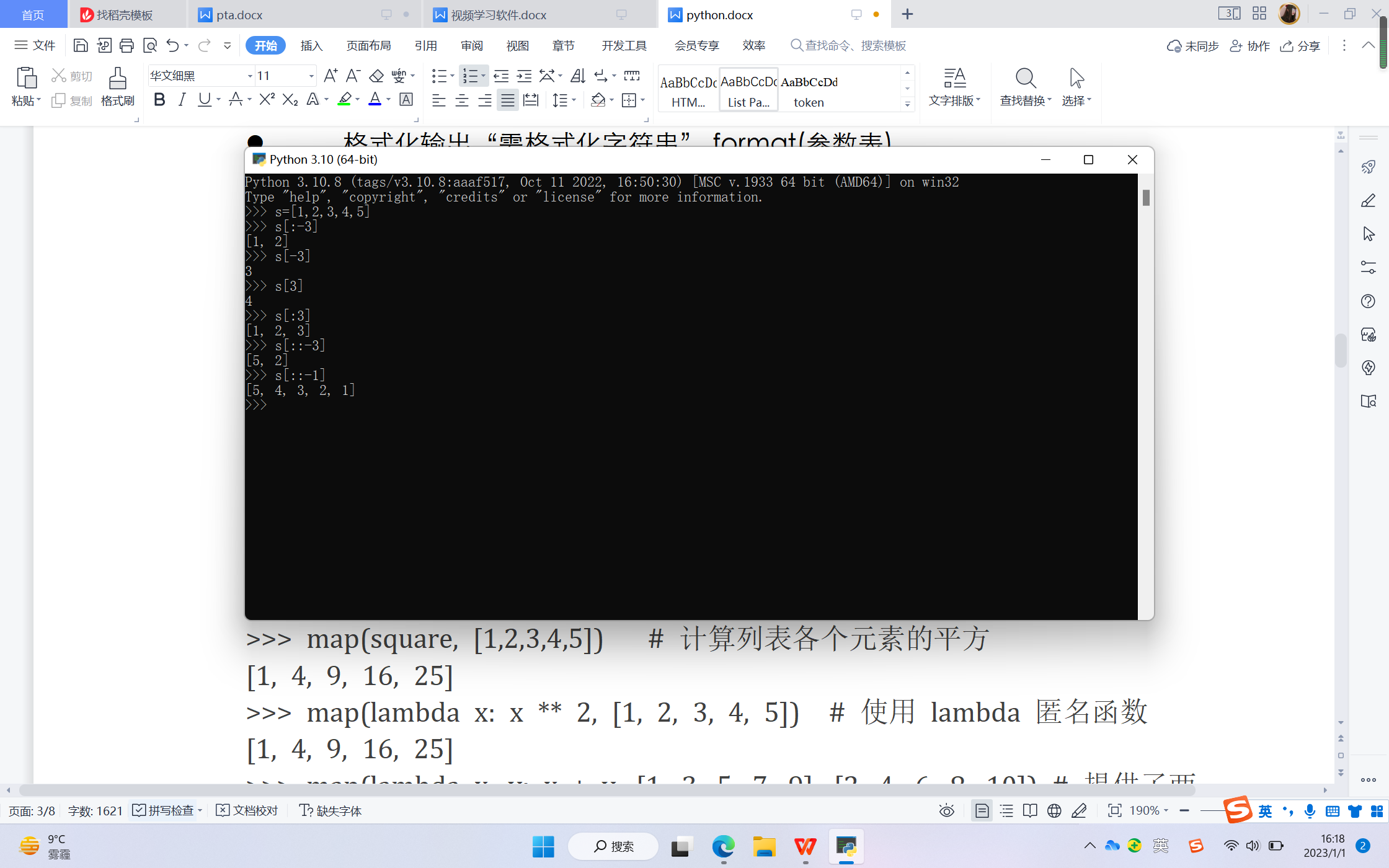
>>>name=“John”

>>>“Hello,{:>6}”.format(name) >表示右对齐 ； 6代表宽度

>>>‘Hello, John’

“{:d}”.format() 格式化成十进制数 “{:b}”.format() 格式化成二进制数

“{:o}”.format() 格式化成八进制数 “{:d}”.format() 格式化成十六进制数



* 字符串索引

1. 从左向右a[0]——a[n] 从右向左a[-1]——a[-m]ss
2. s1=[1,2,3,4,5]，则print(s1[::-1])将输出[5, 4, 3, 2, 1]
3. a = [1,2,3] b = a[:] print(b)输出[1,2,3]
4. x=5.145 print(round(x,2)) 输出5.14

* map函数

>>> map(square, [1,2,3,4,5]) # 计算列表各个元素的平方

[1, 4, 9, 16, 25]

>>> map(lambda x: x \*\* 2, [1, 2, 3, 4, 5]) # 使用 lambda 匿名函数

[1, 4, 9, 16, 25]

>>> map(lambda x, y: x + y, [1, 3, 5, 7, 9], [2, 4, 6, 8, 10]) # 提供了两个列表，对相同位置的列表数据进行相加 [3, 7, 11, 15, 19]

e.g.1 注意输入时有空格和逗号

a , b=map(int, input().split(','))

for i in range(b):

print(a, end='')

e.g.2

a,b,c=map(int,input().split())

if(a>b):

a,b=b,a

if(a>c):

a,c=c,a

if(b>c):

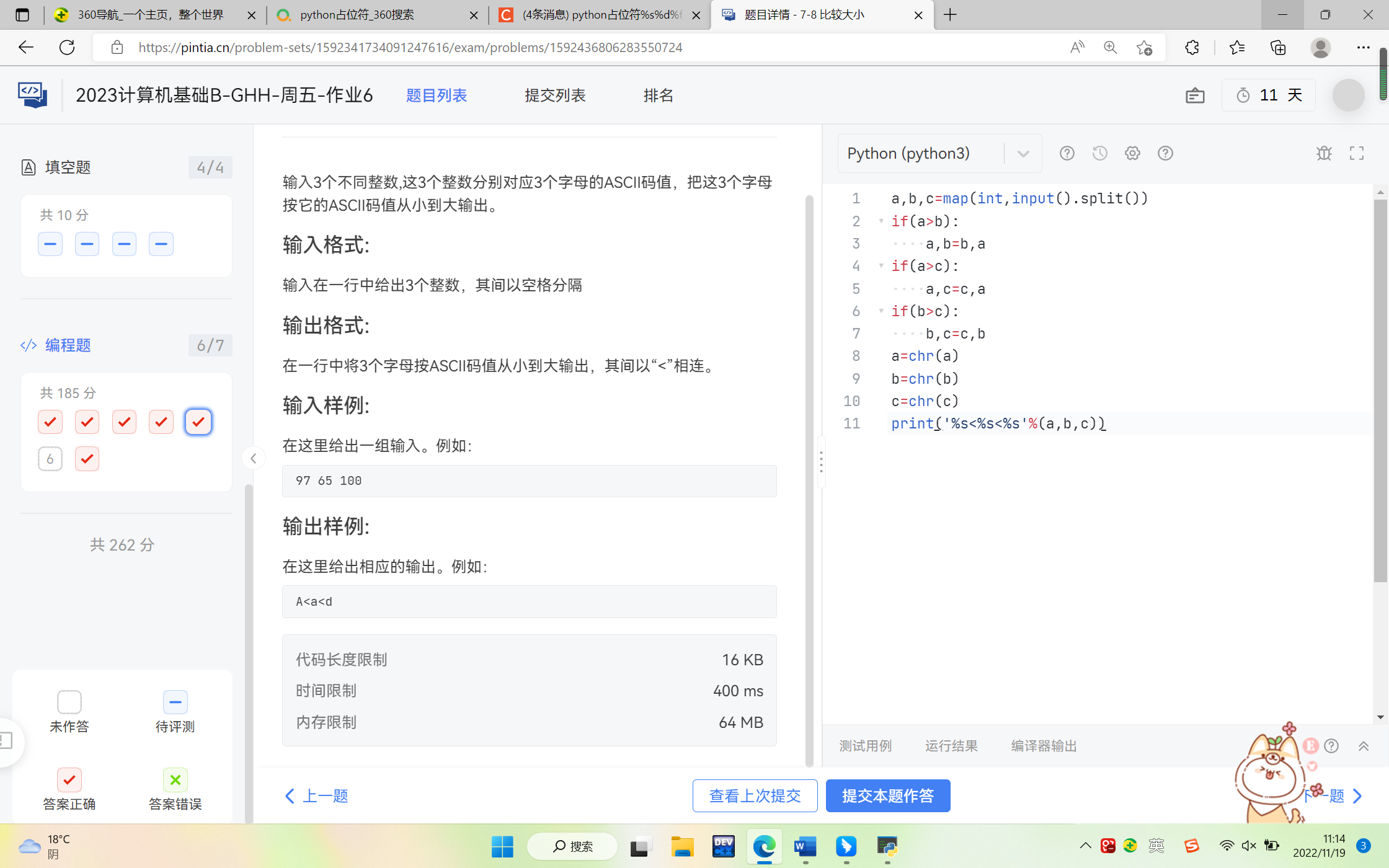
b,c=c,b

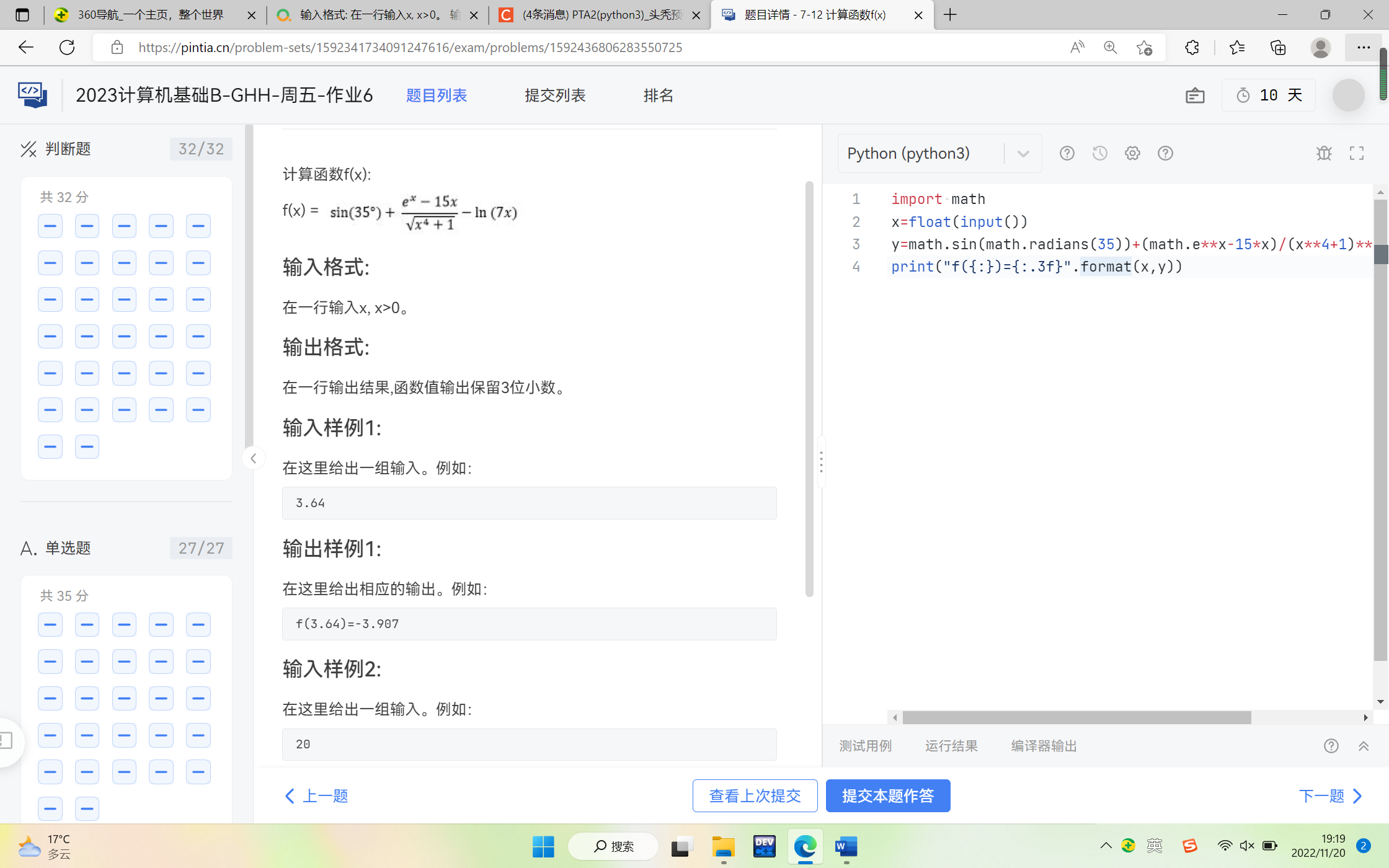
a=chr(a)

b=chr(b)

c=chr(c)

print('%s<%s<%s'%(a,b,c)) #占位符





e.g.3

import math

x=float(input())

y=math.sin(math.radians(35))+(math.e\*\*x-15\*x)/(x\*\*4+1)\*\*0.5-math.log(7\*x)

print("f({:})={:.3f}".format(x,y))

e.g.1进制转换 输出为[1,0,0,0,0]

n=int(input())

lst=[]

while n>0:

lst.insert(0,n%2)

n=n//2

print(lst)

其中list.insert(index, obj) index：插入的索引位置。obj：插入列表中的对象。

e.g.2

下面选择法排序程序的输出是 100

def selectionSort(arr):

for i in range(len(arr) - 1):

# 记录最小数的索引

minIndex = i

for j in range(i + 1, len(arr)):

if arr[j] < arr[minIndex]:

minIndex = j

if i != minIndex: # i 不是最小数时，将 i 和最小数进行交换

arr[i], arr[minIndex] = arr[minIndex], arr[i]

if i==2:print(arr[5])

return arr

selectionSort([33,6,9,7,-1,100,-32])

e.g.3

下面冒泡排序程序的输出结果是 33

def bubble\_sort(nums):

# 这个循环负责设置冒泡排序进行的次数

for i in range(len(nums) - 1):

for j in range(len(nums) - i - 1): # j为列表下标

if nums[j] > nums[j + 1]:

nums[j], nums[j + 1] = nums[j + 1], nums[j]

if i==2:print(nums[5])

return nums

bubble\_sort([45, 32, 8, 33, 12, 22, 19, 97])

e.g.3

下面程序的输出值是 26

def f(n):

result=[3,4]

for i in range(n-2):

result.append(result[-2]+result[-1]+2) #数组的扩充（加在最后）

return result[-1] #【负数】表示从后索引

print(f(5))

e.g.4

下面是用分治算法编写的程序，程序的输出是 12

def min\_max(a):

if len(a)==1:

return (a[0],a[0])

elif len(a)==2:

return (min(a),max(a))

else:

m=len(a)//2

lmin,lmax=min\_max(a[:m]) #取正着数m个数 并用到了递归

rmin,rmax=min\_max(a[m:])

return (min(lmin,rmin),max(lmax,rmax))

a,b=min\_max([3,8,9,4,10,5,1,17,9,-5])

print(a+b)

e.g.5猜生日运行下面程序，输入是10011时，输出是25

lst1=[1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31]

lst2=[2,3,6,7,10,11,14,15,18,19,22,23,26,27,30,31]

lst3=[4,5,6,7,12,13,14,15,20,21,22,23,28,29,30,31]

lst4=[8,9,10,11,12,13,14,15,24,25,26,27,28,29,30,31]

lst5=[16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31]

s=input()

print((lst1[0] if s[0]=="1" else 0)

+(lst2[0] if s[1]=="1" else 0)

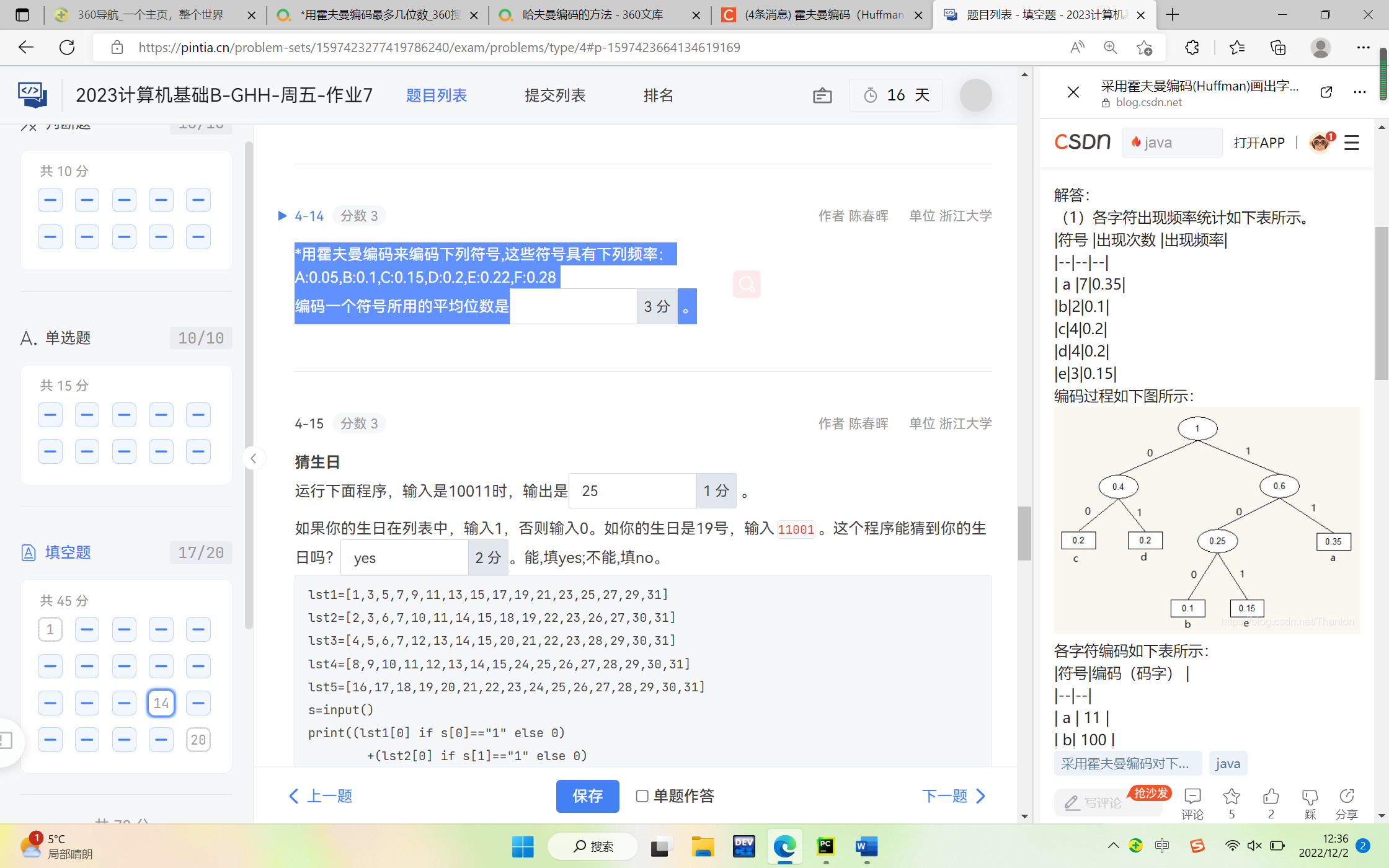
+(lst3[0] if s[2]=="1" else 0)

+(lst4[0] if s[3]=="1" else 0)

+(lst5[0] if s[4]=="1" else 0))

算法分析：每个月最多31天<2^5 用5个集合代表五个二进制 31个数字表示不同的二进制 每一位上是1则在该集合中。

e.g.6



其中平均码长用加权平均数

e.g.7在数组a=[1,2,3,...,19,20]查找数字13，调用以下算法进行二分查找，while循环一共做了几次？

def binary\_search(alist,item):

n = len(alist)

first = 0

last = n-1

while first <= last:

mid = (first + last)//2

if alist[mid] ==item:

return True

elif item < alist[mid]:

last = mid - 1

else:

first = mid + 1

return False

A.3 B.4 C.5 D.6



e.g.8