. https://edu.py2fun.com/learn#/problemset/all/code/698?redirect=%2Fproblemset%2Fall%2F2

答案：

def can\_form\_palindrome(word):

    c = {}

    count = 0

    for char in word:

        if char in c:

            c[char] += 1

        else:

            c[char] = 1

    for num in c.values():

        if num % 2 == 1 :

            count += 1

            if count > 1 :

                return False

    return True

# 从用户处获取输入

word = input()

# 调用函数

print(can\_form\_palindrome(word))

2. https://edu.py2fun.com/learn#/problemset/all/code/645?redirect=%2Fproblemset%2Fall%2F1

解答：

def check\_double\_base\_palindrome(number):

    number = int(number)

    a = bin(number)[2:]

    if str(a) == str(a)[::-1] and str(number) == str(number)[::-1]:

        return True

    else:

        return False

# 获取用户输入

number = input()

# 调用函数

print(check\_double\_base\_palindrome(number))

3. https://edu.py2fun.com/learn#/problemset/all/code/699?redirect=%2Fproblemset%2Fall%2F1

解答

def list\_into\_chunks(num\_list, chunk\_size):

    lists = []

    for i in range(0,len(num\_list),chunk\_size):

        listx = num\_list[i:i + chunk\_size]

        lists.append(listx)

    return lists

# 从用户输入中获取数据，并将其转换为列表

num\_list = list(map(int, input().split()))

# 从用户输入中获取块大小

chunk\_size = int(input())

# 调用函数

print(list\_into\_chunks(num\_list, chunk\_size))

4. https://edu.py2fun.com/learn#/problemset/all/code/639?redirect=%2Fproblemset%2Fall%2F1

解答

def calculate\_sum(numbers\_list):

    co = 0

    cj = 0

    li = []

    for i in numbers\_list:

        if i % 2 == 0:

            co += i

        if i % 2 == 1:

            cj += i

    li.append(co)

    li.append(cj)

    return li

# 获取输入转为列表

numbers\_list = list(map(int,input().split()))

# 打印偶数和奇数的和

print(calculate\_sum(numbers\_list))

5. https://edu.py2fun.com/learn#/problemset/all/code/648?redirect=%2Fproblemset%2Fall%2F1

解答

def get\_unique\_elements(nested\_tuples):

    dic = {}

    li = []

    for i in nested\_tuples:

        for x in i:

            if x not in dic:

                dic[x] = 1

            else:

                dic[x] += 1

    for i, x in dic.items():

        li.append(i)

    li.sort()

    return li

# 初始化嵌套元组

nested\_tuples = []

# 获取用户输入

for \_ in range(3):

    tuple\_elements = tuple(map(int, input().split()))

    nested\_tuples.append(tuple\_elements)

# 调用函数

print(get\_unique\_elements(nested\_tuples))

6. https://edu.py2fun.com/learn#/problemset/all/code/682?redirect=%2Fproblemset%2Fall%2F1

def min\_removals\_to\_anagram(str1, str2):

    dic = {}

    coicc = 0

    for i in str1:

        if i in dic:

            dic[i] += 1

        if i not in dic:

            dic[i]  = 1

    for i in str2:

        if i in dic:

            dic[i]  -= 1

        if i not in dic:

            coicc   += 1

    for x,y in dic.items():

        if y == 0:

            pass

        if y > 0:

            coicc += y

        if y < 0:

            coicc -= y

    return coicc

# 获取输入

str1 = input()

str2 = input()

# 调用函数，输出结果

print(min\_removals\_to\_anagram(str1, str2))

下面是chatgpt给出的方法，将列表作为计数器：

def min\_removals\_to\_anagram(str1, str2):

# 初始化两个计数器

counter1 = [0] \* 26

counter2 = [0] \* 26

# 计算第一个字符串中每个字符的出现次数

for ch in str1:

counter1[ord(ch) - ord('a')] += 1

# 计算第二个字符串中每个字符的出现次数

for ch in str2:

counter2[ord(ch) - ord('a')] += 1

# 计算两个计数器之间的差异并累加到一个变量中

removals = 0

for i in range(26):

removals += abs(counter1[i] - counter2[i])

return removals

# 获取输入

str1 = input()

str2 = input()

# 调用函数，输出结果

print(min\_removals\_to\_anagram(str1, str2))

其中的ord是在获取字符的asc||的值，进而获得字符的位置。  
7. https://edu.py2fun.com/learn#/problemset/all/code/706?redirect=%2Fproblemset%2Fall%2F1

def break\_down\_list(items):

    dic = {}

    li  = []

    for i in items:

        for x,y in i.items():

            if x == 'type':

                pass

            if x == 'quantity':

                for z in range(y):

                    i['quantity'] = 1

                    li.append(i)

                else:

                    pass

    return li

# 获取物品列表

items = eval(input())

# 调用函数，输出分解后的物品列表

print(break\_down\_list(items))

8. https://edu.py2fun.com/learn#/problemset/all/code/702?redirect=%2Fproblemset%2Fall%2F2

答案

def extreme\_words\_in\_sentence(sentence):

    dic = {}

    li = sentence.split(' ')

    coicc = ''

    coic  = 0

    mcoicc = ''

    mcoic = 114514

    for i in li:

        x = len(i)

        dic[i] = x

    for x,y in dic.items():

        if coic>= y:

            pass

        else:

            coic = y

            coicc = x

    for x,y in dic.items():

        if y < mcoic:

            mcoic = y

            mcoicc = x

    tu = (coicc.lower(),mcoicc.lower())

    return tu

# 处获取输入

sentence = input()

# 调用函数

print(extreme\_words\_in\_sentence(sentence))

gpt的

def extreme\_words\_in\_sentence(sentence):

    dic = {}

    li = sentence.split(' ')

    for i in li:

        x = len(i)

        dic[i] = x

    coicc = max(dic, key=dic.get)

    mcoicc = min(dic, key=dic.get)

    tu = (coicc.lower(), mcoicc.lower())

    return tu

# 获取输入

sentence = input()

# 调用函数

print(extreme\_words\_in\_sentence(sentence))

这里使用max，与min也获取的是第一个，不用担心。