

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. Дробные трансформации

Дана правильная дробь a/b в несократимом виде (т.е. a и b взаимно просты). С ней проводят следующее преобразование:

- К числителю и знаменателю прибавляют 1 (например, дробь $1/2$ должна превратиться в $2/3$).
- Дробь сокращают, если это возможно (например, дробь $2/4$ должна превратиться в $1/2$).

Требуется определить минимальное число таких операций, чтобы получить другую правильную несократимую дробь c/d (c и d взаимно просты), или вывести 0.

Если знаменатель одной из дроби (числа b или d) равен 0, следует вывести строку "ERROR" и завершить выполнение программы.

Если изначальные дроби равны, требуется вывести число -1 и завершить выполнение программы.

Формат ввода:

На вход подаются числа a , b , c и d , при этом каждое новое число расположено на новой строке.

Гарантируется, что введенные числа a , b , c и d будут в диапазоне от 0 до 1000 (включительно), при этом числа a и b , b и d попарно взаимно просты.

Формат вывода:

Выведите строку "ERROR", если знаменатель хотя бы одной из дроби равен нулю.

Выведите число -1, если изначальные дроби равны.

Выведите минимальное количество операций, необходимых для получения c/d из a/b . Если это невозможно, выведите 0.

Примеры входных и выходных данных:

Ввод	Вывод	Объяснение
1 3 2 3	2	Исходная дробь: $1/3$, целевая дробь – $2/3$. После первой операции: $2/4$, сокращается до $1/2$. После второй операции: $2/3$. Цель достигнута! Соответственно следует вывести ответ – 2.
1 3 2 5	0	Из дроби $1/3$ через преобразования не получить дробь $2/5$. Попробуйте догадаться, из-за чего это происходит.
1 0 4 5	ERROR	Знаменатель первой дроби равен нулю.
6 7 6 7	-1	Дроби изначально равны.

Задание 2. Анализ ДНК

Вы работаете в биотехнологической компании, которая разрабатывает алгоритмы для анализа цепочек ДНК. В процессе исследований биоинженеры представляют последовательности фрагментов ДНК в виде квадратных матриц, где каждая клетка содержит символ нуклеотида (A, T, C, G).

Было замечено, что некоторые виды мутаций проявляются как палиндромные последовательности нуклеотидов вдоль строки, столбца или диагонали. Палиндром — это биологический маркер симметричной мутации, и его наличие может указывать на потенциально нестабильные участки ДНК. Необходимо найти все палиндромы.

Формат входных данных:

На вход программа принимает размер матрицы N ($2 \leq N \leq 10$), потом на N строках по N символов нуклеотидов (**A, T, C, G**). Необходимо найти все палиндромы (слова, читающиеся одинаково слева направо и справа налево), которые можно получить:

- В строках
- В столбцах
- На главных диагоналях (из левого верхнего в правый нижний угол и наоборот)

Гарантируется, что матрица вводится нужного размера.

Формат выходных данных:

- После нахождения всех уникальных палиндромов, программа выводит их в алфавитном порядке.
- Если палиндромов нет, вывести **'NONE'**.
- Если в матрице попадет символ, не являющийся нуклеотидом (**A, T, C, G**), вывести **'ERROR'** и завершить программу.
- Если N не входит в допустимый диапазон, вывести **'ERROR'** и завершить программу.

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

Примеры входных и выходных данных:

Ввод	Вывод	Объяснение
3 ATA CGC TTT	ATA CGC TTT	В этой матрице присутствуют только палиндромы в строках
3 ACA CAC ACA	AAA ACA CAC	В этой матрице присутствуют палиндромы в строках (ACA, CAC), в столбцах (ACA, CAC) и на главных диагоналях (AAA), после проверки на уникальность и сортировке остаются AAA, ACA, CAC
4 ATCG AAAC GTAA TTTC	NONE	В этой матрице отсутствуют палиндромы.
4 ABCD EFGH IJKL MNOP	ERROR	В матрице присутствуют символы, не являющиеся нуклеотидами (A, T, C, G)
52 ...	ERROR	Размер матрицы не может превышать 10.

Задание 3. Система обнаружения

На складе установлены датчики, ориентированные в одном из четырёх направлений (вверх, вниз, влево, вправо). Эти датчики испускают сигналы в выбранном направлении для сканирования коридоров. Сигнал продвигается по прямой, пока не встретит препятствие в виде стены или не выйдет за границы склада. Если на пути сигнала находится другой датчик, система фиксирует его и помечает как "обнаруженный". Задача — смоделировать прохождение сигнала от каждого датчика и определить, какие объекты будут зафиксированы системой.

Формат входных данных:

На вход программе подаются два числа N и M — длина и ширина склада. N и M находятся в диапазоне от 2 до 10. Затем вводится план склада размером $N \times M$, состоящее из символов:

- Стены:
 - 'I' (вертикальная стена)
 - '-' (горизонтальная стена)
- Пустое пространство:
 - '.'
- Датчики:
 - '>' (Датчик, ориентированный вправо)
 - '<' (Датчик, ориентированный влево)
 - '^' Датчик, ориентированный вверх)
 - 'v' (Датчик, ориентированный вниз)

Гарантируется, что план склада вводится нужного размера.

Условия задачи:

Необходимо промоделировать испускание сигналов датчиками:

1. Если сигнал попадает в другой датчик, этот датчик помечается как обнаруженный (латинский символ 'O')
2. Если сигнал попадает в стену ('I' или '-') или выходит за границы склада, ничего не происходит
3. Сигналы в конечную точку прилетают мгновенно

Формат выходных данных:

- Если код выполнен успешно, то
 - На первой строчке пишется количество обнаруженных датчиков, если их нет, то написать **'NONE'**.
 - Показать склад после моделирования сигналов (с отображением обнаруженных датчиков).
- Если встречен неопознанный символ, вывести **'ERROR'** и завершить программу.
- Если N или M не входит в допустимый диапазон, вывести **'ERROR'** и завершить программу.

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

Примеры входных и выходных данных:

Ввод	Вывод	Объяснение
5 7>>...< .-...-. .^...<.	3 2 3 2 7 4 2>0...0 .-...-. .0...<.	Датчики с позиций (2;3) и (2;7) попали друг в друга. Датчик с позиции (2;2) попал в (2;3). Датчик с позиции (4;2) попал в стену. Датчик с позиции (6;6) попал в (4;2)
2 2 <> ..	NONE <> ..	Ни один из датчиков не попал в другие датчики
2 2 >! ..	ERROR	Неопознанный символ
52 ...	ERROR	Размер склада не должен превышать 10