1. print("{:.1f}".format(ans))特定小数结果
2. eval（‘str’）运算字符串的公式
3. dis = [eval(x)[0]+eval(x)[1] for x in input().split()]

输入：(100,200) (50,50) (100,300) (150,50) (50,50)

dis = [eval(x)[0]+eval(x)[1] for x in input().split()]

pri = [int(x) for x in input().split()]

vau = [dis[x]/pri[x] for x in range(n)]

4.埃⽒筛法

def SieveOfEratosthenes(n, prime):

p = 2

while (p \* p <= n):

if (prime[p] == True):

for i in range(p \* 2, n+1, p):

prime[i] = False

p += 1

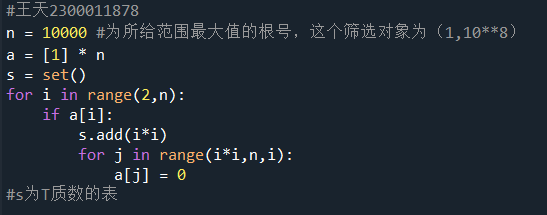
for i in range(2,n):

if a[i]:

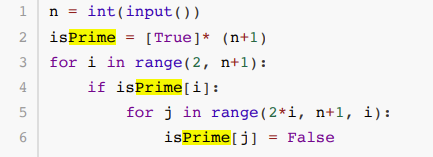
s.add(i\*i)

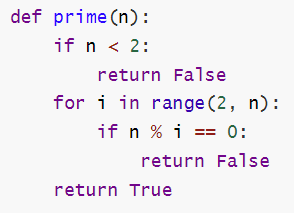
for j in range(i\*i,n,i):

a[j] = 0



质数打表





线性筛（欧拉筛）

def euler(r):

prime = [0 for i in range(r+1)]

common = []

for i in range(2, r+1):

if prime[i] == 0:

common.append(i)

for j in common:

if i\*j > r:

break

prime[i\*j] = 1

if i % j == 0:

break

return prime

1. **import** math之后math.log（真数N，底数a）就是 log a N

（1）math.sqrt(x)求根号

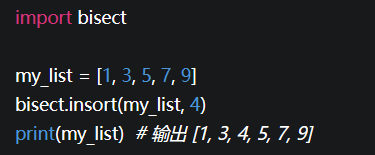
（2）math.floor(x)向下取整

（3）math.ceil(x)向上取整

（4）例如，如果您调用 re.match('hello', 'hello world')，它会返回一个匹配对象，因为字符串 'hello world' 是以 'hello' 开头的。但是如果您调用 re.match('world', 'hello world')，它会返回 None，因为字符串 'hello world' 不是以 'world' 开头的。

（5）re.findall(r'\b'+word+r'\b', article) 的意思是在给定的文章（article）中查找单词（word）的所有匹配项，并以列表输出。

（6）⼆维数组需要深度拷⻉：import copy；A=copy.deepcopy(X)

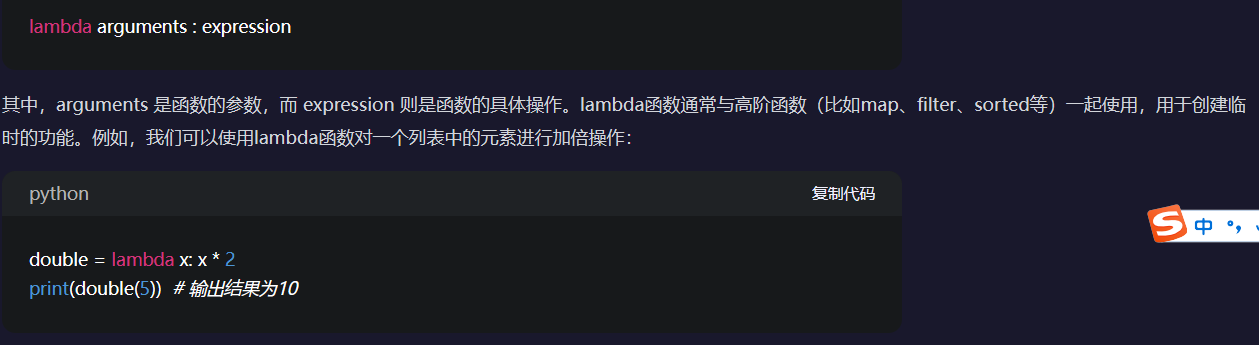
（7）

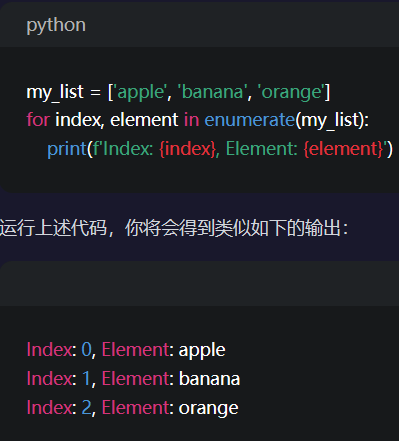
（8）bisect.bisect\_left() 函数会返回一个索引，这个索引是将要插入目标元素的位置，如果目标元素已经存在，则返回它应该插入的位置，而不是它已经存在的位置。

bisect.bisect\_right() 函数同样是返回一个索引，表示将要插入目标元素的位置，但如果目标元素已经存在，它会返回该元素右侧的插入位置。与bisect.bisect()

6. a[i] = sorted(a[i], reverse=True)

List.sort（）

1. Lambda
2. Enumerate



字符串，元组也可

1. Zip



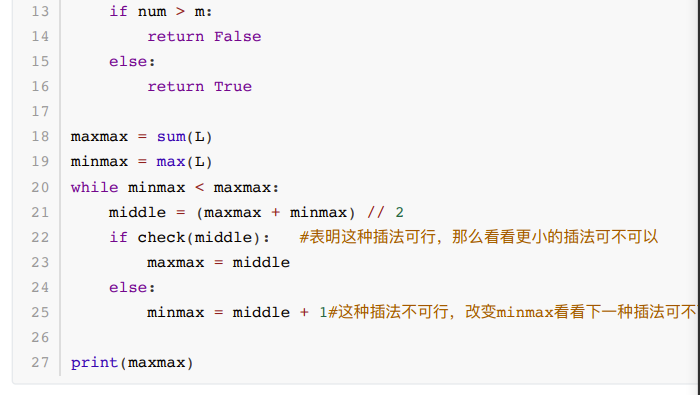
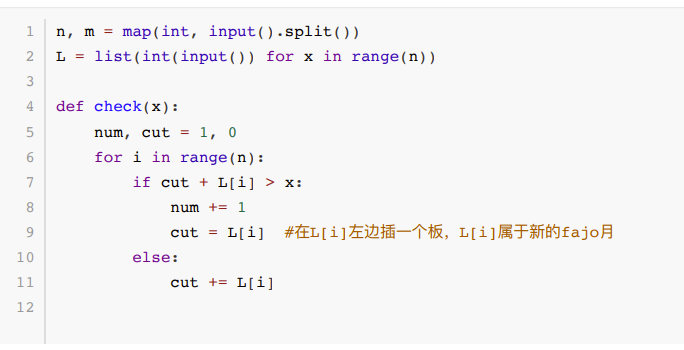
1. tt = t.copy()

tt.sort(key = lambda x: x[1])

这行代码会根据每个元素的第二个值来进行排序。

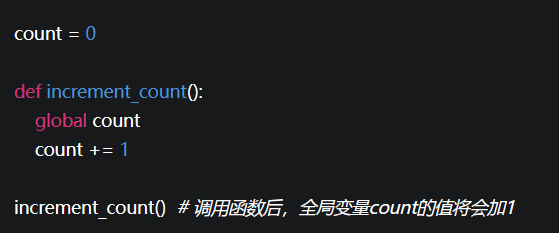
举个例子，如果 t 是一个二维列表，比如 t = [[1, 5], [2, 3], [3, 9]]，那么这行代码执行后，tt 将会成为 [[2, 3], [1, 5], [3, 9]]。这是因为根据第二个元素进行排序，元素 [2, 3] 的第二个元素最小，所以它将在排序后的列表中排在最前面。

1. print(‘\n’.join(answer)) 换行输出
2. 二分查找





2，8，16

1. Global
2. s2=s1.swapcase() 大小写互换
3. for i in t:

print('.'+i, sep='',end='')

1. 列表中元素的乘积：

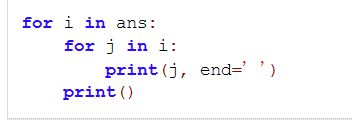
from functools import reduce

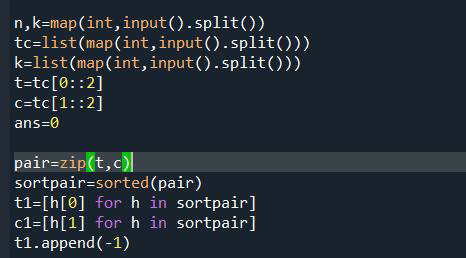
import operator

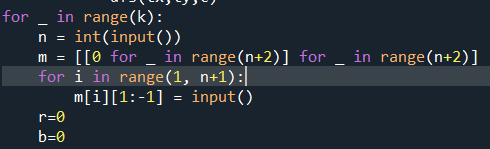
S=【】

product = reduce(operator.mul,s)

1. 输出矩阵



1. 数据绑定处理
2. 将字符串放进矩阵中



1. L.reverse（）列表倒转