

整理了100道Python的题目，如果你是一位初学者，这一百多道题可以帮助你轻松的使用Python。初学者每天可以尝试3-5个问题，经过这一百道题的练习，要把练习昨晚并且完全懂了，基本上Python就已经入门了。如果你不是初学者，来检查一下你掌握Python的程度到了哪里！加油冲冲冲！

问题1

问题：

编写一个程序，查找所有此类数字，它们可以被7整除，但不能是5的倍数（在2000和3200之间（均包括在内））。获得的数字应以逗号分隔的顺序打印在一行上。

提示：

考虑使用范围（#begin，#end）方法。

我的解决方案：Python 3

- 使用for循环

```
l=[]
for i in range(2000, 3201):
    if (i%7==0) and (i%5!=0):
        l.append(str(i))

print ','.join(l)
```

- 使用生成器和列表理解

```
print(*(i for i in range(2000, 3201) if i%7 == 0 and i%5 != 0), sep=",")
```

问题2

问题：

编写一个程序，可以计算给定数字的阶乘，结果应以逗号分隔的顺序打印在一行上，假设向程序提供了以下输入：8然后，输出应为：40320

提示：

如果将输入数据提供给问题，则应假定它是控制台输入。

我的解决方案：Python 3

- 使用While循环

```
n = int(input()) #input() function takes input as string type
                  #int() converts it to integer type

fact = 1
i = 1
while i <= n:
    fact = fact * i;
    i = i + 1
print(fact)
```

- 使用For循环

```
n = int(input()) #input() function takes input as string type
                  #int() converts it to integer type

fact = 1
for i in range(1,n+1):
    fact = fact * i
print(fact)
```

- 使用Lambda函数

```
n = int(input())
def shortFact(x): return 1 if x <= 1 else x*shortFact(x-1)
print(shortFact(n))
```

```
while True:
    try:
        num = int(input("Enter a number: "))
        break
    except ValueError as err:
        print(err)

org = num
fact = 1
while num:
    fact = num * fact
    num = num - 1
print(f'the factorial of {org} is {fact}')
```

```
from functools import reduce

def fun(acc, item):
    return acc*item

num = int(input())
print(reduce(fun,range(1, num+1), 1))
```

问题3

问题:

使用给定的整数n，编写程序以生成包含 (i, i²) 的字典，该字典为1到n之间的整数（都包括在内）。然后程序应打印字典。假设向程序提供了以下输入：8\

然后，输出应为：

```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}
```

提示:

如果将输入数据提供给问题，则应假定它是控制台输入。考虑使用dict ()

我的解决方案：Python 3:

- 使用for循环

```
n = int(input())
ans = {}
for i in range (1,n+1):
    ans[i] = i * i
print(ans)
```

- 使用字典理解

```
n = int(input())
ans={i : i*i for i in range(1,n+1)}
print(ans)
```

```
# 演进
try:
    num = int(input("Enter a number: "))
except ValueError as err:
    print(err)

dictio = dict()
for item in range(num+1):
    if item == 0:
        continue
    else:
        dictio[item] = item * item
print(dictio)
```

```
num = int(input("Number: "))
print(dict(enumerate([i*i for i in range(1, num+1)], 1)))
```

这些是第一天解决的问题。对于基本的初学者学习者来说，上述问题非常容易。我在解决方案中展示了一些简单的编码方法。让我们看看第二天如何面对和应对新问题。

问题4

问题：

编写一个程序，该程序从控制台接受一个逗号分隔的数字序列，并生成一个列表和一个包含每个数字的元组。假设向该程序提供了以下输入：

```
34,67,55,33,12,98
```

然后，输出应为：

```
['34', '67', '55', '33', '12', '98']  
( '34', '67', '55', '33', '12', '98')
```

```
lst = input().split(',') #输入被当作字符串，因为它是字符串，所以它具有内 in  
                        # 方法名称split。','内部split函数会在找到任何','的地方进行拆  
                        分， 并将输入保存为列表到lst变量中  
  
tpl = tuple(lst)        # tuple方法将列表转换为tuple  
  
print(lst)  
print(tpl)
```

```
# 一行解决  
print(tuple(input("Enter a series of numbers separated by a comma  
:").split(',')))
```

问题5

问题：

定义一个至少具有两个方法的类：

- `getString`：从控制台输入获取字符串
- `printString`：以大写字母打印字符串

还请包括简单的测试功能来测试类方法。

提示：

****使用*init****方法构造一些参数

```
class IOstring():  
    def get_string(self):  
        self.s = input()  
  
    def print_string(self):  
        print(self.s.upper())  
  
xx = IOstring()  
xx.get_string()  
xx.print_string()
```

问题6

问题：

编写一个程序，根据给定的公式计算并打印该值：

$Q = [(2 \cdot C \cdot D) / H]$ 的平方根

以下是C和H的固定值：

C为50。H为30。

D是变量，其值应以逗号分隔的顺序输入到您的程序中，例如，让我们假设以下逗号分隔的输入顺序被赋予了程序：

100,150,180

该程序的输出应为：

18,22,24

提示：

如果接收到的输出为十进制形式，则应四舍五入至最接近的值（例如，如果接收到的输出为26.0，则应将其打印为26）。被假定为控制台输入。

```
from math import sqrt # 导入特定功能，因为使用*导入所有功能
                        # 是不好的做法

C,H = 50,30

def calc(D):
    return sqrt((2*C*D)/H)

D = [int(i) for i in input().split(',')]# 在逗号位置# 分裂和在列表中设置
D = [int(i) for i in D] #转换字符串整数
D = [calc(i) for i in D] # 返回浮点值通过在d每个项目计算方法
D = [round(i) for i in D]# 所有的浮动值是圆形的
D = [str(i) for i in D] # 所有整数被转换成字符串，以便能够应用加入操作

print(",".join(D))
```

或者

```

from math import sqrt

C,H = 50,30

def calc(D):
    return sqrt((2*C*D)/H)

D = input().split(',') # 在逗号位置#分裂和设置在列表中
D = [str(round(calc(int(i)))) for i in D] # 使用理解方法#。它按先前的代码
print(",".join(D))

```

或者

```

从 数学 进口 SQRT
ç, h = 50, 30

def calc (D) :
    返回 sqrt ( ( 2 * C * D ) / H )

打印 ("" 。加入 ( [ STR ( INT ( 计算值 ( INT ( 我 ) ) ) ) 为 我 在 输入 ( ) 。分裂 ( ' ' ) ] ) )

```

或者

```

from math import * #所有导入的数学函数
ç, h = 50, 30

C,H = 50,30

def calc(D):
    D = int(D)
    return str(int(sqrt((2*C*D)/H)))

D = input().split(',')
D = list(map(calc,D)) # 在D上应用calc函数并存储为列表
print(",".join(D))

```

```

from math import sqrt
C, H = 50, 30
mylist = input().split(',')
print(*(round(sqrt(2*C*int(D)/H)) for D in mylist), sep=",")

```

```

my_list = [int(x) for x in input('').split(',')]
C, H, x = 50, 30, []

for D in my_list:
    Q = ((2*C*D)/H)**(1/2)
    x.append(round(Q))

print(','.join(map(str, x)))

```

问题7

问题：

编写一个程序，该程序将X，Y两位数字作为输入并生成一个二维数组。数组的第i行和第j列中的元素值应为 $i \cdot j$ 。*

注意： $i = 0, 1 \dots, X-1; j = 0, 1, \dots, Y-1$ 。假设将以下输入提供给程序：3,5

然后，程序的输出应为：

```
[[0, 0, 0, 0, 0], [0, 1, 2, 3, 4], [0, 2, 4, 6, 8]]
```

提示：

注意：如果将输入数据提供给问题，则应假定它是以逗号分隔形式的控制台输入。

Python 3

```
x,y = map(int,input().split(','))
lst = []

for i in range(x):
    tmp = []
    for j in range(y):
        tmp.append(i*j)
    lst.append(tmp)

print(lst)
```

或者

```
x,y = map(int,input().split(','))
lst = [[i*j for j in range(y)] for i in range(x)]
print(lst)
```

问题8

问题：

编写一个程序，该程序接受以逗号分隔的单词序列作为输入，并在按字母顺序对单词进行排序后以逗号分隔的顺序打印这些单词。

假设将以下输入提供给程序：

```
without,hello,bag,world
```

然后，输出应为：

```
bag,hello,without,world
```

提示：

如果将输入数据提供给问题，则应假定它是控制台输入。

我的解决方案：Python 3

```
items = [x for x in raw_input().split(',')]
items.sort()
print ','.join(items)
```

```
def my_func(e):
    return e[0]

my_list = input('Enter a comma separated string: ').split(",")
my_list.sort(key=my_func)
print(",".join(my_list))
```

问题9

问题：

编写一个接受行序列作为输入的程序，并在使句子中的所有字符都大写之后打印行。

假设将以下输入提供给程序：

```
Hello world
Practice makes perfect
```

然后，输出应为：

```
HELLO WORLD
PRACTICE MAKES PERFECT
```

提示：

如果将输入数据提供给问题，则应假定它是控制台输入。

我的解决方案：Python 3


```
lst = []

while True:
    x = input()
    if len(x)==0:
        break
    lst.append(x.upper())

for line in lst:
    print(line)
```

或者

```
def user_input():
    while True:
        s = input()
        if not s:
            return
        yield s

for line in map(str.upper, user_input()):
    print(line)
```

问题10

问题

编写一个程序，该程序接受一系列由空格分隔的单词作为输入，并在删除所有重复的单词并将其按字母数字顺序排序后打印这些单词。

假设将以下输入提供给程序：

```
hello world and practice makes perfect and hello world again
```

然后，输出应为：

```
again and hello makes perfect practice world
```

提示：

***如果将输入数据提供给问题，则应假定它是控制台输入。我们使用set容器自动删除重复的数据，然后使用sorted（）对数据进行排序。

我的解决方案：Python 3

```
word = input().split()

for i in word:
    if word.count(i) > 1:      #count函数返回作为参数
        word.remove(i)      #每次调用仅删除一个元素

word.sort()
print(" ".join(word))
```

或者

```
inp_string = input("Enter string: ").split()
out_string = []
for words in inp_string:
    if words not in out_string:
        out_string.append(words)
print(" ".join(sorted(out_string)))
```

问题11

问题

编写一个程序，该程序接受以逗号分隔的4位二进制数字序列作为输入，然后检查它们是否可被5整除。被5整除的数字将以逗号分隔的顺序打印。

例子：

```
0100,0011,1010,1001
```

然后输出应为：

```
1010
```

注意：假设数据是通过控制台输入的。

提示：

如果将输入数据提供给问题，则应假定它是控制台输入。

我的解决方案：Python 3

```
def check(x):                                #转换二进制整数&返回由5零中频整除
    total, pw = 0, 1
    reversed(x)

    for i in x:
        total+=pw * (ord(i) - 48)           #ORD()函数返回ASCII值
        pw*=2
    return total % 5
```

```
data = input().split(",")          # 输入此处，并在','位置分割
lst = []

for i in data:
    if check(i) == 0:              # 如果零发现它是指由零整除并添加到列
        lst.append(i)

print(",".join(lst))
```

或者

```
def check(x):                      # 如果被5整除，则check函数返回true
    return int(x, 2) % 5 == 0      # int(x, b) 将x作为字符串，将b作为基数，
                                   # 将其转换为十进制

数据 = 输入()。分割(',')

data = list(filter(check(data)))   # 在filter(func, object)函数中，如果通
过'check'函数
print(",".join(data))             找到True，则从'data'中选取元素。
```

或者

```
data = input().split(',')
data = [num for num in data if int(num, 2) % 5 == 0]
print(','.join(data))
```

问题12

问题：

编写一个程序，查找所有介于1000和3000之间的数字（都包括在内），以使数字的每个数字均为偶数。获得的数字应以逗号分隔的顺序打印在一行上。

提示：

如果将输入数据提供给问题，则应假定它是控制台输入。

我的解决方案：Python 3

```

lst = []

for i in range(1000,3001):
    flag = 1
    for j in str(i):          # 每个整数编号i被转换成字符串

        if ord(j)%2 != 0:      # ORD返回ASCII值并且j是i
            flag = 0
    if flag == 1:
        lst.append(str(i))     # i作为字符串存储在列表中

print(",".join(lst))

```

或者

```

def check(element):
    return all(ord(i)%2 == 0 for i in element)  # 所有返回true如果所有的数字，i是即使在元件

lst = [str(i) for i in range(1000,3001)]        # 创建所有给定数字的列表，其字符串数据类型为
lst = list(filter(check,lst))                   # 如果检查条件失败，则过滤器从列表中删除元素
print(",".join(lst))

```

```

lst = [str(i) for i in range(1000,3001)]
lst = list(filter(lambda i:all(ord(j)%2 == 0 for j in i), lst))  # 使用lambda来在过滤器功能内部定义函数
print(",".join(lst))

```

问题13

问题：

编写一个接受句子并计算字母和数字数量的程序。

假设将以下输入提供给程序：

```
hello world! 123
```

然后，输出应为：

```
LETTERS 10
DIGITS 3
```

提示:

如果将输入数据提供给问题，则应假定它是控制台输入。

```
word = input()
letter,digit = 0,0

for i in word:
    if ('a'<=i and i<='z') or ('A'<=i and i<='Z'):
        letter+=1
    if '0'<=i and i<='9':
        digit+=1

print("LETTERS {0}\nDIGITS {1}".format(letter,digit))
```

或者

```
word = input()
letter, digit = 0,0

for i in word:
    if i.isalpha(): # 返回true如果字母表
        letter += 1
    elif i.isnumeric(): # 返回true如果数字
        digit += 1
print(f"LETTERS {letter}\n{digits}") # 两种解决方案均显示两种不同类型的格式化方法
```

10-13以上所有问题大多是与字符串有关的问题。解决方案的主要部分包括字符串替换函数和理解方法，以更短的形式写下代码。

问题14

问题:

编写一个接受句子的程序，并计算大写字母和小写字母的数量。

假设将以下输入提供给程序:

```
Hello world!
```

然后，输出应为:

```
UPPER CASE 1
LOWER CASE 9
```

提示:

如果将输入数据提供给问题, 则应假定它是控制台输入。

我的解决方案: Python 3

```
word = input()
upper, lower = 0, 0

for i in word:
    if 'a' <= i and i <= 'z':
        lower += 1
    if 'A' <= i and i <= 'Z':
        upper += 1

print("UPPER CASE {0}\nLOWER CASE {1}".format(upper, lower))
```

或者

```
word = input()
upper, lower = 0, 0

for i in word:
    lower += i.islower()
    upper += i.isupper()

print("UPPER CASE {0}\nLOWER CASE {1}".format(upper, lower))
```

或者

```
string = input("Enter the sentence")
upper = 0
lower = 0
for x in string:
    if x.isupper() == True:
        upper += 1
    if x.islower() == True:
        lower += 1

print("UPPER CASE: ", upper)
print("LOWER CASE: ", lower)
```

问题15

问题:

编写一个程序, 以给定的数字作为a的值来计算 $a + aa + aaa + aaaa$ 的值。

假设将以下输入提供给程序:

然后，输出应为：

11106

提示：

如果将输入数据提供给问题，则应假定它是控制台输入。

我的解决方案：Python 3

```
a = input()
total, tmp = 0, str()      # 初始化一个整数和空字符串

for i in range(4):
    tmp += a                # 串接'a'到'TMP'
    total += int(tmp)        # 将字符串类型转换为整数类型

print(total)
```

或者

```
a = input()
total = int(a) + int(2*a) + int(3*a) + int(4*a) # N*a=Na, for example a="23",
2*a="2323", 3*a="232323"
print(total)
```

```
from functools import reduce
x = input('please enter a digit:')
reduce(lambda x, y: int(x) + int(y), [x*i for i in range(1,5)])
```

问题16

问题：

使用列表推导对列表中的每个奇数取平方。该列表由逗号分隔的数字序列输入。>假设将以下输入提供给程序：

1,2,3,4,5,6,7,8,9

然后，输出应为：

1,9,25,49,81

提示：

如果将输入数据提供给问题，则应假定它是控制台输入。

我的解决方案：Python 3

```
lst = [str(int(i)**2) for i in input().split(',') if int(i) % 2]
print(",".join(lst))
```

```
seq = input().split(',')
lst = [int(i) for i in seq]
def flt(i):                                #定义一个函数
    return i % 2 != 0
result_l = [str(i * i) for i in filter(flt,lst)]
print(",".join(result_l))
```

测试用例中有一个错误，并在@dwedigital的帮助下通知并修复了解决方案的问题。我对他表示感谢。

问题17

问题：

编写一个程序，该程序根据来自控制台输入的交易日志来计算银行帐户的净额。事务日志格式如下所示：

```
D 100
W 200
```

- D表示存款，W表示提款。

假设将以下输入提供给程序：

```
D 300
D 300
W 200
D 100
```

然后，输出应为：

```
500
```

提示：

如果将输入数据提供给问题，则应假定它是控制台输入。


```
total = 0
while True:
    s = input().split()
    if not s:          # 如果字符串为空则退出
        break
    cm,num = map(str,s)    # 两个输入以cm和num的形式分布在字符串数据类型中

    if cm=='D':
        total+=int(num)
    if cm=='W':
        total-=int(num)

print(total)
```

```
lst = []
while True:
    x = input()
    if len(x)==0:
        break
    lst.append(x)

balance = 0
for item in lst:
    if 'D' in item:
        balance += int(item.strip('D '))
    if 'W' in item:
        balance -= int(item.strip('W '))
print(balance)
```

问题18

问题：

网站要求用户输入用户名和密码进行注册。编写程序以检查用户输入的密码是否有效。

以下是检查密码的标准：

- [az]之间至少1个字母
- [0-9]之间至少1个数字
- [AZ]之间至少1个字母
- [\$#@]中的至少1个字符
- 交易密码的最小长度：6
- 交易密码的最大长度：12

您的程序应接受逗号分隔的密码序列，并将根据上述条件进行检查。符合条件的密码将被打印，每个密码之间用逗号分隔。

例子

如果输入以下密码作为程序输入：

```
ABd1234@1,a F1#,2w3E*,2We3345
```

然后，程序的输出应为：

ABd1234@1

提示：

如果将输入数据提供给问题，则应假定它是控制台输入。

```
def is_low(x):                                # 如果字符串中有小写字母则返回true
    for i in x:
        if 'a'<=i and i<='z':
            return True
    return False

def is_up(x):
    for i in x:
        if 'A'<= i and i<='Z':
            return True
    return False

def is_num(x):
    for i in x:
        if '0'<=i and i<='9':
            return True
    return False

def is_other(x):
    for i in x:
        if i=='$' or i=='#' or i=='@':
            return True
    return False

s = input().split(',')
lst = []

for i in s:
    length = len(i)
    if 6 <= length and length <= 12 and is_low(i) and is_up(i) and is_num(i) and is_other(i):
        lst.append(i)

print(",".join(lst))
```

或者

```
def check(x):
    cnt = (6<=len(x) and len(x)<=12)
    for i in x:
        if i.isupper():
            cnt+=1
            break
    for i in x:
```

```

        if i.islower():
            cnt+=1
            break
    for i in x:
        if i.isnumeric():
            cnt+=1
            break
    for i in x:
        if i=='@' or i=='#' or i=='$':
            cnt+=1
            break
    return cnt == 5

s = input().split(',')
lst = filter(check,s)
print(",".join(lst))

```

或者

```

import re

s = input().split(',')
lst = []

for i in s:
    cnt = 0
    cnt+=(6<=len(i) and len(i)<=12)
    cnt+=bool(re.search("[a-z]",i))
    cnt+=bool(re.search("[A-Z]",i))
    cnt+=bool(re.search("[0-9]",i))
    cnt+=bool(re.search("[@#$]",i))
    lst.append(i)

print(",".join(lst))

```

问题19

问题：

您需要编写一个程序来按升序对（名称，年龄，分数）元组进行排序，其中名称是字符串，年龄和分数是数字。元组由控制台输入。排序标准是：

- 1：根据名称排序
- 2：然后根据年龄排序
- 3：然后按分数排序

优先级是名称> 年龄> 得分。

如果给出以下元组作为程序的输入：

```
Tom,19,80
John,20,90
Jony,17,91
Jony,17,93
Json,21,85
```

然后，程序的输出应为：

```
[('John', '20', '90'), ('Jony', '17', '91'), ('Jony', '17', '93'), ('Json', '21', '85'), ('Tom', '19', '80')]
```

提示：

如果将输入数据提供给问题，则应假定它是控制台输入。我们使用itemgetter启用多个排序键。

```
lst = []
while True:
    s = input().split(',')
    if not s[0]:
        break
    lst.append(tuple(s))

lst.sort(key= lambda x:(x[0],int(x[1]),int(x[2])))
print(lst)
```

在出现上述问题之前，我甚至不知道re（正则表达式）模块及其使用。我甚至不知道如何按多个键进行排序。为了以不同的方式解决这些问题，我不得不探索和学习那些语法。re模块中有很多有趣的内容，尽管我很难理解其中的许多内容。

问题20

问题：

用生成器定义一个类，该生成器可以迭代在给定范围0和n之间可以被7整除的数字。

假设将以下输入提供给程序：

```
7
```

然后，输出应为：

```
0
7
14
```

提示:

考虑使用类，功能和理解力。

```
class MyGen():
    def by_seven(self, n):
        for i in range(0, int(n/7) + 1):
            yield i * 7

for i in MyGen().by_seven( int(input('Please enter a number... ')) ):
    print(i)
```

```
class Divisible:

    def by_seven(self, n):
        for number in range(1,n + 1):
            if number % 7 == 0: yield number

divisible = Divisible()
generator = divisible.by_seven(int(input("Please insert a number. --> ")))
for number in generator:
    print(number)
```

问题21

问题:

机器人从原始点 (0,0) 开始在平面上移动。机器人可以按照给定的步骤向上，下，左和右移动。机器人运动的轨迹如下图所示：

```
UP 5
DOWN 3
LEFT 3
RIGHT 2
```

方向后的数字为步长。请编写一个程序，以计算一系列移动和原始点之后与当前位置的距离。如果距离是浮点数，则只需打印最接近的整数即可。*示例：* 如果给出以下元组作为程序的输入：

```
UP 5
DOWN 3
LEFT 3
RIGHT 2
```

然后，程序的输出应为：

2

提示:

如果将输入数据提供给问题，则应假定它是控制台输入。这里的距离表示欧几里得距离。导入数学模块以使用sqrt函数。

我的解决方案: Python 3

```
import math

x,y = 0,0
while True:
    s = input().split()
    if not s:
        break
    if s[0]=='UP':
        x-=int(s[1])
    if s[0]=='DOWN':
        x+=int(s[1])
    if s[0]=='LEFT':
        y-=int(s[1])
    if s[0]=='RIGHT':
        y+=int(s[1])

dist = round(math.sqrt(x**2 + y**2))
print(dist)
```

```
from math import sqrt
lst = []
position = [0,0]
while True:
    a = input()
    if not a:
        break
    lst.append(a)
for i in lst:
    if 'UP' in i:
        position[0] -= int(i.strip('UP '))
    if 'DOWN' in i:
        position[0] += int(i.strip('DOWN '))
    if 'LEFT' in i:
        position[1] -= int(i.strip('LEFT '))
    if 'RIGHT' in i:
        position[1] += int(i.strip('RIGHT '))
print(round(sqrt(position[1]**2 + position[0]**2)))
```

问题22

问题:

编写一个程序来计算输入中单词的频率。在按字母数字顺序对键进行排序后，应输出输出。

假设将以下输入提供给程序:

```
New to Python or choosing between Python 2 and Python 3? Read Python 2 or Python 3.
```

然后，输出应为:

```
2:2
3.:1
3?:1
New:1
Python:5
Read:1
and:1
between:1
choosing:1
or:2
to:1
```

提示

如果将输入数据提供给问题，则应假定它是控制台输入。

我的解决方案: Python 3

```
ss = input().split()
word = sorted(set(ss))      # split words are stored and sorted as a set

for i in word:
    print("{0}:{1}".format(i,ss.count(i)))
```

或者

```
ss = input().split()
dict = {}
for i in ss:
    i = dict.setdefault(i,ss.count(i))    # setdefault() function takes key &
    value to set it as dictionary.

dict = sorted(dict.items())                # items() function返回两个字典的键和值作为
列表

# 然后进行排序。默认情况下，在dict中以 i 的第
一键->第二键的顺序进行排序:
for i in dict:
    print("%s:%d"%(i[0],i[1]))
```

或者

```
from collections import Counter

ss = input().split()
ss = Counter(ss)          # returns key & frequency as a dictionary
ss = sorted(ss.items())   # returns as a tuple list

for i in ss:
    print("%s:%d"%(i[0],i[1]))
```

问题23

问题:

编写一种可以计算数字平方值的方法

```
n=int(input())
print(n**2)
```

问题24

问题:

Python具有许多内置函数，如果您不知道如何使用它，则可以在线阅读文档或查找一些书籍。但是Python对于每个内置函数都有一个内置文档函数。

请编写程序以打印一些Python内置函数文档，例如abs () , int () , raw_input ()

并添加您自己的功能的文档

提示:

```
The built-in document method is __doc__
```

我的解决方案: Python 3

```
print(str.__doc__)
print(sorted.__doc__)

def pow(n,p):
    """
    param n: This is any integer number
    param p: This is power over n
    return: n to the power p = n^p
    """

    return n**p
```



```
print(pow(3,4))
print(pow.__doc__)
```

问题25

问题：

定义一个类，该类具有一个类参数并且具有相同的实例参数。

提示：

Define an instance parameter, need add it in `__init__` method. You can init an object with construct parameter or set the value later

我的解决方案：Python 3

```
class Car:
    name = "Car"

    def __init__(self, name = None):
        self.name = name

honda=Car("Honda")
print("%s name is %s"%(Car.name, honda.name))

toyota=Car()
toyota.name="Toyota"
print("%s name is %s"%(Car.name, toyota.name))
```

问题26

问题：

定义一个可以计算两个数字之和的函数。

提示：

定义一个以两个数字作为参数的函数。您可以在函数中计算总和并返回。

我的解决方案：Python 3

```
sum = lambda n1,n2 : n1 + n2      # here lambda is use to define little function
as sum
print(sum(1,2))
```

问题27

问题:

定义一个可以将整数转换为字符串并在控制台中打印的函数。

提示:

使用`str ()` 将数字转换为字符串。

```
conv = lambda x : str(x)
n = conv(10)
print(n)
print(type(n))           # checks the type of the variable
```

问题28

问题:

定义一个函数，该函数可以接收两个字符串形式的整数并计算它们的总和，然后在控制台中打印它。

提示:

使用`int ()` 将字符串转换为整数。

```
def printValue(s1,s2):
    print int(s1) + int(s2)
printValue("3","4") #7
```

问题29

问题:

定义一个函数，该函数可以接受两个字符串作为输入并将其连接起来，然后在控制台中将其打印出来。

提示:

使用`+`号连接字符串。

我的解决方案: Python 3

```
sum = lambda s1,s2 : s1 + s2
print(sum("10","45"))      # 1045
```

问题30

问题：

定义一个函数，该函数可以接受两个字符串作为输入，并在控制台中打印最大长度的字符串。如果两个字符串的长度相同，则该函数应逐行打印所有字符串。

提示：

***使用len () 函数获取字符串的长度

我的解决方案：Python 3

```
def printVal(s1,s2):
    len1 = len(s1)
    len2 = len(s2)
    if len1 > len2:
        print(s1)
    elif len1 < len2:
        print(s2)
    else:
        print(s1)
        print(s2)
```

```
s1,s2=input().split()
printVal(s1,s2)
```

```
func = lambda a,b: print(max((a,b),key=len)) if len(a)!=len(b) else
print(a+'\n'+b)
```

第31题

问题：

定义一个可以打印字典的函数，其中键是1到20之间的数字（都包括在内），值是键的平方。

我的解决方案：Python 3

```
def printDict():
    dict={i:i**2 for i in range(1,21)}    # Using comprehension method and
    print(dict)

printDict()
```

第32题

问题：

定义一个可以生成字典的函数，其中键是1到20之间的数字（都包括在内），值是键的平方。该功能应仅打印键。

```
def printDict():
    d=dict()
    for i in range(1,21):
        d[i]=i**2
    for k in d.keys():
        print k
printDict()
```

第33题

问题：

定义一个函数，该函数可以生成和打印一个列表，其中值是介于1到20之间的数字的平方（均包括在内）。

```
def printList():
    lst = [i ** 2 for i in range(1, 21)]
    print(lst)

printList()
```

第34题

问题：

定义一个函数，该函数可以生成一个列表，其中值是介于1到20之间的数字的平方（均包括在内）。然后，该功能需要打印列表中的前5个元素。

```
def printList():
    lst = [i ** 2 for i in range(1, 21)]

    for i in range(5):
        print(lst[i])

printList()
```

```
def squares(n):
    squares_list = [i**2 for i in range(1,n+1)]
    print(squares_list[0:5])
squares(20)
```

```
func = lambda :print([i**2 for i in range(1,21)][:5])
```

第35题

问题:

定义一个函数，该函数可以生成一个列表，其中值是介于1到20之间的数字的平方（均包括在内）。然后，该功能需要打印列表中的最后5个元素。

```
def printList():
    lst = [i ** 2 for i in range(1, 21)]
    for i in range(19,14,-1):
        print(lst[i])

printList()
```

```
def squares(n):
    squares_list = [i**2 for i in range(1,n+1)]
    print(squares_list[-5:])
squares(20)
```

第36题

问题:

定义一个函数，该函数可以生成一个列表，其中值是介于1到20之间的数字的平方（均包括在内）。然后，该函数需要打印列表中前5个元素以外的所有值。

```
def printList():
    lst = [i ** 2 for i in range(1, 21)]
    for i in range(5,20):
        print(lst[i])

printList()
```

第37题

问题：

定义一个函数，该函数可以生成和打印一个元组，其中值是介于1到20之间的数字的平方（均包括在内）。

```
def printTuple():
    lst = [i ** 2 for i in range(1, 21)]
    print(tuple(lst))

printTuple()
```

```
def square_of_numbers():
    return tuple(i ** 2 for i in range(1, 21))

print(square_of_numbers())
```

评论

本节中的问题非常容易，所有这些都是对同类型问题的修改，主要集中在使用一些常用的函数与list，dictionary，tuple 结合使用。在我的整个解决方案中，我都没有尝试解决问题相反，我试图以一种可以解决的方式解决问题。这将有助于初学者了解如何以不同的方式解决最简单的问题。

第38题

问题：

对于给定的元组 (1,2,3,4,5,6,7,8,9,10) ，编写一个程序以一行打印前半值，然后以一行打印后半值。

提示：

使用[n1: n2]表示法从元组中获取一个切片。

```
tpl = (1,2,3,4,5,6,7,8,9,10)

for i in range(0,5):
    print(tpl[i],end = ' ')
print()
for i in range(5,10):
    print(tpl[i],end = ' ')
```

或者

```
tpl = (1,2,3,4,5,6,7,8,9,10)
lst1,lst2 = [],[]

for i in range(0,5):
    lst1.append(tpl[i])

for i in range(5,10):
    lst2.append(tpl[i])

print(lst1)
print(lst2)
```

第39题

问题：

编写程序以生成并打印另一个在给定元组 (1,2,3,4,5,6,7,8,9,10) 中其值为偶数的元组。

提示：

使用“for”来迭代元组。使用tuple () 从列表中生成一个元组。

```
tpl = (1,2,3,4,5,6,7,8,9,10)
tpl1 = tuple(i for i in tpl if i%2 == 0)
print(tpl1)
```

或者

```
tpl = (1,2,3,4,5,6,7,8,9,10)
tpl1 = tuple(filter(lambda x : x%2==0,tpl)) # Lambda function returns True if
found even element.

print(tpl1)
```

第40题

问题:

编写一个接受字符串作为输入的程序，如果该字符串是"yes"或"YES"或"Yes"，则打印"Yes"，否则打印"No"。

提示:

使用if语句判断条件。

```
text = input("Please type something. --> ")
if text == "yes" or text == "YES" or text == "Yes":
    print("Yes")
else:
    print("No")
```

```
input = input('Enter string:')
output = ''.join(['Yes' if input == 'yes' or input == 'YES' or input == 'Yes' else 'No' ])
print(str(output))
```

```
x = str(input().lower())
if x == 'yes':
    print('Yes')
else:
    print('No')
```

第41题

问题:

编写一个程序，该程序可以map () 创建一个列表，该列表的元素是[1,2,3,4,5,6,7,8,9,10]中的元素的平方。

提示:

使用map () 生成列表。使用lambda定义匿名函数。

```
li = [1,2,3,4,5,6,7,8,9,10]
squaredNumbers = map(lambda x: x**2, li) # returns map type object data
print(list(squaredNumbers))             # converting the object into list
```



```
def sqrs(item):  
    return item ** 2  
  
lst = [i for i in range(1, 11)]  
print(list(map(sqrs, lst)))
```

第42题

问题:

编写一个程序，该程序可以`map ()` 和`filter ()` 组成一个列表，该列表的元素为`[1,2,3,4,5,6,7,8,9,10]`中的偶数平方。

提示:

使用`map ()` 生成列表。使用`filter ()` 过滤列表元素。使用`lambda`定义匿名函数。

```
def even(x):  
    return x%2==0  
  
def squer(x):  
    return x*x  
  
li = [1,2,3,4,5,6,7,8,9,10]  
li = map(squer,filter(even,li)) # first filters number by even number and the  
    apply map() on the resultant elements  
print(list(li))
```

```
def even(item):  
    if item % 2 == 0:  
        return item**2  
  
lst = [i for i in range(1, 11)]  
print(list(filter(lambda j: j is not None, list(map(even, lst)))))
```

第43题

问题:

编写一个程序，该程序可以`filter ()` 创建一个列表，该列表的元素为1到20之间的偶数（均包括在内）。

提示:

使用 `filter()` 过滤列表中的元素, 使用 `lambda` 定义匿名函数。

```
def even(x):  
    return x%2==0  
  
evenNumbers = filter(even, range(1,21))  
print(list(evenNumbers))
```

第44题

问题:

编写一个程序, 该程序可以 `map()` 创建一个列表, 该列表的元素为1到20之间的数字平方 (均包括在内)。

提示:

使用 `map()` 生成一个列表。使用 `lambda` 定义匿名函数。

```
def sqr(x):  
    return x*x  
  
squaredNumbers = list(map(sqr, range(1,21)))  
print(squaredNumbers)
```

第45题

问题:

定义一个名为 `American` 的类, 该类具有一个称为 `printNationality` 的静态方法。

```
class American():  
    @staticmethod  
    def printNationality():  
        print("I am American")  
  
american = American()  
american.printNationality() # this will not run if @staticmethod does not  
                             # decorates the function.  
                             # Because the class has no instance.  
  
American.printNationality() # this will run even though the @staticmethod  
                             # does not decorate printNationality()
```

第46题

问题:

定义一个名为American的类及其子类NewYorker。

提示:

使用类Subclass (ParentClass) 定义一个子类。*

```
class American():
    pass

class NewYorker(American):
    pass

american = American()
newyorker = NewYorker()

print(american)
print(newyorker)
```

第47题

问题

定义一个名为Circle的类，该类可以由半径构造。Circle类具有一种可以计算面积的方法。

提示

使用def methodName (self) 定义一个方法。

```
class Circle():
    def __init__(self,r):
        self.radius = r

    def area(self):
        return 3.1416*(self.radius**2)

circle = Circle(5)
print(circle.area())
```

第48题

问题

定义一个名为Rectangle的类，该类可以通过长度和宽度来构造。Rectangle类具有可以计算面积的方法。

提示

使用`def methodName (self)` 定义一个方法。

```
class Rectangle():
    def __init__(self,l,w):
        self.length = l
        self.width = w

    def area(self):
        return self.length*self.width

rect = Rectangle(2,4)
print(rect.area())
```

第49题

问题

定义一个名为`Shape`的类及其子类`Square`。`Square`类具有一个`init`函数，该函数以长度作为参数。这两个类都有一个`Area`函数，该函数可以打印`Shape`的区域默认为0的区域的形状的区域。

提示

要覆盖超类中的方法，我们可以在超类中定义一个具有相同名称的方法。

```
class Shape():
    def __init__(self):
        pass

    def area(self):
        return 0

class Square(Shape):
    def __init__(self,length = 0):
        Shape.__init__(self)
        self.length = length

    def area(self):
        return self.length*self.length

Asqr = Square(5)
print(Asqr.area())      # prints 25 as given argument

print(Square().area())  # prints zero as default area
```

第50题

问题

请引发`RuntimeError`异常。

提示

U使用`raise ()` 引发异常。

解决方案:

```
raise RuntimeError('something wrong')
```

结论

嗯，上述问题似乎主要集中在面向对象程序设计的基本概念和实现上。由于这些概念并不是要解决任何功能性问题，而是设计结构，因此，这两个代码在实现部分都非常相似。

第51题

问题

编写一个函数以计算`5/0`并使用`try / except`捕获异常。

提示

使用`try / except`捕获异常。

```
def divide():  
    return 5/0  
  
try:  
    divide()  
except ZeroDivisionError as ze:  
    print("why on earth you are dividing a number by ZERO!!")  
except:  
    print("Any other exception")
```

第52题

问题

定义一个自定义异常类，该类将字符串消息作为属性。

提示

要定义自定义异常，我们需要定义一个从Exception继承的类。

```
class CustomException(Exception):
    """Exception raised for custom purpose

    Attributes:
        message -- explanation of the error
    """

    def __init__(self, message):
        self.message = message

num = int(input())

try:
    if num < 10:
        raise CustomException("Input is less than 10")
    elif num > 10:
        raise CustomException("Input is grater than 10")
except CustomException as ce:
    print("The error raised: " + ce.message)
```

第53题

问题

假设我们有一些“[username@companyname.com](#)”格式的电子邮件地址，请编写程序以打印给定电子邮件地址的用户名。用户名和公司名都仅由字母组成。

示例：如果将以下电子邮件地址作为该程序的输入：

```
john@google.com
```

然后，程序的输出应为：

```
john
```

如果将输入数据提供给问题，则应假定它是控制台输入。

提示

使用 `w` 匹配字母。

```
email = "john@google.com"
email = email.split('@')
print(email[0])
```

或者

```
import re

email = "john@google.com elise@python.com"
pattern = "(\w+)\@\w+.com"
ans = re.findall(pattern,email)
print(ans)
```

第54题

问题

假设我们有一些“[username@companyname.com](#)”格式的电子邮件地址，请编写程序以打印给定电子邮件地址的公司名称。用户名和公司名都仅由字母组成。

示例：如果将以下电子邮件地址作为该程序的输入：

```
john@google.com
```

然后，程序的输出应为：

```
google
```

如果将输入数据提供给问题，则应假定它是控制台输入。

提示

使用 `\w` 匹配字母。

```
import re

email = "john@google.com elise@python.com"
pattern = "\w+(\w+).com"
ans = re.findall(pattern,email)
print(ans)
```

第55题

问题

编写一个程序，该程序接受由空格分隔的单词序列作为输入，以仅打印由数字组成的单词。

示例：如果给出以下单词作为程序输入：

```
2 cats and 3 dogs.
```

然后，程序的输出应为：

```
['2', '3']
```

如果将输入数据提供给问题，则应假定它是控制台输入。

提示

使用`re.findall()` 使用正则表达式查找所有子字符串。

```
import re

email = input()
pattern = "\d+"
ans = re.findall(pattern,email)
print(ans)
```

或者

```
email = input().split()
ans = []
for word in email:
    if word.isdigit():      # can also use isnumeric() / isdecimal() function
        ans.append(word)
    instead
print(ans)
```

或者

```
email = input().split()
ans = [word for word in email if word.isdigit()] # using list comprehension
method
print(ans)
```

第56题

问题

打印一个unicode字符串"hello world"。

```
unicodeString = u"hello world!"
print (unicodeString)
```

第57题

问题

编写程序以读取ASCII字符串并将其转换为utf-8编码的unicode字符串。

提示

使用`unicode ()` / `encode ()` 函数进行转换。

```
s = input()
u = s.encode('utf-8')
print(u)
```

第58题

问题

编写特殊注释以指示Python源代码文件位于unicode中。

提示

使用`unicode ()` 函数进行转换。

解决方案:

```
# -*- coding: utf-8 -*-
```

第59题

问题

编写一个程序，以控制台输入给定的 n 输入来计算 $1/2 + 2/3 + 3/4 + \dots + n / n + 1$ ($n > 0$)。

示例：如果将以下 n 作为程序的输入：

5

然后，程序的输出应为：

3.55

如果将输入数据提供给问题，则应假定它是控制台输入。

```
n = int(input())
sum = 0
for i in range(1, n+1):
    sum+= i/(i+1)
print(round(sum, 2)) # rounded to 2 decimal point
```

```
def question_59(n):  
    print(round(sum(map(lambda x: x/(x+1), range(1, n+1))), 2))  
  
question_59(5)
```

第60题

问题

编写程序进行计算：

$f(n)=f(n-1)+100$ when $n>0$
and $f(0)=0$

通过控制台输入给定的 n ($n>0$)。

示例：如果将以下 n 作为程序的输入：

5

然后，程序的输出应为：

500

如果将输入数据提供给问题，则应假定它是控制台输入。

提示

我们可以在Python中定义递归函数。

```
def f(n):  
    if n == 0:  
        return 0  
    return f(n-1) + 100  
  
n = int(input())  
print(f(n))
```

```
n = int(input())  
f = lambda x: f(x-1)+100 if x > 0 else 0  
print(f(n))
```

第61题

问题

斐波那契数列是根据以下公式计算的：

```
f(n)=0 if n=0
f(n)=1 if n=1
f(n)=f(n-1)+f(n-2) if n>1
```

请编写一个程序，以控制台输入给定的n个输入来计算 $f(n)$ 的值。

示例：如果将以下n作为程序的输入：

7

然后，程序的输出应为：

13

如果将输入数据提供给问题，则应假定它是控制台输入。

提示

我们可以在Python中定义递归函数。

```
def f(n):
    if n < 2:
        return n
    return f(n-1) + f(n-2)

n = int(input())
print(f(n))
```

```
n = int(input())
f = lambda x: 0 if x == 0 else 1 if x == 1 else f(x-1)+f(x-2)
print(','.join([str(f(x)) for x in range(0, n+1)]))
```

第62题

问题

斐波那契数列是根据以下公式计算的：

```
f(n)=0 if n=0
f(n)=1 if n=1
f(n)=f(n-1)+f(n-2) if n>1
```

请编写一个程序，以控制台输入给定的n个输入来计算 $f(n)$ 的值。

示例：如果将以下n作为程序的输入：

7

然后，程序的输出应为：

0,1,1,2,3,5,8,13

如果将输入数据提供给问题，则应假定它是控制台输入。

提示

我们可以在Python中定义递归函数。使用列表推导从现有列表生成列表。使用string.join () 连接字符串列表。

```
def f(n):
    if n < 2:
        fibo[n] = n
        return fibo[n]
    fibo[n] = f(n-1) + f(n-2)
    return fibo[n]

n = int(input())
fibo = [0]*(n+1) # initialize a list of size (n+1)
f(n)             # call once and it will set value to fibo[0-n]
fibo = [str(i) for i in fibo] # converting integer data to string type
ans = ",".join(fibo) # joining all string element of fibo with ',' character
print(ans)
```

```
def fibo(n):
    if n < 2: return n
    return fibo(n-1)+fibo(n-2)

def print_fiblist(n):
    fib_list = [(str(fibo(i))) for i in range(0, n+1)]
    return print(",".join(fib_list))

n = int(input())
print_fiblist(n)
```

```
def question_62(n):
    if n == 0:
        return [0]
    if n == 1:
        return [0, 1]
    sequence = [0, 1]
    a, b = 0, 1
    for x in range(2, n+1):
        c = a + b
        sequence.append(c)
        a = b
```

```
b = c
return sequence

print(question_62(10))
```

第63题

问题

当控制台输入n时，请使用生成器编写一个程序，以逗号分隔的形式打印0到n之间的偶数。

示例：如果将以下n作为程序的输入：

10

然后，程序的输出应为：

0,2,4,6,8,10

如果将输入数据提供给问题，则应假定它是控制台输入。

提示

使用yield来生成生成器中的下一个值。

解决方案：

```
def EvenGenerator(n):
    i=0
    while i<=n:
        if i%2==0:
            yield i
        i+=1

n=int(raw_input())
values = []
for i in EvenGenerator(n):
    values.append(str(i))

print ",".join(values)
```

或者

```
# Solution by: StartZero
n = int(input())

for i in range(0, n+1, 2):
    if i < n - 1:
        print(i, end = ', ' )
    else:
        print(i)
```

第64题

问题

当控制台输入n时，请使用生成器编写一个程序，以逗号分隔的形式打印可以在0和n之间被5和7整除的数字。

示例：如果将以下n作为程序的输入：

100

然后，程序的输出应为：

0,35,70

如果将输入数据提供给问题，则应假定它是控制台输入。

提示

使用yield来生成生成器中的下一个值。

```
def generate(n):
    for i in range(n+1):
        if i % 35 == 0:      # 5*7 = 35, if a number is divisible by a & b then it
            is also divisible by a*b
            yield i

n = int(input())
resp = [str(i) for i in generate(n)]
print(",".join(resp))
```

第65题

问题

请编写assert语句以验证列表[2,4,6,8]中的每个数字都是偶数。

提示

使用“断言表达式”进行断言。

```
data = [2,4,5,6]
for i in data:
    assert i%2 == 0, "{} is not an even number".format(i)
```

第66题

问题

请编写一个从控制台接受基本数学表达式的程序，并打印评估结果。

示例：如果将以下n作为程序的输入：

35 + 3

然后，程序的输出应为：

38

提示

使用eval () 评估表达式。

```
expression = input()
ans = eval(expression)
print(ans)
```

第67题

问题

请编写一个二进制搜索功能，该功能可以搜索已排序列表中的项目。该函数应返回列表中要搜索的元素的索引。

提示

使用if / elif处理条件。

```
def binary_search(lst, item):
    low = 0
    high = len(lst) - 1

    while low <= high:
        mid = round((low + high) / 2)
```

```
    if lst[mid] == item:
        return mid
    elif lst[mid] > item:
        high = mid - 1
    else:
        low = mid + 1
return None

lst = [1,3,5,7,]
print(binary_search(lst, 9))
```

```
def binary_search_Ascending(array, target):
    lower = 0
    upper = len(array)
    print('Array Length:',upper)
    while lower < upper:
        x = (lower + upper) // 2
        print('Middle Value:',x)
        value = array[x]
        if target == value:
            return x
        elif target > value:
            lower = x
        elif target < value:
            upper = x

Array = [1,5,8,10,12,13,55,66,73,78,82,85,88,99]
print('The Value Found at Index:',binary_search_Ascending(Array, 82))
```

第68题

问题

请使用Python模块生成一个随机浮点，其值在10到100之间。

提示

使用`random.random ()`在[0,1]中生成随机浮点数。

```
import random
rand_num = random.uniform(10,100)
print(rand_num)
```

第69题

问题

请使用Python模块生成一个随机浮点，其值在5到95之间。

提示

使用`random.random ()`在`[0,1]`中生成随机浮点数。

```
import random
rand_num = random.uniform(5,95)
print(rand_num)
```

第70题

问题

请使用随机模块和列表推导编写一个程序，以输出介于0到10之间（含0和10）的随机偶数。

提示

使用`random.choice ()`对列表中的随机元素进行处理。

```
import random
resp = [i for i in range(0,11,2)]
print(random.choice(resp))
```

第71题

问题

请编写一个程序以输出一个随机数，使用随机模块和列表推导，该随机数可被5和7整除，介于10和150之间（包括10和150）。

提示

使用`random.choice ()`对列表中的随机元素进行处理。

```
import random
resp = [i for i in range(10,151) if i % 35 == 0 ]
print(random.choice(resp))
```

第72题

问题

请编写程序以生成包含5个随机数（介于100和200之间，包括100和200）的列表。

提示

使用`random.sample ()` 生成随机值列表。

```
import random
resp = random.sample(range(100,201),5)
print(resp)
```

第73题

问题

请编写一个程序以随机生成一个列表，该列表包含100至200之间（含两端）的5个偶数。

提示

使用`random.sample ()` 生成随机值列表。

```
import random
resp = random.sample(range(100,201,2),5)
print(resp)
```

第74题

问题

请编写一个程序以随机生成一个包含5个数字的列表，这些数字可以被5和7整除，且介于1和1000之间（含1和1000）。

提示

使用`random.sample ()` 生成随机值列表。

```
import random
lst = [i for i in range(1,1001) if i%35 == 0]
resp = random.sample(lst,5)
print(resp)
```

第75题

问题

请编写一个程序以随机打印7到15之间的一个整数。

提示

使用`random.randrange ()` 到给定范围内的随机整数。

解决方案:

```
import random
print random.randrange(7,16)
```

第76题

问题

请编写一个程序来压缩和解压缩字符串“hello world! hello world! hello world! hello world! ”。

提示

使用`zlib.compress ()` 和`zlib.decompress ()` 压缩和解压缩字符串。

解决方案:

```
import zlib
s = 'hello world!hello world!hello world!hello world!'
t = zlib.compress(s)
print t
print zlib.decompress(t)
```

```
s = 'hello world!hello world!hello world!hello world!'
# In Python 3 zlib.compress() accepts only DataType <bytes>
y = bytes(s, 'utf-8')
x = zlib.compress(y)
print(x)
print(zlib.decompress(x))
```

第77题

问题

请编写程序以打印“1 + 1”的执行运行时间100次。

提示

使用`timeit ()` 函数来测量运行时间。

```
import datetime

before = datetime.datetime.now()
for i in range(100):
    x = 1 + 1
after = datetime.datetime.now()
execution_time = after - before
print(execution_time.microseconds)
```

或者

```
import time

before = time.time()
for i in range(100):
    x = 1 + 1
after = time.time()
execution_time = after - before
print(execution_time)
```

第78题

问题

请编写一个程序以随机播放并打印列表[3,6,7,8]。

提示

使用`shuffle ()` 函数可以随机播放列表。

```
import random

lst = [3,6,7,8]
random.shuffle(lst)
print(lst)
```

或者

```
import random

# shuffle with a chosen seed
lst = [3,6,7,8]
seed = 7
random.Random(seed).shuffle(lst)
print(lst)
```

第79题

问题

请编写一个程序以生成所有句子，其中主语位于["I", "You"]中，动词位于["Play", "Love"]中，而宾语位于["Hockey", "Football"]中。

提示

使用list [index]表示法从列表中获取元素。

```
subjects=["I", "You"]
verbs=["Play", "Love"]
objects=["Hockey", "Football"]

for sub in subjects:
    for verb in verbs:
        for obj in objects:
            print("{} {} {}".format(sub, verb, obj))
```

```
import itertools
subject = ["I", "You"]
verb = ["Play", "Love"]
objects = ["Hockey", "Football"]

sentence = [subject, verb, objects]
n = list(itertools.product(*sentence))
for i in n:
    print(i)
```

```
from itertools import product

def question_79():
    subject = ["I", "You"]
    verb = ["Play", "Love"]
    object = ["Hockey", "Football"]
    prod = [p for p in product(range(2), repeat=3)]
    for combination in prod:
        print(f'{subject[combination[0]]} {verb[combination[1]]} {object[combination[2]]}')

question_79()
```

第80题

问题

请删除[5,6,77,45,22,12,24]中的偶数后编写程序以打印列表。

提示

使用列表推导从列表中删除一堆元素。

```
def isEven(n):
    return n%2!=0

li = [5,6,77,45,22,12,24]
lst = list(filter(isEven,li))
print(lst)
```

或者

```
li = [5,6,77,45,22,12,24]
lst = list(filter(lambda n:n%2!=0,li))
print(lst)
```

第81题

问题

通过使用列表理解，请编写一个程序以删除[12,24,35,70,88,120,155]中被5和7整除的数字，然后打印列表。

提示

使用列表推导从列表中删除一堆元素。

```
li = [12,24,35,70,88,120,155]
li = [x for x in li if x % 35!=0]
print(li)
```

第82题

问题

通过使用列表理解，请编写一个程序以删除[12,24,35,70,88,120,155]中的第0、2、4、6个数字，然后打印列表。

提示

使用列表推导从列表中删除一堆元素。使用enumerate () 获取 (索引, 值) 元组。

```
li = [12,24,35,70,88,120,155]
li = [li[i] for i in range(len(li)) if i%2 != 0 and i <= 6]
print(li)
```

```
orig_lst = [12,24,35,70,88,120,155]
indices = [0, 2, 4, 6]

new_list = [i for (j, i) in enumerate(orig_lst) if j not in indices]
print(new_list)
```

第83题

问题

通过使用列表理解，请编写一个程序以删除[12,24,35,70,88,120,155]中的第2至第4个数字，然后打印列表。

提示

***使用列表推导从列表中删除一堆元素。使用enumerate () 获取 (索引, 值) 元组。

```
#to be written
li = [12,24,35,70,88,120,155]
li = [li[i] for i in range(len(li)) if i < 3 or i > 4]
print(li)
```

```
orig_list = [12,24,35,70,88,120,155]
new_list = [i for (j, i) in enumerate(orig_list) if j not in range(1,4)]
print(new_list)
```

```
lst = [12,24,35,70,88,120,155]
print([i for i in lst if lst.index(i) not in range(2,5)])
```

第84题

问题

通过使用列表理解，请编写一个程序，生成一个3 * 5 * 8 3D数组，每个元素为0。

提示

使用列表推导来创建数组。

解决方案：

```
array = [[ [0 for col in range(8)] for col in range(5)] for row in range(3)]
print array
```

第85题

问题

通过使用列表理解，请编写一个程序来删除[12,24,35,70,88,120,155]中的第0、4、5个数字，然后打印列表。

提示

使用列表推导从列表中删除一堆元素，使用enumerate () 获取 (索引, 值) 元组。

```
li = [12,24,35,70,88,120,155]
li = [li[i] for i in range(len(li)) if i not in (0,4,5)]
print(li)
```

```
li = [12, 24, 35, 70, 88, 120, 155]
print(list(j for i, j in enumerate(li) if i != 0 and i != 4 and i != 5))
```

第86题

问题

通过使用列表理解，请在删除[12,24,35,24,88,120,155]中的值24后编写程序以打印列表。

提示

使用列表的remove方法删除一个值。

```
li = [12,24,35,24,88,120,155]
li.remove(24) # this will remove only the first occurrence of 24
print(li)
```

第87题

问题

用两个给定的列表[1,3,6,78,35,55]和[12,24,35,24,88,120,155]，编写一个程序来制作一个列表，其元素是上述给定列表的交集。

提示

使用set () 和'&='进行交集操作。

```
list1 = [1,3,6,78,35,55]
list2 = [12,24,35,24,88,120,155]
set1= set(list1)
set2= set(list2)
intersection = set1 & set2
print(intersection)
```

或者

```
list1 = [1,3,6,78,35,55]
list2 = [12,24,35,24,88,120,155]
set1= set(list1)
set2= set(list2)
intersection = set.intersection(set1,set2)
print(intersection)
```

第88题

问题

使用给定的列表`12,24,35,24,88,120,155,88,120,155`，编写一个程序，在删除所有保留原始顺序的重复值之后，打印此列表。

提示

使用`set ()` 存储许多值，不能重复。

```
li = [12,24,35,24,88,120,155,88,120,155]
for i in li:
    if li.count(i) > 1:
        li.remove(i)
print(li)
```

或者

```
def removeDuplicate( li ):
    seen = {} # dictionary
    for item in li:
        if item not in seen:
            seen[item] = True
            yield item

li = [12, 24, 35, 24, 88, 120, 155, 88, 120, 155]
ans = list(removeDuplicate(li))
print(ans)
```

第89题

问题

定义一个类`Person`及其两个子类：`Male`和`Female`。所有类都有一个方法`getGender`，可以为“男”类打印“男”，为“女”打印“女”。

提示

使用`Subclass (Parentclass)` 定义子类。

解决方案：

```
class Person(object):
    def getGender( self ):
        return "Unknown"

class Male( Person ):
    def getGender( self ):
        return "Male"

class Female( Person ):
```

```
def getGender( self ):
    return "Female"

aMale = Male()
aFemale= Female()
print aMale.getGender()
print aFemale.getGender()
```

```
class Person(object):
    def __init__(self):
        self.gender = "unknown"

    def getGender(self):
        print(self.gender)

class Male(Person):
    def __init__(self):
        self.gender = "Male"

class Female(Person):
    def __init__(self):
        self.gender = "Female"

sharon = Female()
doug = Male()
sharon.getGender()
doug.getGender()
```

第90题

问题

请编写一个程序，该程序对控制台输入的字符串中的每个字符的数量进行计数和打印。

示例：如果将以下字符串作为程序的输入给出：

```
abcdefghijkl
```

然后，程序的输出应为：

```
a,2
c,2
b,2
e,1
d,1
g,1
f,1
```

提示

使用dict来存储键/值对。使用dict.get () 方法查找具有默认值的键。

```
import string

s = input()
for letter in string.ascii_lowercase:
    cnt = s.count(letter)
    if cnt > 0:
        print("{} {}".format(letter, cnt))
```

或者

```
s = input()
for letter in range(ord('a'),ord('z')+1):    # ord() gets the ascii value of a
    letter = chr(letter)                    # chr() gets the char of an ascii
    cnt = s.count(letter)
    if cnt > 0:
        print("{} {}".format(letter, cnt))
```

```
s = 'abcdefgabc'
for i in sorted(set(s)):
    print(f'{i}, {s.count(i)}')
```

```
def character_counter(text):
    characters_list = list(text)
    char_count = {}
    for x in characters_list:
        if x in char_count.keys():
            char_count[x] += 1
        else:
            char_count[x] = 1
    return char_count

def dict_viewer(dictionary):
    for x, y in dictionary.items():
        print(f"{x},{y}")

text = input("> ")
dict_viewer(character_counter(text))
```

第91题

问题

请编写一个程序，该程序从控制台接收一个字符串，然后以相反的顺序打印它。

示例：如果以下字符串作为程序输入：*

```
rise to vote sir
```

然后，程序的输出应为：

```
ris etov ot esir
```

提示

使用`list[::-1]`以相反的顺序迭代列表。

```
s = input()
s = ''.join(reversed(s))
print(s)
```

第92题

问题

请编写一个程序，该程序从控制台接受一个字符串，并打印具有偶数索引的字符。

示例：如果将以下字符串作为程序的输入给出：

```
H1e2l3l4o5w6o7r8l9d
```

然后，程序的输出应为：

```
Helloworld
```

提示

使用`list[:: 2]`通过步骤2迭代列表。

```
s = "H1e2l3l4o5w6o7r8l9d"
s = [ s[i] for i in range(len(s)) if i%2 ==0 ]
print(''.join(s))
```

或者

```
s = "H1e2l3l4o5w6o7r8l9d"
ns = ''
for i in range(len(s)):
    if i % 2 == 0:
        ns+=s[i]
print(ns)
```

第93题

问题

请编写一个打印所有[1,2,3]排列的程序

提示

使用`itertools.permutations ()` 获取列表的排列。

解决方案:

```
import itertools
print list(itertools.permutations([1,2,3]))
```

```
from itertools import permutations

def permutation_generator(iterable):
    p = permutations(iterable)
    for i in p:
        print(i)
```

```
x = [1,2,3]
permutation_generator(x)
```

第94题

问题

编写一个程序解决经典的中国古代难题：我们在农场的鸡和兔中数出35头和94腿。我们有多少只兔子和几只鸡？

提示

使用for循环迭代所有可能的解决方案。

解决方案:

```
def solve(numheads,numlegs):
    ns='No solutions!'
    for i in range(numheads+1):
        j=numheads-i
        if 2*i+4*j==numlegs:
            return i,j
    return ns,ns

numheads = 35
numlegs = 94
solutions=solve(numheads,numlegs)
print solutions
```

```
import itertools

def animal_counter(lst):
    chickens = 0
    rabbits = 0
    for i in lst:
        if i == 2:
            chickens += 1
        elif i == 4:
            rabbits += 1
    print(f"Number of chickens is {chickens}\nNumber of rabbits is {rabbits}")

def animal_calculator(total_legs, total_heads, legs_of_each_species):
    combinations = itertools.combinations_with_replacement(legs_of_each_species,
total_heads)
    correct_combos = []
    for i in list(combinations):
        if sum(i) == total_legs:
            correct_combos.append(i)
    print(correct_combos)
    for i in correct_combos:
        animal_counter(i)

animal_calculator(94, 35, legs_of_each_species=[2,4])
```

第95题

问题

根据您在大学运动日的成绩单，您需要找到亚军得分。您会得到分数。将它们存储在列表中并找到亚军的得分。

如果给出以下字符串作为程序的输入：

```
5
2 3 6 6 5
```

然后，程序的输出应为：

提示

使分数唯一，然后找到第二名

```
n = int(input())
arr = map(int, input().split())
arr = list(set(arr))
arr.sort()
print(arr[-2])
```

```
num = int(input("Enter num: "))
L = []

while True:
    L.append(num)
    num = int(input("Enter another: "))
    if num == 0:
        break

L1 = list(set(L[:]))
L2 = sorted(L1)
print(L2)

print(f'The runner up is {L2[-2]}')
```

```
num = int(input())
scores = list(map(int, input().split(' ')))
winner = max(scores)
lst = []

if len(scores) != num:
    print('length of score is greater than input given')
else:
    for score in scores:
        if winner > score:
            lst.append(score)

runnerup = max(lst)
print(runnerup)
```

第96题

问题

给您一个字符串 S 和宽度 W 。您的任务是将字符串包装成一段宽度。

如果给出以下字符串作为程序的输入：

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
4
```

然后，程序的输出应为：

```
ABCD
EFGH
IJKL
IMNO
QRST
UVWX
YZ
```

提示

使用textwrap模块的包装功能

```
import textwrap

def wrap(string, max_width):
    string = textwrap.wrap(string,max_width)
    string = "\n".join(string)
    return string

if __name__ == '__main__':
    string, max_width = input(), int(input())
    result = wrap(string, max_width)
    print(result)
```

```
import textwrap

string = input()
width = int(input())

print(textwrap.fill(string,width))
```

第97题

问题

您将得到一个整数N。您的任务是打印一个大小为N的字母rangoli。（Rangoli是一种基于图案创建的印度民间艺术形式。）

字母rangoli的不同大小如下所示：

```
#size 3

----c----
--c-b-c--
c-b-a-b-c
--c-b-c--
----c----

#size 5

-----e-----
-----e-d-e-----
----e-d-c-d-e----
--e-d-c-b-c-d-e--
e-d-c-b-a-b-c-d-e
--e-d-c-b-c-d-e--
----e-d-c-d-e----
-----e-d-e-----
-----e-----
```

提示

首先以给定的方式打印一半的Rangoli，并将每行保存在列表中。然后以相反的顺序打印列表以获取其余信息。

```
import string
def print_rangoli(size):
    n = size
    alph = string.ascii_lowercase
    width = 4 * n - 3

    ans = []
    for i in range(n):
        left = '-'.join(alph[n - i - 1:n])
        mid = left[-1:0:-1] + left
        final = mid.center(width, '-')
        ans.append(final)

    if len(ans) > 1:
        for i in ans[n - 2::-1]:
            ans.append(i)
    ans = '\n'.join(ans)
    print(ans)

if __name__ == '__main__':
```

```
n = int(input())
print_rangoli(n)
```

```
def rangoli(n):
    # your code goes here
    l1=list(map(chr,range(97,123)))
    x=l1[n-1::-1]+l1[1:n]
    mid=len('.'.join(x))
    for i in range(1,n):
        print('.'.join(l1[n-1:n-i:-1]+l1[n-i:n]).center(mid,'-'))
    for i in range(n,0,-1):
        print('.'.join(l1[n-1:n-i:-1]+l1[n-i:n]).center(mid,'-'))
rangoli(5)
```

第98题

问题

您有个约会。您的任务是查找当天的星期几。

输入

一行输入，分别包含以MM DD YYYY格式分隔的月，日和年。

08 05 2015

输出

用大写字母输出正确的日期。

WEDNESDAY

提示

使用日历模块的工作日功能

解决方案:

```
import calendar

month, day, year = map(int, input().split())

dayId = calendar.weekday(year, month, day)
print(calendar.day_name[dayId].upper())
```

第99题

问题

给定2组整数M和N，按升序打印它们的对称差。术语'对称差'表示存在于M或N中但不存在于两者中的那些值。

输入

输入的第一行包含一个整数M，第二行包含M个空格分隔的整数，第三行包含一个整数N，第四行包含N个空格分隔的整数。

```
4
2 4 5 9
4
2 4 11 12
```

输出

以升序输出对称差整数，每行一个。

```
5
9
11
12
```

提示

使用'^'进行对称差运算。

解决方案:

```
if __name__ == '__main__':
    n = int(input())
    set1 = set(map(int, input().split()))

    m = int(input())
    set2 = set(map(int, input().split()))

    ans = list(set1 ^ set2)
    ans.sort()
    for i in ans:
        print(i)
```

第100题

问题

给你的话。有些话可能会重复。对于每个单词，输出其出现次数。输出顺序应与单词出现的输入顺序相对应。请参阅样本输入/输出以进行澄清。

如果给出以下字符串作为程序的输入：

```
4
bcdef
abcdefg
bcde
bcdef
```

然后，程序的输出应为：

```
3
2 1 1
```

提示

列出列表以获取输入顺序，并创建字典以计算单词频率

我的解决方案：Python 3

```
n = int(input())

word_list = []
word_dict = {}

for i in range(n):
    word = input()
    if word not in word_dict:
        word_list.append(word)
        word_dict[word] = word_dict.get(word, 0) + 1

print(len(word_list))
for word in word_list:
    print(word_dict[word], end=' ')
```

第101题

问题

给您一个字符串，您的任务是计算字符串字母的频率并以频率降序打印字母。

如果给出以下字符串作为程序的输入：

```
aabbbccde
```

然后，程序的输出应为：

```
b 3
a 2
c 2
d 1
e 1
```

提示

用字典计数频率并按字典项目中的值排序

我的解决方案：Python 3

```
word = input()
dct = {}
for i in word:
    dct[i] = dct.get(i,0) + 1

dct = sorted(dct.items(),key=lambda x: (-x[1],x[0]))
for i in dct:
    print(i[0],i[1])
```

```
x = input()
my_set = set(x)
arr = []
for item in my_set:
    arr.append([item,x.count(item)])
tmp = sorted(arr,key = lambda x: (-x[1],x[0]))

for i in tmp:
    print(i[0]+' '+str(i[1]))
```

第102题

问题

编写一个接受字符串并计算数字和字母数的Python程序。

输入

Hello321Bye360

输出

Digit - 6
Letter - 8

提示

使用 `isdigit ()` 和 `isalpha ()` 函数

解决方案:

```
word = input()
digit, letter = 0, 0
for char in word:
    digit += char.isdigit()
    letter += char.isalpha()

print('Digit -', digit)
print('Letter -', letter)
```

第103题

问题

给定数N. 使用递归从1到N求和

输入

5

输出

15

提示

进行递归运算以求和

解决方案:

```
def rec(n):
    if n == 0:
        return n
    return rec(n-1) + n

n = int(input())
sum = rec(n)
print(sum)
```

```
def summer(counter, n, current):  
    if n == 0:  
        return 0  
    if counter == n:  
        return current+n  
    else:  
        current = current + counter  
        counter += 1  
        return summer(counter, n, current)
```

```
N = int(input("> "))  
print(summer(1, N, 0))
```