

С вопросами по всем задачам десятого ДЗ можно писать Дианкину Игорю.

Задание 1 (1 балл)

Мнемоника: *breadth*

В задании дается граф, и несколько пар вершин этого графа. Требуется проверить поиском в ширину существование пути между этими вершинами.

Инпут читается из файла **in.txt**, записывается в файл **out.txt**.

В первых строках входного файла записан граф в формате списка смежности, затем в строках начинающихся с символа "\$" записаны пары вершин, по одной на строку.

Граф не ориентирован. Каждое ребро указывается единожды (Если в списке смежности уже есть "A B", то "B A" уже не будет).

В выходном файле следует печатать строку True, если путь между парой вершин есть, и False, если пути нет.

Input	0 1 1 \$0 1
Output	True

Input	0 1 2 1 2 2 3 3 \$0 1 \$2 1 \$1 3
Output	True True True

Задание 2 (1 балл)

Мнемоника: *deer*

В задании дается граф, и несколько пар вершин этого графа. Требуется проверить поиском в глубину существование пути между этими вершинами.

Инпут читается из файла **in.txt**, записывается в файл **out.txt**.

В первых строках входного файла записан граф в формате списка смежности, затем в строках начинающихся с символа "\$" записаны пары вершин, по одной на строку.

Граф ориентирован. Каждое ребро указывается единожды, и обратные друг другу ребра считаются разными (Если в списке смежности уже есть "А В", то "В А" может присутствовать).

В выходном файле следует печатать строку True, если путь между парой вершин есть, и False, если пути нет.

Input	0 1 2 1 0 2 \$0 1
Output	True

Input	0 1 1 2 2 3 3 2 4 4 5 \$0 1 \$3 2 \$4 3
Output	True True False

Задание 3 (1 балл)

Мнемоника: component

Дан неориентированный граф, требуется посчитать число связанных компонент в нем. В пустом графе 0 связанных компонент.

Иньут читается из файла **in.txt**, записывается в файл **out.txt**.

Так же как и в первом задании, ребра дважды не указываются.

Во входном файле список смежности, задающий требуемый граф. В выходном файле ожидается число в первой строке, равное числу компонент.

Input	0 1 2 1 2 3 3 4 5 5
Output	2

Задание 4 (2 балла)

Мнемоника: topo

Дан ориентированный ациклический граф. Требуется его топологически отсортировать. Инпут читается из файла **in.txt**, записывается в файл **out.txt**.

Граф может быть несвязным.

Во входном файле перечислены ребра графа, по одному ребру на строку. Подразумевается, что указанное ребро идет из первой указанной вершины во вторую.

Input	0 1 0 2 1 3
Output	0 2 1 3

Input	0 1 0 2 1 3 4 5
Output	4 5 0 2 1 3

Задание 5 (2 балла)

Мнемоника: short

Дан ориентированный ациклический граф и пары из его вершин. Требуется алгоритмом динамического программирования найти кратчайший путь между парами вершин.

Инпут читается из файла **in.txt**, записывается в файл **out.txt**.

В первых строках входного файла записан граф в формате списка ребер с весами (вершина1 вершина2 вес_ребра), как в четвертом задании, затем в строках начинающихся с символа "\$" записаны пары вершин, по одной на строку.

В выходном файле записывается кратчайший путь перечислением его вершин, либо False, если пути не существует.

Input	0 1 1 0 2 2 1 3 1 2 3 2 \$0 3 \$1 2
Output	0 1 3 False

Задание 6 (1 балл)

Мнемоника: path_count

Дан ориентированный ациклический граф и пары из его вершин. Требуется подсчитать количество путей между данными вершинами.

Инпут читается из файла **in.txt**, записывается в файл **out.txt**.

Граф может быть несвязным. Если путей нет, то их количество равно 0. Количество путей, из вершины в нее саму в этом задании равно нулю.

В первых строках входного файла записан граф в формате списка ребер, как в четвертом задании, затем в строках начинающихся с символа "\$" записаны пары вершин, по одной на строку.

В выходном файле записывается количество путей для каждой вершины. По одному значению на строку.

Input	0 1 0 2 1 3 2 3 \$0 3 \$1 2
Output	2 0

Input	0 1 0 2 0 3 0 4 0 5 1 5 2 5 3 5 4 5 \$0 5
-------	--

	\$3 5
Output	5 1

***Задание 7 (2 балла)**

Мнемоника: kosaraju

Дан ориентированный граф и пары из его вершин. Требуется подсчитать количество сильно связанных компонент в нем алгоритмом косарайо.

Подразумевается, что из каждой вершины есть путь в саму себя (в 0 из 0, в 1 из 1 и т.д.).

Названия вершин - случайные числа (может быть ситуация когда граф представлен вершинами 5 3 9).

Инпут читается из файла **in.txt**, записывается в файл **out.txt**.

В первых строках входного файла записан граф в формате списка ребер.

В выходном файле записывается количество связанных компонент.

Input	0 1
Output	2

Input	0 1 1 2 2 0
Output	1

Input	0 1 1 0
Output	1

Input	0 1 1 2 2 0 3 4 4 5 5 3
Output	2

Input	0 1 1 2 2 0 3 5
Output	2