

Общие положения об Олимпиаде

- Каждое задание находится в отдельной папке с именем вида **pXX_название**.
- К каждому заданию прилагается файл **README.txt** в котором можно найти краткое описание задачи и **требования к оформлению** решения.
- К некоторым задачам также прилагается **заготовка**, которую вы можете использовать для написания своей программы.

Проверяются только те файлы, которые находятся в соответствующей папке задания.

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

Тестирование программы “Умное окно”

Важно! В этой задаче не нужно писать программу.

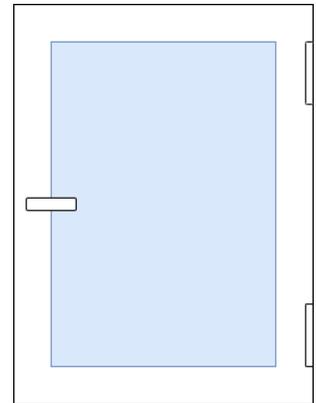
Составьте набор входных и выходных данных, позволяющих протестировать работу программы, которая моделирует работу умного окна. Программу писать в этом задании не нужно.

Вам дана эталонная (правильная) программа **window.exe**, которая принимает команды управления пластиковым окном и выполняет их, если команда корректная.

Представляем вам новую разработку “умное пластиковое окно”).

Программе на вход подаются действия, которые можно выполнять над пластиковым окном. Поддерживаются следующие команды:

- Распахнуть (**open**). Распахнуть можно только закрытое окно, у которого рукоятка направлена вбок.
- Откинуть для проветривания (**aerate**). Откидывать разрешается только закрытое окно, у которого рукоятка направлена вверх.
- Закрыть (**close**). Открытое окно можно закрыть, только когда рукоятка направлена вбок. Откинутае окно можно закрыть только когда рукоятка направлена вверх.
- Повернуть вверх (**up**).
 - Только в этом состоянии окно можно откинуть для проветривания
- Повернуть вниз (**down**). Закрывает замок окна.
 - В этом состоянии окно нельзя ни открывать, ни проветривать.
- Повернуть рукоятку вправо (**right**). Открывает замок окна.
 - В таком состоянии окно можно открывать и закрывать.



Поворачивать рукоятку можно, только когда окно закрыто.

Изначально окно закрыто, а его рукоятка направлена вниз.

Программа считывает с клавиатуры команды, по одной в каждой строке, и выполняет их.

- **open**. Распахивает окно и выводит строку “opened” (открыто).
- **aerate**. Откидывает окно для проветривания и выводит строку “aerating” (проветривается).
- **close**. Закрывает окно и выводит строку “closed” (закрыто).
- **up**. Поднимает рукоятку вверх выводит строку “handle is up” (рукоятка направлена вверх).
 - **down**. Поворачивает рукоятку вниз и выводит строку “locked” (заблокировано).

- **right**. Программа должна повернуть рукоятку вправо и выводит строку “unlocked” (разблокировано).
- **exit**. Заканчивает работу программы. Ввод данных можно также завершить, нажав комбинацию клавиш Ctrl+Z, а затем нажать Enter.

Если команда не может быть выполнена, например, при попытке открыть окно, когда рукоятка направлена вниз, программа должна оставить окно в прежнем состоянии и вывести строку “**error**” (ошибка), и продолжить ввод следующих команд.

Строка “**error**” также должна быть выведена, если пользователь ввёл команду, отличную от вышеперечисленных.

Если пользователь пытается повторно выполнить разрешённое действие (например, попытка открыть уже открытое окно), должно выводиться соответствующее сообщение об успехе.

Примеры входных и выходных данных

Ввод	Вывод	Пояснение
right open open	unlocked opened opened	Разблокировали окно (рукоятка вправо) Открыли его Открыли ещё раз
up aerate exit	handle is up aerating	Перевели рукоятку вверх Начали проветривать Завершить работу программы.
open	error	Изначально окно заблокировано (рукоятка вниз), поэтому открыть его нельзя

Указания

Изучите как работает данная вам **правильная программа** window.exe. Запустите её и посмотрите какие данные она выведет в ответ на вводимые данные.

Предоставьте папку с набором файлов с именами **<N>-input.txt** и **<N>-output.txt**, где **<N>** – некоторое целое число. Файлы **<N>-input.txt** содержат входные данные, а **<N>-output.txt** — ожидаемые выходные данные.

Пример: в файле 3-output.txt запишите данные, которые правильная программа должна вывести, если ей подать на вход данные из файла 3-input.txt.

Созданные вами файлы будут использованы для проверки работы правильных и неправильных версий программы window.exe.

Правильная программа – та, которая выдаёт результат в соответствии с требованиями.

Неправильная программа – та, которая выдаёт результат, отличный от требуемого.

Набор подготовленных вами файлов должен быть таким, чтобы:

- **все правильные программы** для каждого `<N>-input.txt` вывели результат, совпадающий с `<N>-output.txt`;
- **каждая из неправильных программ** для данных из хотя бы одного из входных `<N>-input.txt` вывела результат, не совпадающий с `<N>-output.txt`.

Счастливым билет

Вступление

Шестиклассник Жора узнал, что билет называют счастливым, если его номер читается в обе стороны одинаково. Его заинтересовал этот факт и он решил написать программу, которая сможет определить, является ли номер на билета счастливым. Так как программирование он начал изучать недавно, его знаний и навыков пока не хватает для решения этой задачи. Помогите Жоре написать программу, чтобы он смог понять как она работает.

Палиндром — число, буквосочетание, слово или текст, одинаково читающееся в обоих направлениях.

Требования к программе

На вход программа принимает целое положительное число не превышающее 1 000 000. Если число является палиндромом, программа выводит сообщение YES, в противном случае NO. Если на вход была подана пустая строка, не числовое значение или отрицательное число, программа должна корректно завершиться, отобразив сообщение ERROR.

Примеры входных и выходных данных:

Ввод	Вывод	Пояснение
121	YES	Введённое число является палиндромом
10	NO	Введённое число не является палиндромом
7A	ERROR	На вход была подана строка, содержащая нецифровые символы
-10	ERROR	Программа не работает с отрицательными числами

32 февраля

Напишите программу, которая будет переводить формат даты:

<номер дня в году> <год> в
<день месяца> <номер месяца>.

Формат входных данных:

На вход программе подаются два числа. Первое число - порядковый номер дня от 1 до 365 (366 в високосном), второе — год. Год может быть от 1900 до 9999.

В случае ввода некорректных данных пользователем программа выводит сообщение "ERROR".

Примеры входных и выходных данных:

Ввод	Вывод	Пояснение
2 2024	2 1	Второй день 2024 года — второе января. Программа вывела день (2) и месяц (1).
366 2024	31 12	2024 — високосный год, в нем не 365, а 366 дней. 366-й день 2024 года — 31 декабря. Программа вывела день месяца и месяц.
366 2023	ERROR	2023 год — невисокосный. В нем 365 дней. Поэтому программа вывела ERROR, так как 366 выходит за допустимые пределы

Гарантируется, что программе на вход подаются только целые числа. Год не выходит за пределы 1900...9999, а день - от -1000 до +1000.

Заготовка функции для чтения входных данных:

В файле DateConverter.pas вам предоставлена заготовка программы. Она умеет считывать данные, но не обрабатывает все ошибки. Вам также даны файлы с тестовыми данными и скрипт для облегчения проверки вашей программы.

Прочитайте файл README.txt в папке с заданием.

В високосном году в феврале 29 дней, а в невисокосном — 28. Год считается високосным, если делится на 4, но не делится на 100. При этом год считается високосным, если делится на 400. Поэтому 1900 год — не високосный, а 2000 — високосный.