

Assignment2问题总结

常见输出问题

1.输入格式问题

input()默认输入的类型为字符串类型，所以题目明确了输入的数据如果是整型，实数，则应该在输入后进行类型转换，使用int(),float()方法。split()则是指定输入的格式，下面举了两个输入的例子

```
#输入以空格分割输入
#例如7-3
m,x=input().split()
#输入以,分割输入
#例如7-6
A,B=input().split(",")
```

2.输出格式问题

许多同学在输出上存在问题，在这里向大家再简单说明一下输出的格式用法

基本写法

(1) 不带编号，以format中的参数顺序输出

```
print('{} {}'.format('hello','world'))
#输出: hello world
```

(2) 带编号，可以调整输出顺序

```
print('{0} {1}'.format('hello','world'))
#输出: hello world
print('{1} {0}'.format('hello','world'))
#输出: world hello
print('{0} {1} {0}'.format('hello','world'))
#输出: hello world hello
```

带格式写法

(1) < (默认，可以不写) 左对齐

```
print('{: <10s}{: <10s}{: <10s}'.format('name','age','grade')) #10s表示间隔10个字符的位置
#输出: name      age      grade
```

(2) > 右对齐

```
print('{: >10s}{: >10s}{: >10s}'.format('name','age','grade')) #10s表示间隔10个字符的位置
#输出:          name      age      grade
```

(3) ^ 中间对齐

```
print('{:^10s}{:^10s}{:^10s}'.format('name', 'age', 'grade')) #10s表示间隔10个字符的位置
#输出:      name      age      grade
print('{:->20}'.format(99))
#输出: -----99
#冒号后面的-, 是填充字符, 可以自定义字符的类型, 比如可以写为0>20, 以0填充
```

(4) 小数取小数点后面的位数, : 号后面设置精度 (以 . 开始加上精度), 然后用 f 结束, 若不设置, 默认为精度为6, 自动四舍五入, 可带符号显示数字正负标志。

```
a = 3.1415926
print("a取小数点后面2位: {:.2f}".format(a))
print("a取小数点后面2位且右对齐: {:>10.2f}".format(a))
print("a取小数点后面2位且居中对齐: {:^10.2f}".format(a))
print("第一个数{0:.1f}, 第二个数{1:.2f}, 第三个数{2:.3f}, 冒号前面的数表示编号, 可以省略".format(a,a,a))
'''
输出:
a取小数点后面2位: 3.14
a取小数点后面2位且右对齐:      3.14
a取小数点后面2位且居中对齐:    3.14
第一个数3.1, 第二个数3.14, 第三个数3.142, 冒号前面的数表示编号, 可以省略
'''
```

3.if....else, for等语句块问题

请注意if....else中的语句格式

```
if x!=0: #不要忘了
    result=1/(2*x) # 属于if的语句块, 要缩进4个空格
else: #不要忘了
    result=0 #属于else的语句块, 也要缩进4个空格
```

```
for i in range(1,a+1):#不要忘了
    if i%2!=0: #属于for的语句块, 要缩进四个空格
        b+=1/i # 属于if的语句块, 要缩进4个空格
```

判断题

1-2/1-3

对于其他编程语言, 如C, i++/++i表示i=i+1,即让i自增1, 但由于Python中无i++/++i计算

1-17

链式赋值问题

```
x=[1,2,3,4,5]
i=0
# 链式赋值的顺序例如“a=b=c=1 顺序是:a=1,b=a,c=a”
i=x[i]=3 #这里i会先进行赋值, i=3, x[3]=3,所以x=[1,2,3,3,5]
print(x)
```

填空题

4-19

生日计算的原理解释:

```
x=float(input())
print("{:.2f}".format(4*x/3 if x<=15 else 2.5*x-17.5)) #

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))
```

这道题目利用的是二进制表示数据的范围, 5位二进制最高能表示的是11111, 即最大为31, 则它能表示的范围是[0, 31], 这里生日不可能为0, 可以去掉,

题目中有5个list, 这5个list的作用, 就是在控制每一位上的是否为1, 比如我们

以生日为31号为例, 你可以看到每个list都有31, 则说明输入的二进制为11111,

而每个lst[0]表示的是就是这个二进制对应的十进制的制, lst1[0]=1表示第1位2进制

lst2[0]=2, 表示第2位, 以此类推, 通过控制5位二进制上是否为1, 从而表示出了你的生日,

而这个控制通过你在表上看是否有你的生日从而达到判断该位是否为1还是0

再以生日19为例, 19数字转换为二进制, 则为11001, 和表中查询得到的结果一致

4-20

```
ord('张') #返回该字符对应的Unicode码
```

编程题

7-1

注意题目的输出格式

```
# 参考写法
print("sum =",sum) #注意中间有个空格
```

7-2

1.注意输入值x为实数，在input时不能将其转换为int

```
# 参考写法
x=float(input()) #输入转换为浮点数
```

2.注意if ... else 语句的书写格式

```
# 参考写法
if x!=0: #不要忘了
    result=1/(2*x) # 注意缩进
else:
    result=0
```

3.注意题目的要求输出格式，x,result均要输出，且都要保留三位小数，推荐使用format实现

```
# 参考写法
print("g({:.3f}) = {:.3f}".format(x,result))
```

7-3

1. 注意m的输入转换为int就可以，而x的输入应该转换为float

```
# 参考写法
m,x=input().split()
m=int(m)
x=float(x)
```

2. 输出格式

```
# 参考写法
print("cost = {:.2f}".format(cost))
```

7-4

1.注意题目要求的结果S是大于序列和的最小整数，比如s=2.3,则大于它的最小整数是3，这点使用题目提到的ceil做到

```
# 参考写法
import math
a=int(input())
b=0
for i in range(1,a+1):
    if i%2!=0:
        b+=1/i
b=math.ceil(b)
```

2.输出格式

```
# 参考写法
print("sum ≈",b)
```

7-6

1.注意题目的输入使用,隔开, 同时对于题目的输入数据, 注意应将A,B都先转换为int数字, 再做后面的计算

```
# 参考写法
A,B=input().split(",")
A=int(A)
B=int(B)
```

2.生成B个A, 可以考虑使用字符串乘法实现, 要使用字符串乘法, 需要将A变为字符串, 而变成字符串, 则是使用str()

```
# 参考写法
A=str(A)
result=A*B #字符串乘法
```

7-7

1.注意输入使用,隔开, 同时输入的x,y均为整数, 要做类型变换

2.将任意进制转换为10进制, 推荐使用课件中的int()

```
# 输入x,y后
x=int(x)
y=int(y)
res=int(str(x),y) #将y进制的字符串x, 变为10进制数字, 这里int中的第一个参数要为字符串类型
```

7-8

1.注意输出格式

```
# 参考写法
print("{0}<{1}<{2}".format(a,b,c) # a,b,c为要输出的三个字符
```

7-9

1.注意if...else写法是否正确，注意for 语句是否正确

2.注意输出格式，第2位数据是14位右对齐

```
# 参考写法
print('{0}{1:>14.1f}'.format(i,i*1.8+32))
```

7-10

1.注意如果是通过字符串乘法实现的，应当注意在做乘法之前，要使用str把整数a变成字符串类型

7-11

1.注意使用题目中提到的floor函数

7-12

1.注意sin函数的计算，要将35°转换位弧度制，再调用sin函数

```
# 参考写法
math.sin(math.pi*35/180)
```

2.输出格式，第一位左对齐，数据长度3位

```
# 参考写法
print("f({0:3})={1:.3f}".format(x,f))
```