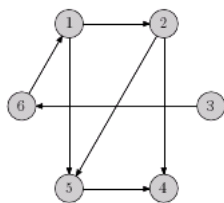


Графы: топологическая сортировка

Во всех задачах этого раздела храните граф в программе в виде *матрицы смежности* — двумерного массива `int A[N][N]`, где N — количество вершин в графе. Значение `A[i][j]` равно 1, если в графе есть ребро $i \rightarrow j$, и 0 в противном случае.

На рисунке изображён ориентированный граф (рёбра имеют направление) и его представление в виде матрицы смежности.



0	1	0	0	1	0
0	0	0	1	1	0
0	0	0	0	0	1
0	0	0	0	0	0
0	0	0	1	0	0
1	0	0	0	0	0

24	Чтение графа	3 балла
-----------	---------------------	----------------

Прочитайте из файла `input.txt` граф и напечатайте на экране матрицу смежности этого графа.

Файл имеет следующую структуру: сначала число N — количество вершин графа; затем M — количество рёбер, затем M пар чисел: начало и конец каждого ребра. Читайте числа при помощи функции `fscanf`, в этом случае расположение чисел по строкам не имеет значения. Граф на рисунке выше может быть задан в файле, например, так:

```
6 7           6 вершин, 7 рёбер:
1 2           ребро из 1 в 2,
1 5           ребро из 1 в 5
2 4           и т. д.
2 5
5 4
6 1
3 6
```

Результат для этого входного файла — такая же таблица, как в примере выше.

25	Вершина без входящих рёбер	4 балла
-----------	-----------------------------------	----------------

В данном графе (введённом из `input.txt`, как в предыдущей задаче) найдите вершину, в которую не входит ни одно ребро, и напечатайте её номер. Если таких вершин несколько, напечатайте номер любой из них.

26	Удаление рёбер из вершины	2 балла
-----------	----------------------------------	----------------

Удалите все рёбра, выходящие из вершины с данным номером (считайте граф из `input.txt`, а номер вершины — с клавиатуры). Напечатайте матрицу смежности получившегося графа.

27	Топологическая сортировка	6 баллов
-----------	----------------------------------	-----------------

Топологической сортировкой называют такое упорядочение вершин графа, при котором для каждого ребра его начальная вершина перечислена раньше, чем конечная. Конечно, речь идёт только об ориентированных графах.

Если представить себе, что вершины графа — это работы, которые нужно выполнить, а рёбра — зависимости между ними (чтобы вскипятить чайник, нужно сначала набрать воды, и т. п.), то топологическая сортировка покажет, в каком порядке нужно выполнять работы.

Отметим все вершины графа как неиспользованные (понадобится ещё один массив: `int used[N]`, изначально заполненный нулями). Для выполнения топологической сортировки найдите неиспользованную вершину, в которую не входит ни одно ребро, напечатайте её номер и отметьте её, как использованную. Затем удалите все рёбра, выходящие из этой вершины. Повторяйте эти действия, пока находятся неиспользованные вершины.

Для графа из примера топологическая сортировка выдаст ответ
3 6 1 2 5 4

28	Более одного решения	1 балл
-----------	-----------------------------	---------------

Приведите пример графа, для которого топологическая сортировка может дать два различных (правильных) ответа.

29	Нет решений	1 балл
-----------	--------------------	---------------

Приведите пример графа, для которого задача топологической сортировки не имеет решений.