# python习题集

> QQ：475679136制作
>
> 苏州大学计算机考研交流群：1094796091

## 第一章 python语言基础

1

```
    a,b=map(eval,input().split())
    print(a/b,a%b)
```

2

```
    name=input()
    print(len(name))
```

3

```
import math
if __name__ == '__main__':
    r,h=map(eval,input().split(' '))
    pi=3.14
    dst=20*1000
    v=pi*(r**2)*h
    print(math.ceil(dst/v))
```

一升等于1000立方厘米

4

```
    x=list(map(eval,input().split()))
    y=list(map(eval,input().split()))
    print(((x[0]-y[0])**2+(x[1]-y[1])**2)**0.5)
```

5

```
    a=random.randint(100,999)
    # a=130
    print(a)
    s=str(a).strip('0')
    ans=''
    for i in reversed(s):
        ans=ans+i
    print(int(ans))
```

注意130转置后是31

reversed只能转成list，或者使用迭代读取，直接转成str是不行的

```
    a=random.randint(100,999)
    # a=130
    print(a)
    s=str(a).strip('0')[::-1]
    print(int(s))
```

用切片是更简单的方法 str也有切片，并且str可以很方便的和int互转

strip后还是str，split之后就是list了

6

```
a=list(map(eval,list(input())))
print(a)
ans=0
for i in a:
    ans+=i
print("{0:-4}".format(ans))
```

另外format中左中右对齐分别是 <^> 很方便

7

```
 def getLen(a,b):
    return math.sqrt((a[0]-b[0])**2+(a[1]-b[1])**2)
if __name__ == '__main__':
    a=[1,2]
    b=[3,4]
    c=[5,2]
    s1=getLen(a,b)
    s2=getLen(a,c)
    s3=getLen(b,c)
    S=(s1+s2+s3)/2
    area=math.sqrt(S*(S-s1)*(S-s2)*(S-s3))
    print(S*(S-s1)*(S-s2)*(S-s3))
    print("{:<7.2f}".format(area))
```

保留两位小数，一定要写成.2f的形式，不加f格式会不正确

8

```
def f18():
    num=0
    for i in range(5):
        num=(num+100)*(1+0.005)
        print(num)
    print("{:>.5f}".format(num))
    print("{:.2%}".format(num/500))
```

9

```
    h,m,s=14,26,32
    print("{}:{}:{}".format(h,m,s))
```

10

```
def f110():
    r=random.randint(5,20)
    print("{:>15.3f}{:>15.3f}".format(r,(4/3)*math.pi*r**3))
```

11

```
def f111():
    a,b=random.randint(10,50),random.randint(10,50)
    n=complex(a,b)
    print("{:>15.2f}{:>15.2f}{:>15.2f}".format(n,abs(n),math.acos(a/abs(n))*180/math.pi))
```

12

```
def f112():
    d1=datetime.datetime(1970,1,1)
    d2=datetime.datetime.now()
    print((d2-d1).days,(d2-datetime.datetime(2020,3,6)).total_seconds()/3600)
```

输出 18327 12.201716466111112

用标准库调一下就行了..

如果不能用标准库的话，那就按照闰年和平年计算1970年1月1日到2020年1月1日的天数，然后再加上额外的天数就行了

闰年是能被400整除，或者能被4整除但不能被100整除的年份

闰年有366天

# 第二章 列表

13

```
def isPrime(n):
    if (n == 2):
        return True
    if (n % 2 == 0):
        return False
    if (n == 0 or n == 1):
        return False
    for i in range(2, math.ceil(math.sqrt(n))):
        if (n % i == 0):
            return False
    return True

def f13():
    num = [i for i in range(0, 501)]
    flag = [0 for i in range(0, 501)]
    L = []
    for ind in range(2, len(num)):
        if (flag[ind] == 1):
            continue
        if (isPrime(num[ind])):
            for i in range(ind, len(num), ind):
                flag[i] = 1
            L.append(num[ind])
    for i in range(0,len(L),5):
        print(L[i:i+5])
```

输出的时候是这样的，按照切片来输出

14

```
def f14():
    L=[35, 46, 57, 13, 24, 35, 99, 68, 13, 79, 88, 46]
    L=list(set(L))
    L.sort()
    print(L)
```

利用set去重

如果非要手写sort的话，那就写一个冒泡吧。。

```
def f14():
    L=[35, 46, 57, 13, 24, 35, 99, 68, 13, 79, 88, 46]
    L=list(set(L))
    for i in range(len(L)-1):
        flag=False
        for j in range(0,len(L)-i-1):
            if(L[j]>L[j+1]):
                L[j],L[j+1]=L[j+1],L[j]
                flag=True
        if not flag:
            break
    print(L)
```

15

```
def f15():
    s1=input()
    s2=input()
    s1=sorted(s1)
    s2=sorted(s2)
    if s1==s2 :
        return True
    else:
        return False
```

暴力sort大法

用hash也可以，题目说了字母都是小写字母，L1,L2存放两个字符串组成的字母个数，判断L1L2的值是否相同即可

```
def f15n(s,t):
    if len(s)!=len(t):
        return False
    L1=[0]*128
    L2=[0]*128
    for ch in s:
        L1[ord(ch)]+=1
        L2[ord(ch)]+=1
    return L1==L2
```

16

```
def f16():
    L=[[random.randint(0,100) for i in range(4)] for i in range(4)]
    print(L)
    L=[[L[i][j] for i in range(4)] for  j in range(4)]
    print(L)
    return L
```

17

```
def f17():
    a=[78,75]
    b=[92,67]
    c=[84,88]
    d=[50,50]
    e=[99,98]
    L={"a":a,"b":b,"c":c,"d":d,"e":e}
    L=sorted(L.items(),key=lambda item:sum(item[1]),reverse=True)
    print(L)
```

主要涉及到key的参数指定，排序的是按照L的items排序，也就是('a',[..])，指定key为list的总量，按照逆序排序

18

奇数阶魔方阵的生成方法如下：

（1） 第一个位置在第一行正中；

（2） 新位置应当处于最近一个插入位置右上方，但如右上方位置已超出方阵上边界，则新位置取应选列的最下一个位置，如超出右边界则新位置取应选行的最左一个位置；

（3） 若最近一个插入元素为N的整倍数，则选下面一行同列上的位置为新位置。

```
def f18(n):
    L = [[0] * n for i in range(n)]
    ind = [0, n // 2]
    def func(x):
        if x < 0: return n-1
        elif x >= n: return 0
        else: return x
    for i in range(1, n ** 2 + 1):
        L[ind[0]][ind[1]] = i
        if (i % n == 0):
            ind = [func(ind[0]+1),ind[1]]
        else:
            ind=[func(ind[0]-1),func(ind[1]+1)]
    print(L)
```

# 第三章 选择与循环

19

负数开根号要用cmath.sqrt()

```
def f19(a, b, c):
    if a==0:
        print("a=0,error")
    d=b ** 2 - 4 * a * c
    if(d<0):
        x1=(-b+cmath.sqrt(d))/(2*a)
        x2=(-b-cmath.sqrt(d))/(2*a)
        print("{}{}".format(x1, x2))
    elif(d==0):
        x1 = (-b ) / 2 * a
        print("{:10.5f}".format(x1))
    else:
        x1 = (-b + math.sqrt(d)) / (2 * a)
        x2 = (-b - math.sqrt(d)) / (2 * a)
        print("{:10.5f}{:10.5f}".format(x1, x2))
```

20

```
def f20(a,b,c):
    def dist(a,b):
        return math.sqrt((a[0]-b[0])**2+(a[1]-b[1])**2)
    L1=dist(a,b)
    L2=dist(a,c)
    L3=dist(b,c)
    if( max(L1,L2,L3) > L1+L2+L3-max(L1,L2,L3)):
        print("不能构成三角形")
    q=(L1+L2+L3)/2
    S=math.sqrt(q*(q-L1)*(q-L2)*(q-L3))
    print("周长是:{},面积是:{}".format((L1+L2+L3),S))
```

21

```
def f21():
    x1=float(input())
    y1=float(input())
    r=float(input())
    x2=float(input())
    y2=float(input())
    d=math.sqrt((x1-x2)**2+(y1-y2)**2)
    if(d<=r):
        print("在圆内")
    else:
        print("在圆外")
```

22

```
def f22():
    num=input()
    size=len(num)
    print('size={}'.format(size))
    for i in num:
        print(i,end=' ')
    print()
    for i in num[::-1]:
        print(i,end=' ')
```

23

```
def f23():
    L=list(map(eval,input().split(' ')))
    print(sorted(L))
```

24

```
def f24():
    num = int(input())
    rate = 0
    if num < 100000:
        rate = 0.015
    elif num < 500000:
        rate = 0.02
    elif num < 1000000:
        rate = 0.03
    else:
        rate = 0.035
    print(num+(num*rate))
```

25

```
def f25():
    ch=input()
    if( not ch.isalpha()):
        print(ch)
    elif ch.islower():
        print(ch.upper())
    else:
        print(ch.lower())
```

26

```
def f26(n):
    size=n*2-1
    for i in range(1,size+1,2):
        print(('*'*i).center(size))
```

.center返回一个序列，序列位于指定宽度的中心

27

```
def f27(n):
    for i in range(1,n+1):
        print('\t',end=str(i))
    print()
    for i in range(1,n+1):
        print(i,end='\t')
        for j in range(1,i+1):
            print(i*j,end='\t')
        print()
```

28

```
def f28(n):
    size = 2 * n - 1
    for i in range(n):
        print(('*    ' * (n + i)).center(size * 5))
    for i in range(n - 2, -1, -1):
        print(('*    ' * (n + i)).center(size * 5))
```

29

```
def f29():
    while True:
        n=int(input("Please input the number of numbers:"))
        sum = 0
        if n==0 : break
        for i in range(1,n+1):
            sum+=eval(input("Please input number {}:".format(i)))
        print("输出: sum = ",sum)
```

30

```
def f30():
    n=int(input())
    L=[i for i in range(2,n+1) if 0 not in [i%j for j in range(2,int(math.sqrt(i))+1)]]
    print(L)
```

31

```python
def f31():
    a=int(input())
    n=int(input())
    s=0.
    for i in range(1,n+1):
        s+=int(str(a)*i)
    print(int(s))
```

32

```python
def f32():
    n = int(input())
    m = int(input())
    A, B = [[0] * m for i in range(n)], [[0] * m for i in range(n)]
    for i in range(n):
        for j in range(m):
            A[i][j] = int(input())
    for i in range(n):
        for j in range(m):
            B[i][j] = int(input())
    C = [[A[i][j] + B[i][j] for j in range(m)] for i in range(n)]
    print(C)
```

# 第五章 函数

33

```python
def f33(n):
    # 8 -> 1 2 4
    # 13 -> 1  素数
    def isPrime(n):
        if n == 2:
            return True
        if (n == 1 or n == 0):
            return False
        if (n % 2 == 0):
            return False
        for i in range(2, int(math.sqrt(n)) + 1):
            if n % i == 0:
                return False
        return True
    if isPrime(n):
        return 1
    ans=[1]
    for i in range(2,int(math.sqrt(n))+1):
        if(n%i==0):
            n/=i
            ans.append(i)
    if(n!=1):   # 没被除尽，添加n
        ans.append(n)
    return int(sum(ans))
```

34

```
def f34(n):
    return int(str(n)[::-1])
```

35

```
def f35(n):
    def isPrime(n):
        if n == 2:
            return True
        if (n == 1 or n == 0):
            return False
        if (n % 2 == 0):
            return False
        for i in range(2, int(math.sqrt(n)) + 1):
            if n % i == 0:
                return False
        return True

    if not isPrime(n):
        return False
    if str(n) == str(n)[::-1]:  #如果是回文数
        return False
    if isPrime(int(str(n)[::-1])):  #如果逆置后也是素数
        return True
    return False

if __name__ == '__main__':
    ans=[]
    i=2
    while len(ans)<30:
        if(f35(i)): #如果满足条件
            ans.append(i)   #添加该数和逆置后的数
            ans.append(int(str(i)[::-1]))
        i+=1
    for i in range(0,len(ans),5):
        for j in range(i,i+5):
            print("{:>5}".format(ans[j]),end='')
        print()
```

36

```
def f37e(s,n=5):
    slow=string.ascii_lowercase
    sup=string.ascii_uppercase
    #循环往左移位
    slow=slow[n:]+slow[:n]
    sup=sup[n:]+sup[:n]
    ns=''
    i=0
    while i<len(s):
        if s[i].islower():
            ns+=slow[ord(s[i])-ord('a')]
        elif s[i].isupper():
            ns+=sup[ord(s[i])-ord('A')]
        elif s[i].isnumeric():
            t = ''
            for j in range(i,len(s)):
                if(s[j].isnumeric()):
                    t+=s[j]
                else :
                    break
            t=str(int(t)*5)
            i+=len(t)-1
            ns+=t
        i+=1
    return ns

def f37d(s,n=5):
    slow=string.ascii_lowercase
    sup=string.ascii_uppercase
    #循环往右移位
    slow=slow[-n:]+slow[:-n]
    sup=sup[-n:]+sup[:-n]

    ns=''
    i = 0
    while i < len(s):
        if s[i].islower():
            ns += slow[ord(s[i]) - ord('a')]
        elif s[i].isupper():
            ns += sup[ord(s[i]) - ord('A')]
        elif s[i].isnumeric():
            t = ''
            for j in range(i, len(s)):
                if (s[j].isnumeric()):
                    t += s[j]
                else:
                    break
            t = str(int(t) // 5)
            i += len(t) - 1
            ns += t
        i += 1
    return ns
```

第一个是加密，第二个是解密

38

```
def f38(s):
    rs=s[::-1]
    ts=rs.split(' ')
    print(ts)
    for i in range(len(ts)):
        ts[i]=ts[i][::-1]
    print(' '.join(ts))
```

直接翻转一次，切成list，再反转

39

```python
def f39(s,ind):
    ch=[]
    if(s[ind].isalpha()):
        ch.append(s[ind].lower())
        ch.append(s[ind].upper())
    else:
        ch.append(s[ind])
    L=[s.count(i) for i in ch]
    return sum(L)
```

40

```python
def f40(n):
    a,b=1,1
    i=3
    while(i<=n):
        a,b=b,a+b
        i+=1
    print(b)
```

# 第四章 字符串与正则表达式

41

```python
def f41(s):
    if(len(s)<2):
        return ''
    return s[:2]+s[-2:]
```

42

```python
def f42(s,n):
    if(n>=len(s)):
        return
    return s[:n]+s[n+1:]
```

43

同f38

```python
def f38(s):
    rs=s[::-1]
    ts=rs.split(' ')
    print(ts)
    for i in range(len(ts)):
        ts[i]=ts[i][::-1]
    print(' '.join(ts))
```

44

```
def f44(s):
    d=dict(zip(s,[0 for i in range(len(s))]))
    for ch in s:
        d[ch]+=1
    print(d)
```

45

```
def f45(s):
    a=["o","s","x","z"]
    b=["ch","sh"]
    if(s[-1]=="y"):
        return s[:-1]+'ies'
    elif s[-1] in a:
        return s+'es'
    elif s[-2:] in b:
        return s+'es'
    else:
        return s+'s'
```

46

```
def f46():
    s="This  is  very funny and   cool.Indeed!"
    a=re.sub('(\s)+',' ',s)
    b=re.sub('\.','. ',a)
    print(b)
```

47

```
def f47():
    s="<composer>Wolfgang Amadeus Mozart</composer>\n<author>Samuel
Beckett</author>\n<city>London</city>"
    L=s.split('\n')
    for line in L:
        line=re.sub(r'</\b\w+\b>','',line)  #去掉末尾，返回字符串
        line=re.sub('[<>]',' ',line)    #去掉箭头，返回字符串
        tlist=line.split()  #按空格分割，返回列表
        print(tlist[0],end=': ')    #输出标签名
        for s in tlist[1:]:
            print(s,end=' ')
        print()
```

# 第二章 序列（元组、字典与集合）

48

```
def f48():
    t = (65.5, 70.2, 100.5, 45.5, 88.8, 55.5, 73.5, 67.8)
    m=sum(t)/len(t)
    s=sum([(m-i)**2 for i in t])/len(t)
    print(math.sqrt(s))
```

49

```
def f49():
    d1={'a':1,'b':2,'c':3}
    d2={'b':2,'c':3,'d':4}
    print(set(d1) & set(d2))
```

用集合取交集

50

```
def f49():
    d1={'a':1,'b':2,'c':3}
    d2={'b':2,'c':3,'d':4}
    print(set(d1.values()) & set(d2.values()))
```

51

```
def f51():
    s=string.ascii_lowercase
    L=[(''.join([s[random.randint(0,len(s)-1)] for i in range(3)]), random.randint(0,50)) for j in range(10)]
    L.sort()    #按姓名sort
    print(L)
    L.sort(key=lambda x:(x[1],x[0]))    #按编号sort，编号相同按姓名sort
    print(L)
```

52

```
def f52():
    L1 = set(random.randint(0, 500) for i in range(random.randint(200, 300)))
    L2 = set(random.randint(0, 500) for i in range(random.randint(200, 300)))

    a=list(L1 & L2)
    b=list(L1 ^ L2)
    print(len(a))
    for i in range(0,len(a),10):
        for j in range(i,i+10 if i+10<len(a) else len(a)):
            print("{:>5}".format(a[j]),end='')
        print()
    print()
    print(len(b))
    for i in range(0, len(b), 10):
        for j in range(i, i + 10 if i + 10 < len(b) else len(b)):
            print("{:>5}".format(b[j]), end='')
        print()
```

53

```
def f53():
    A = {random.randint(0, 1000) for i in range(random.randint(1, 10))}
    B = {random.randint(0, 1000) for i in range(random.randint(1, 10))}
    AorB=A | B
    AandB=A&B
    print(AorB)
    for i in range(3):
        t=input('请输入A | B的结果(, 分割): ').split(', ')
        # print(t)
        t=set(map(eval,t))
        if(t==AorB):
            break
    print('A | B',t)
    for i in range(3):
        t = set(map(eval, input('请输入A & B的结果(，分割)： ').split(',')))
        if (t == A&B):
            break
    print('A & B',t)
```

# 第七章 文件操作

54

```
def f54():
    with open('./copy.txt')as rb,open('./new.txt','w') as wb:
        for line in rb:
            wb.write(line)
```

55

```
def f55():
    with open('./Numbers.txt') as rfb,open('./Sort.txt','w') as wfb:
        numbers=[]
        for line in rfb:
            numbers.append(float(line.strip()))
        # print(numbers)
        wfb.write(str(sorted(numbers))+'\n')
        mean=sum(numbers)/len(numbers)
        wfb.write(str(mean)+'\n')
        variance=math.sqrt(sum([(num-mean)**2 for num in numbers])/len(numbers))
        wfb.write(str(variance)+'\n')
```

56

```
def f56():
    path='./folder'
    L=[]
    for i in range(1,5):
        with open(path+'/file'+str(i)+'.txt') as fb:
            for line in fb:
                L.append(line)
    with open(path+'/merge.txt','w') as fb:
        for line in L:
            fb.write(line)
    print(L)
```

57

```
def f57():
    path='./word.txt'
    L=['a','e','i','o','u','A','E','I','O','U']
    ans=[]
    with open(path) as fb:
        for line in fb:
            if line[0] in L:
                ans.append(line)
    with open('new_word.txt','w') as fb:
        for line in ans:
            fb.write(line)
```

58

```
Aaron
Abbott
Addison
Bard
Bartholomew
Benedict
Cornell
Darnell
Edmund
Felix
Gabriel
Harlan
Lennon
```

```
def f58():
    path='./Names.txt'
    L=[]
    # 字典已经有序，使用该方法找到新字符串应该插入的位置
    def lower_bound(left,right,L,target):
        while left<right:
            mid=(left+right)//2
            if(L[mid] >= target):
                right=mid
            else:
                left=mid+1
        return left

    with open(path) as fb:
        for line in fb:
            L.append(line.strip())
        print(L)
    with open('./new_word.txt','w') as fb:
        s='Addaa'
        ind=lower_bound(0,len(L),L,s)
        print(ind)
        if(ind==len(L)):
            L.append(s)
        else:
            L.insert(ind,s)
        for line in L:
            fb.write(line+'\n')
```

59

```
from os import path
def f59():
    index=[]
    cur=0
    with open('bigfile.txt') as fb:
        for line in fb:
            index.append(cur)
            cur+=len(line)
        pos=int(input("line(1~{}) : ".format(len(index))))
        if pos not in range(1,len(index)+1):
            print('error')
        fb.seek(index[pos-1])
        print(fb.readline())
```