

计算概论 C-py1

From 陈旭峥 北京大学智能学院

2401112120@stu.pku.edu.cn

版本: 1.0

日期: 2024 年 10 月 10 日

1 作业-01 两整数求和及 02 简单的 A*B 问题

没什么难度，只需要会基本的输入和输出即可。需要注意的是 `input()` 函数接收到输入后的类型是字符串，需要通过转换为整型或浮点型才能进行计算：

```
1 x = input() # 这时候我们如果输入一个数字进去，把它赋值给x，x的类型是字符串(str)
2 x = int(x) # 转为整型
3 x = float(x) # 转为浮点数
4 x = eval(x) # 不知道上课讲过没有，可以自己查一查
```

2 作业-03 计算三角形的周长和面积(基础版) 及 04 计算两点之间距离

首先是关于 python 的 `math` 库：一方面是各种函数的使用，如这道题里面会用到的 `sqrt`；另一方面是如何调用 python 库，一般有以下两种

```
1 x = 3
2 import math # 第一种，直接导入这个库，方法为import [某个库的名字]
3 print(math.sqrt(3)) # 输出根号3的值
4
5 from math import sqrt # 第二种，从库中导入某个函数，方法为 from [某个库的名字] import [该库中某个函数的名字]
6 print(sqrt(3)) # 输出根号3的值
```

然后是关于输出，我们经常会见到这种输出保留多少位小数的问题，第一个方法是用 `round()` 函数，大家可以自己查一查，但是 `round()` 函数会有问题，比如

```
1 print(round(3.0001, 3))
```

这个代码，输出会是 3.0，也就是如果按照 round 的要求保留到某一位之后，末尾有很多 0，它只会保留一位（就像这里的本来是想让它输出 3.000，但其实只会输出 3.0）。

更常用的是字符串的格式化输出，一般有以下几种方式：

```
1 x = 3.1415926
2 print(f"{x:.3f}") # 会输出 3.142，格式为 f"{{[变量名]}:{{[变量格式]}}"
3 print("{:.3f}".format(x)) # 会输出 3.142，格式为 "{{[变量格式]}}".
    format([变量名])
4 print("%.3f" % x) # 会输出 3.142，我没怎么用过，也建议你别用
```

3 作业-05 字符菱形

很多同学可能在这道题里第一次遇到了 Presentation Error，首先讲一下 OpenJudge 上各种错误的类型：

1. Accept：就是对了，但也不一定就真的对，因为 OpenJudge 测试的时候一般会有几组数据，你的代码可能刚好能过后台的这几个样例数据；
2. Wrong Answer：答案错了；
3. Presentation Error：答案对了，但是表示有问题；
4. Runtime Error：通常是引用了不存在的东西，比如使用了某个你没有定义的变量，或者是访问列表时超出范围等；
5. Compile Error：代码写的有错误，它没运行起来；
6. TLE：超时，MLE：超内存，OLE：超输出长度，这三个见的都不会很多，可能到学完循环之后写几个循环嵌套的时候会见到 TLE。

对于这道题来说，我们首先要注意输出的问题。比如输出的第三行，很多同学会按照下面来写

```
1 x = input() # 接收输入的字符
2 ...
3 # 第三行
4 print(x, x, x, x, x)
5 ...
```

他想用这样的方式来输出是非常好的，其他行可能也是按照这个方式输出的，但是这样就会出现错误，因为 print() 函数默认是用空格分开你用逗号隔开的变量的，比如

```
1 print(2, 3.14, "hello") # 会输出 2 3.14 hello
```

这样就会导致 Presentation Error，也就是说，你按照要求，每一行输出了对应数量的字符，但它最后长得样子（也就是他的排列方式）和标准输出是对不上的。为了改正，我们可以使用 print() 函数的参数：

```
1 print(2, 3.14, "hello", sep=" ") # print() 函数默认 sep=" "，也就是
    一个空格
```

这样输出就会变为 23.14hello。

此外，print() 函数还有 end 参数，大家可以自己输出一下试一试：

```
1 # 第一种
2 print(1, 3.1)
3 print("hello")
4 # 第二种
5 print(1, 3.1, sep=" ")
6 print("hello")
7 # 第三种
8 print(1, 3.1, end=" ")
9 print("hello")
```

掌握这两个之后，你再见到类似的输出就很容易解决了。

另外，如果大家学了字符串，本题还能用很多方法来解决，我们以输出的第二行为例：

```
1 x = input()
2 ...
3 # 输出的第二行
4 print(" "+x+x+x+" ") # 用+把字符串连接起来
5 print(f" {x}{x}{x} ") # 用格式化输出
6 ...
```

4 练习

好像没什么难的，应该不用讲。