

2026 年河北省职业院校技能大赛

“Python 编程”赛项样题

模块一 语法基础与算法分析

语法基础与算法分析任务包含任务详情、代码编辑区和控制台。每个算法设计任务都具有详细的任务说明、提交要求和测试用例。通过限定学生的输入和输出，对学生的输入和输出条件加以限定，学生能够更加准确高效地完成算法程序设计任务。

学生完成代码编写后，可以点击运行程序查看程序运行结果。学生也可以通过程序运行结果优化代码。

点击提交代码，可以讲程序运行结果上传，系统自动记录代码提交前的调试次数、程序执行时间等。

样题 1:

定一个整数数组 `nums` 和一个目标值 `target`，请你在该数组中找出“和”为目标值的两个整数，并返回他们的数组下标。你可以假设每种输入只会对应一个答案，数组中元素不能重复输入。请根据上面的要求，编写相应的算法，将右边的函数 `twoSum()` 补充完整。

举例

给定 `nums = [2, 7, 11, 15]`，`target = 9`;

`nums[0] + nums[1] = 2 + 7 = 9`，返回`[0, 1]`。

任务要求

1. 代码编写必须在右边指定的区域编写;
2. 函数接收两个变量，一个是整数数组 `nums: list`，另一个是目标值 `target: int`;
3. 如果存在两种或以上符合目标值 `target` 的情况，返回下标相加之和较小的数组;

4. 函数返回的值必须为一个数组，list 数据类型，元素排序为正序排序，如返回 [1, 3] 符合条件，[3, 1] 不符合条件。

当前任务：求两数之和

任务详情提交记录

元组。

举例

给定 nums = [2, 7, 11, 15], target = 9;
nums[0] + nums[1] = 2 + 7 = 9, 返回[0, 1].

● 任务要求

1. 代码编写必须在右边指定的区域编写;
2. 函数接收两个变量，一个是整数数组 nums: list, 另一个是目标值 target: int;
3. 如果存在两种或以上符合目标值 target 的情况，返回下标相加之和较小的数组;
4. 函数返回的值必须为一个数组，list数据类型，元素排序为正序排序，如返回 [1, 3] 符合条件，[3, 1] 不符合条件。

● 测试用例

用例一

输入: [2, 3, 2, 4], 7
输出: [1, 3]

用例二

输入: [2, 6, 7, 7], 9
输出: [0, 2]

twoSum

1 # 1. 运行或提交代码不需要自己编写测试用例，后台自动进行测试检查。
2 # 2. 您编写代码的区域需要限制在Solution类或其他类和函数体内，保证输入与输出符合任务要求即可。
3 # 3. 点击“提交代码”按钮，系统将查看程序是否正确，并保存代码，记录任务数据。
4 # 4. 提交代码次数越多，任务得分越低。
5 # 5. 点击右上方“结束任务”按钮，系统将在后台计算任务得分，任务结束。
6 # -*- coding: utf-8 -*-
7 class Solution:
8
9 def twoSum(self, nums: list, target: int) -> list:
10
11
12
13 pass

控制台重置提交代码

样题 2：

任务详情

给定一段英文句子，判断句中单词拼写是否满足以下规则。

除以下特殊情况外，句子中第一个单词首字母必须大写，其它所有单词小写：

1. 如果句中的某个单词或短语，字母全部为大写，则该单词或短语拼写正确。比如“USA”、“UK”、“JUST DO IT”等；
2. “Python”、“Java”、“MachineLearning”、“DataMining”四个单词必须为双引号中给出的形式，否则拼写不正确；
3. 如果句中单词为“数字+字母”的混合形式，比如“5G”，该单词所有字母全部大写。

任务要求

1. 输入英文句子为 str 类型，输出为 bool 类型；
2. 如果句子满足规则要求，程序返回结果为 True；否则返回 False。

测试用例

输入：'I love Python' 输出：True

输入: 'python love me' 输出: False

输入: 'JUST DO IT' 输出: True

输入: 'I come from HK' 输出: True

输入: 'Machinelearning is so hot' 输出: False

当前任务: 句子拼写检查

任务详情

提交记录

除以下特殊情况外, 句子中第一个单词首字母必须大写, 其它所有单词小写:

1. 如果句中的某个单词或短语, 字母全部为大写, 则该单词或短语拼写正确。比如 "USA"、"UK"、"JUST DO IT" 等;

2. "Python"、"Java"、"MachineLearning"、"DataMining" 四个单词必须为双引号中给出的形式, 否则拼写不正确;

3. 如果句中单词为 "数字+字母" 的混合形式, 比如 "5G", 该单词所有字母全部大写。

● 任务要求

1. 输入英文句子为 `str` 类型, 输出为 `bool` 类型;

2. 如果句子满足规则要求, 程序返回结果为 `True`; 否则返回 `False`。

● 测试用例

输入: 'I love Python' 输出: True

输入: 'python love me' 输出: False

输入: 'JUST DO IT' 输出: True

输入: 'I come from HK' 输出: True

输入: 'Machinelearning is so hot' 输出: False

detectCapital

```
1 # 1. 运行或提交代码不需要自己编写测试用例, 后台自动进行测试检查。
2 # 2. 影响写代码的区域需要限制在Solution类或其他类和函数体内, 保证输入与输出符合任务要求即可。
3 # 3. 点击“提交代码”按钮, 系统将查看程序是否正确, 并保存代码, 记录任务数据。
4 # 4. 提交代码次数越多, 任务得分越低。
5 # 5. 点击右上角“结束任务”按钮, 系统将在后台计算任务得分, 任务结束。
6 # -*- coding: utf-8 -*-
7 class Solution:
8     def detectCapital(self, st: str) -> bool:
9
10
11     pass
```

控制台

重置

提交代码

模块二 数据处理与分析

数据处理任务主要培养学生数据获取与清洗、商务数据分析、基础统计分析的能力, 能够熟练掌握对 Numpy、Pandas 等数据分析库的应用。实训系统内置了多个大数据文件, 可以通过程序直接读取并进行处理。

样题 1:

任务详情

给定一个 DataFrame 对象 `df`, 要求返回各行的平均值。

具体操作如下: 添加新列 `'avg'` 用于计算各行的平均值; 使用数组返回新列 `'avg'`。

任务要求

程序接收 DataFrame 对象 df，返回结果是 list 数据类型；
注意：平均值需要四舍五入保留两位小数。



样题 2:

任务详情

下方是某店铺在 2 月份的订单数据，共 28010 行、7 列。在该数据中，每一行代表一笔订单，每一笔订单可以 看做是一位顾客（买家），价格相同的产品可视为同一产品；

请编写程序，完成以下任务：

程序传入参数 area（收货地址），要求计算该地区订单的‘订单的付款转化率’、‘买家全额支付的转化率’、‘买家实际支付总金额’、‘客单价’、‘销量（订单量）最多的产品的价格’、‘买家实际支付总金额在所有地区实际支付总金额中的占比’，并将结果以列表形式返回。

任务要求

程序接收 str 类型的 area，返回的结果为 list 数据类型；

返回的结果中，列表中各个数据的数据类型为 str，其中转化率数据及占比数据需以百分比的形式输出，并且需要保留小数点后 2 位，如‘13.14%’，其余结果不需要以百分比形式输出，但需要保留小数点后 2 位；

如果某地区所有买家的买家实际支付金额为 0，即该地区没有付费行为。则

该地区的'客单价'、'销量(订单量)最多的产品的价格'需返回字符串'本地区无销量'，其他结果则正常计算；

如果该产品(订单)被全额退款，那么该产品不计入销量。

参考公式：

付款转化率 = 付款订单数 / 总订单数；

买家全额支付的转化率 = 买家全额支付的订单数 / 总订单数；

客单价 = 实际支付金额 / 支付买家数；

测试用例

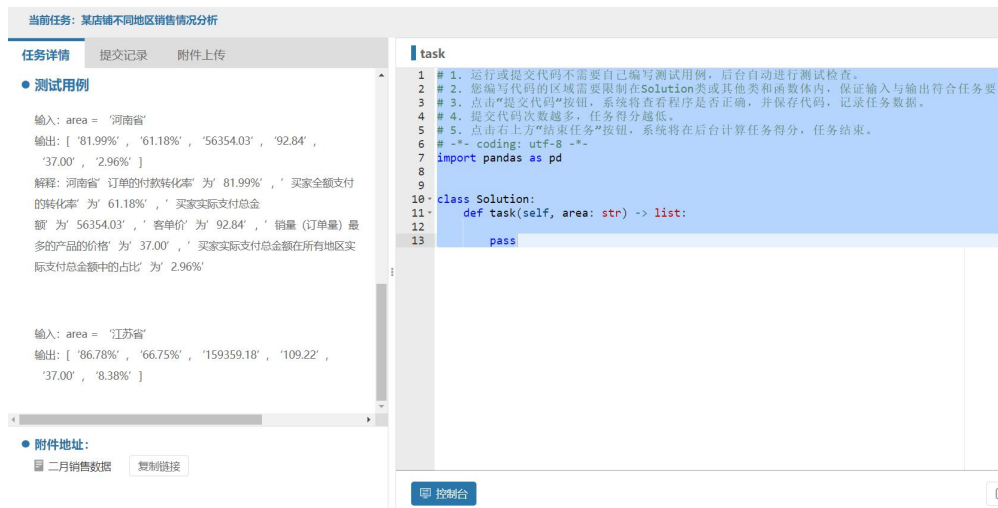
输入：area = '河南省'

输出：['81.99%' , '61.18%' , '56354.03' , '92.84' , '37.00' , '2.96%']

解释：河南省'订单的付款转化率'为'81.99%'，'买家全额支付的转化率'为'61.18%'，'买家实际支付总金额'为'56354.03'，'客单价'为'92.84'，'销量(订单量)最多的产品的价格'为'37.00'，'买家实际支付总金额在所有地区实际支付总金额中的占比'为'2.96%'

输入：area = '江苏省'

输出：['86.78%' , '66.75%' , '159359.18' , '109.22' , '37.00' , '8.38%']



模块三 计算机视觉

样题 1:

任务目标:

实现一个图像阈值处理对比系统，能够对输入的灰度图像进行多种阈值分割处理，并将原始图像与处理结果在同一窗口中可视化展示，以便直观比较不同阈值处理方法的效果差异。

功能要求:

1. 图像预处理:

- 以灰度模式读取指定路径的图像文件
- 对图像进行高斯模糊预处理以降低噪声影响

2. 阈值处理实现:

- 实现固定阈值二值化处理
- 实现自适应均值阈值处理
- 实现 Otsu's 阈值处理
- 实现高斯模糊后的 Otsu's 阈值处理

3. 结果可视化:

- 使用 3×2 子图布局展示所有图像
- 包含原始图像和 5 种阈值处理结果

- 设置合适的窗口大小和图像标题

技术要求:

代码规范:

- 变量命名必须使用有意义的英文名称
- 主要变量命名要求: 原始图像: `image` 固定阈值结果: `thresh1` 自适应阈值结果: `thresh` Otsu's 阈值结果: `th2` 高斯模糊图像: `blur` 高斯模糊后 Otsu's 结果: `th3`

参数设置:

- 固定阈值: 127
- 最大像素值: 255
- 高斯模糊核大小: (3, 3) 和 (5, 5)
- 自适应阈值参数: 邻域大小: 3 常数偏移: 5
- 窗口大小: `figsize=(30, 30)`
- 子图布局: 3 行 2 列

处理类型:

- `cv2.THRESH_BINARY`
- `cv2.ADAPTIVE_THRESH_MEAN_C`
- `cv2.THRESH_BINARY + cv2.THRESH_OTSU`

输出要求:

1. 正确显示包含 6 幅图像的对比窗口
2. 图像标题要求: "Original" - 原始图像 "Threshold Binary" - 固定阈值二值化 "Adaptive Mean Thresholding" - 自适应均值阈值 "Otsu's Thresholding" - Otsu's 阈值处理 "Guassain Otsu's Thresholding" - 高斯模糊后 Otsu's 处理
3. 图像显示清晰, 便于观察阈值分割效果差异
4. 程序运行无错误, 能够完整展示所有处理结果

样题 2:

任务目标:

实现一个完整的图像形态学处理程序，对指定图像进行腐蚀、膨胀、开运算和闭运算操作，并将处理结果可视化展示。

功能要求:

1. 图像读取与预处理:
 - 读取指定路径下的图像文件
 - 将图像颜色格式从 BGR 转换为 RGB 格式
2. 形态学处理操作:
 - 定义形态学操作所需的卷积核
 - 分别实现腐蚀、膨胀、开运算、闭运算四种形态学处理
3. 结果可视化:
 - 在同一窗口中展示原始图像及所有处理结果
 - 采用合理的子图布局（建议 3×2 网格）
 - 为每个子图设置清晰的标题标识

代码规范要求:

变量命名规范:

- 原始图像变量: ``image``
- 卷积核变量: ``kernel``
- 处理结果变量: ``erosion``, ``dilation``, ``opening``, ``closing``

卷积核参数:

- 形状: 5×5 正方形
- 数据类型: `uint8`
- 元素值: 全 1 矩阵

函数调用参数:

- 迭代次数: 1 次
- 形态学操作类型常量使用 OpenCV 标准定义

输出要求:

程序运行后应显示一个可视化窗口，包含以下 6 个子图:

1. 原始图像 (标题: Original)
2. 腐蚀结果 (标题: Erosion)
3. 膨胀结果 (标题: Dilation)
4. 开运算结果 (标题: Opening)
5. 闭运算结果 (标题: Closing)
6. 空白子图 (预留)