

Общие положения об Олимпиаде

Где размещать решения

На компьютере каждого участника создана папка **ispring-challenge-2025**.

Внутри папки размещаются:

- p01_название_задания...
- p02_название_задания...
- p03_название_задания...
- Правила оформления исходного кода

Код решения размещать в папке вида "p01_название_задания".

Каждое задание имеет хорошие и плохие тесты. При хороших тестах программа получает на вход валидные данные, при плохих - невалидные и должна вывести ошибку. 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** для упрощения решения.

В каждой папке находится:

- Папка **good** - хорошие тесты
- Папка **bad** - плохие тесты
- **run_tests.bat** - файл для запуска тестов
- **README.txt** - инструкция по запуску тестов

Правила

Во время олимпиады запрещено пользоваться 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.

Задание 1. Мониторинг аварийных ситуаций в системе давления

Вы работаете инженером на современном производстве, где используется система сжатого воздуха. Эта система отвечает за работу станков и автоматизированных линий.

Стабильное давление воздуха критически важно для бесперебойной работы оборудования. Резкие скачки, падение или рост давления могут привести к:

- Поломке дорогостоящих станков
- Браку продукции
- Остановке конвейера
- К другим аварийным ситуациям

Ваша задача — разработать программу для мониторинга давления и выявления опасных изменений.

Техническое задание

Программа должна анализировать показания датчика давления и обнаруживать следующие аварийные ситуации:

- Опасный рост давления:
 - Если три последовательных значения монотонно возрастают (например, 200 → 250 → 300).
- Опасное падение давления
 - Если три последовательных значения монотонно убывают (например, 500 → 400 → 300).
- Резкий скачок давления
 - Если разница между двумя соседними значениями превышает 50 кПа (например, 100 → 180 или 300 → 200).
- Критическое давление
 - Если давление превышает 1000 кПа (риск разрыва трубопровода).
 - Если давление падает ниже 50 кПа (система не может работать).

Проверки должны проходить в следующем порядке. Сначала проверяем критическое давление, затем резкие скачки в давлении и в конце проверяем на опасный рост или падение.

Формат ввода

- Программа принимает последовательность чисел: показания датчика.
- Ввод завершается числом -1 (сигнал остановки).
- Гарантируется, что сигнал остановки приходит в конце. Дополнительных проверок делать не нужно.
- Минимальное количество значений для анализа помимо сигнала остановки — 1.

Формат вывода

- При обнаружении аварии программа выводит тип проблемы и значения, которые её вызвали. Примеры в таблице ниже.
- В конце подводится итог: общее количество аварий. Если не было ни одной аварии, то программа выводит сообщение: “No alarm were detected”
- Если на вход было подано меньше одного значения, кроме -1, то программа выводит сообщение: “Error”

Примеры входных и выходных данных:

Ввод	Вывод
-1	Error
100 -1	No alarm were detected
10 -1	ACCIDENT: pressure below 50 kPa! Total emergency situations detected: 1
100 200 -1	ACCIDENT: sudden pressure surge (+100 kPa per period) Total emergency situations detected: 1
100 140 180 1001 980 940 49 -1	ACCIDENT: dangerous increase in pressure (100 -> 140 -> 180) ACCIDENT: pressure exceeded 1000 kPa! ACCIDENT: sudden pressure surge (+821 kPa per period) ACCIDENT: dangerous increase in pressure (140 -> 180 -> 1001) ACCIDENT: dangerous pressure drop (1001 -> 980 -> 940) ACCIDENT: pressure below 50 kPa! ACCIDENT: sudden pressure drop (-891 kPa per period) ACCIDENT: dangerous pressure drop (980 -> 940 -> 49) Total emergency situations detected: 8

Задание 2. Максимальное число

Сергей заметил, что иногда можно увеличить значение числа, поменяв местами всего пару цифр. Он решил проверить это утверждение и хотел написать программу, но у него не получилось. Помогите Сергею написать эту программу, которая составляет наибольшее число из цифр числа.

Формат входных и выходных данных:

Напишите программу, которая:

1. Принимает на вход целое положительное число, не превышающее 9999.
2. Находит и выводит максимально возможное число, которое можно получить, поменяв местами две цифры этого числа не более одного раза.
3. Если входные данные некорректны (например, пустая строка, нечисловое значение, отрицательное число или число больше 9999), программа должна корректно завершиться, выведя сообщение **ERROR**.
4. Если исходное число уже является максимальным по своему значению, программа должна вывести его без изменений.

Учтите, что число может состоять из одной цифры – в этом случае менять нечего, и его следует вывести как есть.

Примеры входных и выходных данных:

Ввод	Вывод	Объяснение
2736	7236	При обмене цифр 2 и 7 получается число 7236 – наибольшее возможное.
9973	9973	Число уже максимально, обмен менять не требуется.
123	321	Обмен первой и последней цифры даёт максимальное число 321.
12а	ERROR	Введена строка, содержащая недопустимые символы.
-45	ERROR	Программа не работает с отрицательными числами.

Задание 3. Римский математик

Вы работаете в компании, разрабатывающей цифровые решения для музеев, архивов и библиотек, специализирующихся на исторических документах. В таких учреждениях по-прежнему используется римская система счисления при описании дат, инвентарных номеров и других числовых значений, чтобы сохранить историческую аутентичность документов.

Необходимо разработать алгоритм, способный выполнять базовые арифметические операции с римскими числами, так как сотрудники часто сталкиваются с задачами вычислений в этой системе. Алгоритм должен уметь:

- Складывать два римских числа;
- Вычитать одно римское число из другого;
- Умножать два римских числа.

О римских числах

Римские числа — это способ записи чисел с помощью букв латинского алфавита. Для перевода в десятичную систему нужно учитывать порядок символов и их значения.

Обозначения

- I = 1
- V = 5
- X = 10
- L = 50
- C = 100
- D = 500
- M = 1000

Правила перевода:

- Если символ меньше или равен следующему — прибавляем его значение.
 - Пример: VI = 5 + 1 = 6
- Если символ больше следующего — вычитаем его значение.
 - Пример: IV = 5 - 1 = 4

Примеры:

- XII = 10 + 1 + 1 = 12
- IX = 10 - 1 = 9
- MCMXC = 1000 + (1000 - 100) + (100 - 10) = 1990

Формат входных данных:

На вход программа принимает два римских числа (от I до MMMCMXCIX, т.е. от 1 до 3999) и оператор (*, '-', '+'). Все данные вводятся в одной строке слитно.

Гарантируется, что если римское число введено, то оно введено правильно.

Формат выходных данных:

- Программа выводит результат операции в римской системе счисления.
- Если вход содержит недопустимые символы, программа должна вывести **'ERROR'**.
- Если результат вычисления выходит за пределы допустимых римских чисел (меньше I или больше MMMCMXCIX), также вывести **'ERROR'**.

Вам дана заготовка программы RomanCounter, которую вы можете использовать для написания своей программы.

Примеры входных и выходных данных:

Ввод	Вывод	Объяснение
III+XLIX	LII	$3 \text{ (III)} + 49 \text{ (XLIX)} = 52 \text{ (LII)}$
XX-XXX	ERROR	$20 \text{ (XX)} - 30 \text{ (XXX)} = -10$, операция некорректна так как римскими числами нельзя записать отрицательные числа и ноль
AB+X	ERROR	Введены некорректные символы