

Общие положения об олимпиаде

Где размещать решения

На компьютере каждого участника создана папка **ispring-challenge-2026**.

Внутри папки размещаются:

- p01_название_задания...
- p02_название_задания...
- p03_название_задания...

Код решения размещать в папке вида “p01_название_задания”.

Каждое задание имеет тесты. Доступный вам набор тестов неполный и проверяет только некоторые ситуации. Вы можете написать дополнительные тесты самостоятельно.

Вы можете использовать тесты чтобы проверить себя.

В некоторых заданиях есть **заготовки кода и уже реализованный код** для упрощения решения.

Правила

Во время олимпиады запрещено пользоваться искусственным интеллектом, интернетом, телефоном и другими средствами коммуникации без разрешения наблюдателя. Если замечено использование телефона первый раз — будет сделано предупреждение.

За повторное использование — **дисквалификация и отстранение от соревнований**.

Завершение выполнения заданий

Перед тем, как покинуть рабочее место, сообщите наблюдателю, чтобы он скопировал ваше решение. Не выключайте компьютер до тех пор, пока не убедитесь, что ваше задание скопировано.

RocketSlice Pizza

Привет! Ты только что попал в команду разработки супер-популярной службы доставки пиццы **RocketSlice**. У нас сегодня «Черная пятница», и сервера не справляются с наплывом заказов.

Наш главный инженер ушел на обед и забыл включить систему автоматической обработки. Сейчас все заказы падают в общую кучу в виде "грязного" файла: курьеры вносят данные как попало, адреса невалидные, названия пицц написаны с ошибками (в разном регистре). Кухня стоит, логистика сломана, а клиенты злятся.

Ваша миссия: разработать три скрипта. Каждый скрипт должен быть протестирован в симуляторе. Ошибка в логике может привести к потере компанией денег и репутации!

Задание 1. Умный фильтр заказов

От кого: Анна, менеджер смены

«Я смотрю в список заказов и ничего не понимаю! Кто-то написал "Маргарита", кто-то "мАргАрИТа". Кухня не может начать готовить, пока мы не поймем, какие заказы реальные, а какие — мусор или отменены».

Ваша задача: написать программу, которая будет принимать через консоль список заказов с ошибками и возвращать очищенные данные.

Что необходимо реализовать:

- **Привести порядок в тексте:** названия пицц должны начинаться с заглавной буквы, а все остальные буквы должны быть строчными (например, мАргАрИТа превращается в Маргарита). Имена клиентов необходимо очистить от лишних пробелов в начале и в конце строки.
- **Избавиться от отменённых заказов:** удалить из итогового списка все заказы, которые уже были отменены (статус равен *canceled*).
- **Проверить количество:** если кто-то по ошибке заказал 0 или отрицательное число пицц, такой заказ нужно целиком удалить из итогового списка.

Формат входных данных

На вход через консоль подаются строки с заказами. Значения разделены запятыми. Первым идет **заголовок** (строка).

order_id,client_name,status,street,house,flat,item_name,count

Поля ввода по порядку:

1. **order_id** (целое число): уникальный номер заказа. **Важно:** на один заказ (с одинаковым order_id) может приходиться несколько разных пицц в разных строках.
2. **client_name** (строка): имя клиента. Требуется очистки от лишних пробелов.
3. **status** (строка): текущий статус заказа. Бывает **active** (активен) или **canceled** (отменен).
4. **street** (строка): название улицы для доставки.
5. **house** (строка): номер дома (могут быть буквы, например "10А").
6. **flat** (целое число): номер квартиры.
7. **item_name** (строка): название заказанной пиццы. Требуется исправления регистра.

8. **count** (целое число): количество пицц данного вида. Количество не может быть дробным, нулем или отрицательным числом.

Формат выходных данных

Программа должна вывести в консоль строки, разделенные запятыми, где каждая строка соответствует одной позиции пиццы из ввода. Сначала выводится **заголовок** (строка):

order_id,client_name,status,street,house,flat,item_name,count

Затем выводятся данные по заказам.

Примеры входных и выходных данных

Пример input:

order_id,client_name,status,street,house,flat,item_name,count

6,Иван ,active,Веселая,10,5,пепперони,2

8,Дима ,canceled,Гуманитарная,11,2,мясная,1

9,Саша ,active,Гуманитарная,13,5,мясная,0

Пример output:

order_id,client_name,status,street,house,flat,item_name,count

6,Иван,active,Веселая,10,5,Пепперони,2

Задание 2. Карта спроса

От кого: Борис, главный диспетчер

«Заказы почистили, спасибо! Но я не вижу общей картины. Мне нужно привязать улицы к районам и посчитать выручку с каждого заказа, чтобы понять, где у нас сегодня самая горячая зона. Считать вручную суммы по прайс-листу я буду до утра!»

Ваша задача: написать программу, которая анализирует список корректных заказов, сопоставляет улицы с районами, рассчитывает стоимость и формирует сводную таблицу.

Что необходимо реализовать:

- **Определить район:** используя встроенный в программу справочник (см. ниже), сопоставить улицу из заказа с соответствующим районом.
- **Вычислить стоимость позиции:** для каждой строки заказа вычислить итоговую стоимость (цена конкретной пиццы по справочнику умноженная на заказанное количество).
- **Собрать статистику по району:** подсчитать общее количество уникальных заказов (по `order_id`) в каждом районе и общую сумму выручки со всех заказов в этом районе.
- **Сформировать отчет:** вывести каждую позицию заказа, добавив к ней информацию о районе и его общей статистике.

Справочные данные (для копирования в код)

Чтобы избежать опечаток, используйте данные, предоставленные ниже, в своей программе.

Справочник районов (Регион : [Список улиц]):

- "Северный": ["Гуманитарная", "Серверная", "Облачная", "Базовая", "Скриптовая"]
- "Южный": ["Бомбезная", "Стильная", "Кнопочная", "Адаптивная", "Реактивная"]
- "Западный": ["Светлобудущая", "Контейнерная", "Потоковая", "Докерная", "Сборочная"]
- "Восточный": ["Техническая", "Бажная", "Тестовая", "Прогоночная", "Чеклистная"]
- "Центральный": ["Веселая", "Пиксельная", "Игровая", "Спрайтовая", "Движковая"]

Справочник цен (Пицца : Цена в рублях):

- "Пепперони": 300,
- "Ананасовая": 319,
- "Детская": 200,
- "Мясная": 420,
- "Вегетарианская": 280,
- "Сырная": 400,
- "Барбекю": 420,
- "Маргарита": 350

Формат входных данных

На вход через консоль подаются строки с очищенными заказами. Значения разделены запятыми. Первым идет **заголовок** (*строка*).

order_id,client_name,status,street,house,flat,item_name,count

Поля ввода по порядку:

1. **order_id** (*целое число*): уникальный номер заказа.
2. **client_name** (*строка*): имя клиента.
3. **status** (*строка*): статус заказа (в этом задании всегда active).
4. **street** (*строка*): название улицы.
5. **house** (*строка*): номер дома (может содержать буквы, например, "10А").
6. **flat** (*целое число*): номер квартиры.
7. **item_name** (*строка*): название пиццы (с заглавной буквы).
8. **count** (*целое число*): количество пицц данного вида в этой строке.

Формат выходных данных

Программа должна вывести в консоль строки, разделенные запятыми, где каждая строка соответствует одной позиции пиццы из ввода. Сначала выводится **заголовок** (*строка*):

region,orders_count,total_revenue,order_id,street,house,flat,price

Затем выводятся данные по заказу, предоставленные ниже:

1. **region** (*строка*): название района, к которому относится улица.
2. **orders_count** (*целое число*): общее количество **уникальных** заказов (order_id) в этом районе.
3. **total_revenue** (*целое число*): общая сумма денег, заработанная в этом районе (за все пиццы во всех заказах района).
4. **order_id** (*целое число*): номер текущего заказа.
5. **street** (*строка*): название улицы.
6. **house** (*строка*): номер дома.
7. **flat** (*целое число*): номер квартиры.

8. **price** (целое число): стоимость **текущей позиции** (цена пиццы умножить на количество).

Примеры входных и выходных данных

Пример input:

order_id,client_name,status,street,house,flat,item_name,count

6,Иван,active,Веселая,10,5,Пепперони,2

3,Данил,active,Техническая,10,5,Детская,2

1,Сима,active,Бажная,7,1,Сырная,2

1,Сима,active,Бажная,7,1,Барбекю,2

Пример output:

region,orders_count,total_revenue,order_id,street,home,flat,price

Центральный,1,600,6,Веселая,10,5,600

Восточный,2,2040,3,Техническая,10,5,400

Восточный,2,2040,1,Бажная,7,1,800

Восточный,2,2040,1,Бажная,7,1,840

Пояснение к примеру:

- **Восточный район:** было всего 3 заказа (№2, №3 и №4) на улицах Техническая и Бажная. Он состоит из двух позиций: 2 Сырные ($400 * 2 = 800$), 2 Барбекю ($420 * 2 = 840$) и 2 Детские ($200 * 2 = 400$). Общая выручка района: $800 + 840 + 400 = 2040$ рублей.
- **Центральный район:** был всего 1 заказ (№1) на улице Веселая. Была заказана 2 пицца Пепперони ($300 * 2 = 600$). И того, общая сумма района составляем 600 рублей.

Задание 3. Генератор маршрутов

От кого: Дмитрий, старший курьер

«Аналитика — это отлично, но ребятам на линии нужна конкретика. Сейчас курьеры выходят на смену и не понимают, в какую точку города им выгоднее ехать. Нам требуется алгоритм маршрутизации: нужно направлять людей в самый прибыльный район, чтобы не гонять курьеров впустую».

Ваша задача: проанализировать сводную таблицу по районам из предыдущего задания, найти самый выгодный район (с наибольшей общей выручкой) и рассчитать, сколько курьеров туда нужно отправить.

Условие задачи:

- **Вместимость сумки:** Один курьер может взять **максимум 3 заказа за одну поездку**. (Например: если в районе 4 уникальных заказа, туда потребуется отправить 2 курьеров).

Описание входных полей

Каждая строка ввода содержит информацию о конкретной позиции в заказе и общую статистику района. Значения разделены запятыми.

1. **region** (*строка*): название района (например, Восточный).
2. **orders_count** (*целое число*): общее количество уникальных заказов в этом районе.
3. **total_revenue** (*целое число*): общая выручка (сумма денег) со всех заказов в этом районе.
4. **order_id** (*целое число*): номер заказа.
5. **street** (*строка*): название улицы.
6. **house** (*строка*): номер дома.
7. **flat** (*целое число*): номер квартиры.
8. **price** (*целое число*): стоимость конкретной позиции пиццы.

Формат входных данных

Программа считывает из консоли строки текста. Значения разделены запятыми. Первая строка — всегда заголовок с названиями столбцов:

region,orders_count,total_revenue,order_id,street,house,flat,price

Формат выходных данных

Программа должна вывести в консоль ровно две строки (первая — заголовок, вторая — результат). Значения разделяются запятыми.

Заголовок: region,total_revenue,count_curiers

Поля вывода:

1. **region** (*строка*): название самого прибыльного района.
2. **total_revenue** (*целое число*): общая выручка этого района.
3. **couriers_needed** (*целое число*): сколько курьеров нужно отправить в этот район.

Примеры входных и выходных данных

Пример input:

region,orders_count,total_revenue,order_id,street,home,flat,price

Восточный,2,6600,1,Инпутова,7,1,3400

Восточный,2,6600,1,Инпутова,7,1,3200

Западный,2,5500,3,Аутпутова,10,5,2100

Западный,2,5500,6,Веселая,10,5,3400

Пример output:

region,total_revenue,count_curiers

Восточный,6600,1