Задание 1. Тестовые данные. 40 баллов

Важно! В этой задаче не нужно писать программу.
Цель задания: составить набор входных и выходных данных, позволяющих протестировать работу программ для работы с анаграммами. Программу писать в этом задании не нужно.

Вам дана эталонная программа **anagrams.exe**, которая проверяет, будут ли анаграммами две английские фразы, которые ввёл пользователь.
Анаграмма – литературный приём, при котором перестановка букв слова или фразы в результате даёт новое слово или фразу.
Примеры анаграмм:

- **lamp** и **palm**
- **New York Times** и **Monkeys write**
- **Eleven plus two** и **twelve plus one**

Программа должна игнорировать символы, отличные от заглавных и строчных букв английского алфавита. Например, строки “**2 cats**” и “**3 acts**” – анаграммы, так как в проверке участвуют только буквы.
Регистр букв, которыми записаны введённые строки, значения не имеет. Например, строки “Cats” и “Scat” – анаграммы, даже если регистр букв C и S в них различается.
В анаграммах буквы не удаляются и не добавляются, а только переставляются. Например, строки “history” и “his story” – не анаграммы из-за разного количества букв s.
Строка считается анаграммой самой себя: строки “hello” и “hello” – анаграммы.

Формат входных данных программы

Программа считывает в окне консоли две фразы – каждую в отдельной строке.

Формат выходных данных

После считывания данных правильная программа должна вывести в консоль одну из строк:

- **ANAGRAMS**, если введённые фразы – анаграммы
- **NOT ANAGRAMS**, если фразы - не анаграммы
- **ERROR**, если пользователь досрочно прекратил ввод данных, нажав клавиши Ctrl+Z, а затем Enter
- **ERROR**, если любая из введенных фраз не содержит символов английского алфавита

Примеры

Ввод	**Вывод**
apple banana	NOT ANAGRAMS
lamp palm	ANAGRAMS
hello	ERROR

Указания

Изучите как работает данная вам **правильная программа** anagrams,exe. Запустите её и посмотрите какие данные она выведет в ответ.
Предоставьте папку с набором файлов с именами **input<N>.txt** и **output<N>.txt**, где <N> – некоторое натуральное число. Файлы **input<N>.txt** содержат входные данные, а **output<N>.txt** ожидаемые выходные данные.
Например, в файле *output3.txt* запишите данные, которые правильная программа должна вывести, если ей подать на вход данные из файла *input3.txt*.
Созданные вами файлы будут использованы для проверки работы правильных и неправильных версий программы anagrams.exe.
Правильная программа – та, которая выдаёт результат в соответствии с требованиями.
Неправильная программа – та, которая содержит выдаёт результат, отличный от ожидаемого.
Набор подготовленных вами файлов должен быть таким, чтобы:

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

Задание 2. 60 баллов

Вступление

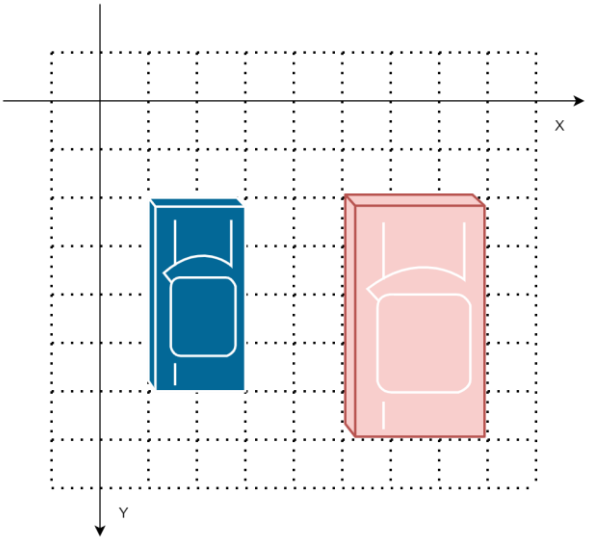
Иван хочет научиться разрабатывать игры. Первая игра, которую он хочет написать сам – это гонки. Но ему нужна помощь в обработке ситуаций, когда одна машина наезжает на другую, то есть происходит столкновение автомобилей. Помогите начинающему программисту и напишите программу, которая определяет, сталкиваются ли автомобили.

Требования к программе

Программа должна определять есть ли столкновение двух автомобилей и выводить результат в виде “YES” или ‘NO“

Входные данные

На вход программе подаются две строки, содержащие по 4 числа: X, Y, W, H. X и Y – координаты верхнего левого угла автомобиля. W, H – ширина и высота автомобиля.
Например, входные данные:
1 2 2 4
5 2 3 5
задают 2 автомобиля, отмеченные на рисунке синим и розовым цветом:



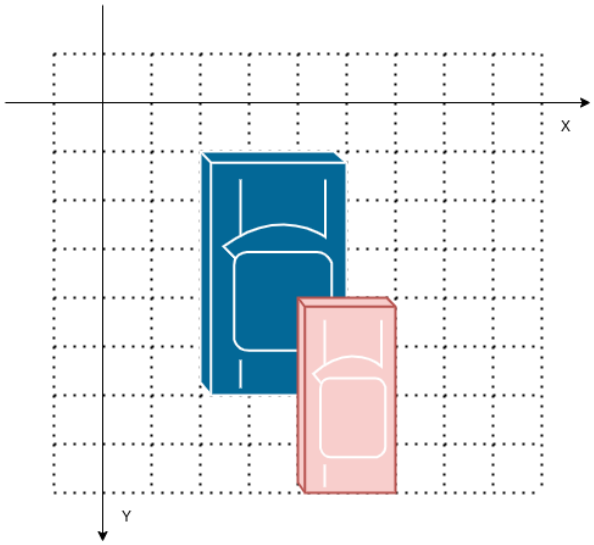
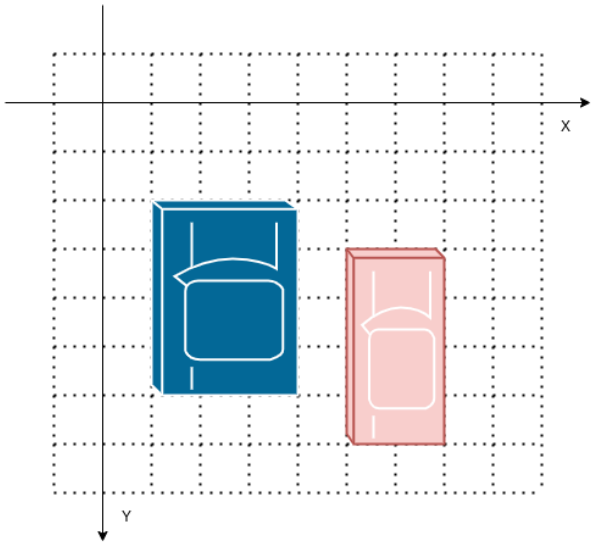
Синий автомобиль имеет координаты (1, 2) и размеры 2x4.
Розовый автомобиль имеет координаты (5, 2) и размеры 3x4.

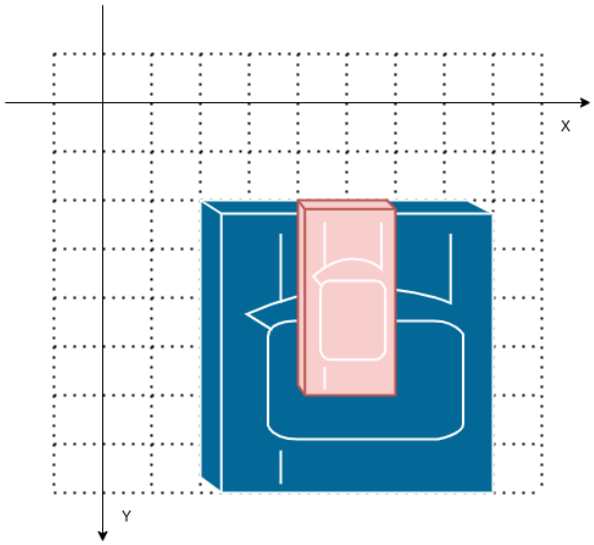
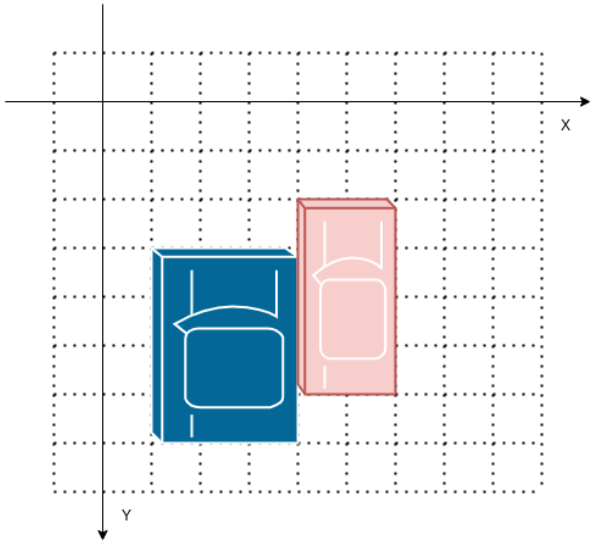
Гарантируется, что вводятся всегда только положительные целые числа от 1 до 1000.

Выходные данные

Сообщение о наличии наезда одного автомобиля на другой автомобиль, об отсутствии наезда.

Примеры

Входные данные	Выходные данные	Описание
2 1 3 5 4 3 2 4	YES	Автомобили сталкиваются 
1 2 3 4 5 3 2 4	NO	Автомобили не сталкиваются 
2 2 6 6 4 2 2 4	YES	Столкновение будет т.к. один автомобиль оказался внутри другого

		
1 3 3 4 4 2 2 4	NO	Столкновения не будет, т.к. два автомобиля находятся вплотную, но не сталкиваются 

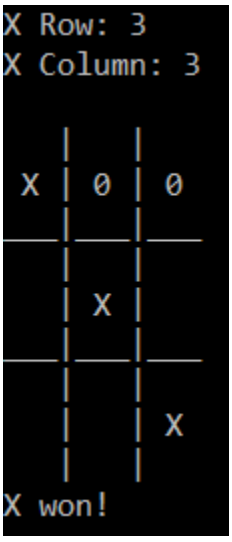
Задание 3. 80 баллов

Вступление

Ученик школы программирования Василий договорился с преподавателями, что играть он будет только в те компьютерные игры, которые сам написал. Одна из любимых игр Василия – “Крестики-нолики”. Помогите Василию написать программу, позволяющую играть двум пользователям в игру “Крестики-нолики” и анализирующую ситуацию игры с полем 3x3, чтобы он мог сыграть в эту игру со своим другом.

Требования к программе

Игроки по очереди ставят на свободные клетки поля знаки (один всегда крестики, другой всегда нолики). Первый, выстроивший в ряд 3 своих знака по вертикали, горизонтали или диагонали, выигрывает. Первый ход делает игрок, ставящий крестики. Игроки поочередно вводят координаты поля, в которое они хотят сделать ход. Координаты поля определяются двумя числами. Первое число указывает строку, число от 1 до 3 (выводится текст ‘X Row: ’ или ‘0 Row: ’ и далее считывается число). Второе число указывает столбец, число от 1 до 3 (выводится текст ‘X Column: ’ или ‘0 Column: ’ и далее считывается число). Например: чтобы сделать ход в верхний левый угол надо будет ввести 1 и 1. Для простоты, предполагается, что игроки всегда вводят корректные числа (от 1 до 3). После хода каждого игрока выводиться актуальное состояние игры (игровое поле). При выводе игрового поля необходимо использовать символы проблема (‘ ‘), вертикальной черты (‘|’) и нижнего подчеркивания (‘_’). Если один из пользователей выиграл, то игра заканчивается, и выводится соответствующее сообщение: ‘X won!’ или ‘0 won!’.



Функция вывода игрового поля:

```
PROCEDURE PrintGameField(C11, C12, C13, C21, C22, C23, C31, C32, C33: CHAR);
BEGIN
  WRITELN('   |   |   ');
  WRITELN(' ' + C11 + ' | ' + C12 + ' | ' + C13 + ' ');
  WRITELN('___|___|___');

  WRITELN('   |   |   ');
  WRITELN(' ' + C21 + ' | ' + C22 + ' | ' + C23 + ' ');
  WRITELN('___|___|___');

  WRITELN('   |   |   ');
  WRITELN(' ' + C31 + ' | ' + C32 + ' | ' + C33 + ' ');
  WRITELN('   |   |   ');
END;
```

Где C11, C12, C13, C21, C22, C23, C31, C32, C33 - это значения ячеек игрового поля.