

## Домашнее задание №5

*Задача 1.* Напишите функцию `inverse`, которая обращает квадратную матрицу вещественных чисел типа `double` размера  $K \times K$  методом Гаусса (без выбора ведущего элемента или с выбором ведущего элемента). Максимальный размер квадратной матрицы по одному измерению задаётся с помощью константы `MAX_N`, определённой в Вашей программе. Её значение должно быть 100. Действительный размер матрицы передаётся в функцию `inverse` первым параметром. Таким образом, функция должна иметь следующий прототип:

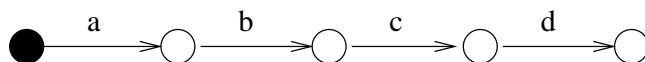
```
int inverse(int K, double in[*][*], double out[*][*]);
```

Функция `inverse` возвращает `-1`, если матрица `in` вырожденная (точнее, если на некотором шаге ведущий элемент оказался по модулю меньше  $1 \cdot 10^{-10}$ ). Если вычисление обратной матрицы завершилось успешно (то есть матрица невырожденная), возвращается `0`. Матрица `in` не должна изменяться как побочный эффект работы функции.

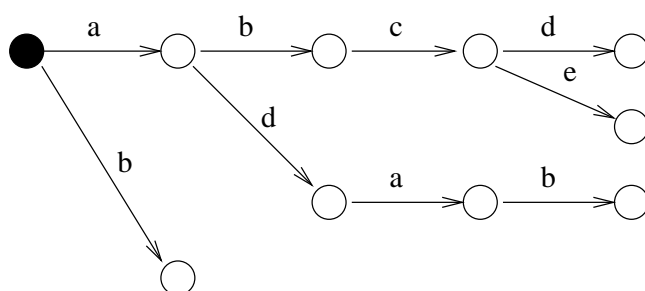
С использованием написанной Вами функции `inverse` напишите программу, которая считывает исходную матрицу со стандартного потока ввода и выводит обратную ей матрицу на стандартный поток вывода. Вы можете предполагать, что входные данные заданы корректно. Первым числом на стандартном потоке ввода задаётся число  $K$  — размер матрицы. Число  $K$  удовлетворяет ограничениям  $0 < K \leq \text{MAX\_N}$ . За ним идут  $K^2$  вещественных чисел, определяющих входную матрицу. Матрица задаётся по строкам.

Первым числом на стандартный поток вывода напечатайте результат работы функции `inverse` (`0` или `-1` в зависимости от того, вырождена ли матрица). Это число напечатайте на отдельной строке. Далее, если матрица невырождена, на стандартный поток вывода напечатайте  $K^2$  вещественных чисел, определяющих результирующую матрицу. Матрица должна быть напечатана по строкам. Каждое вещественное число должно быть напечатано с помощью спецификации формата `%.10g`.

**Задача 2.** Пусть дано множество строк  $S_1 \dots S_N$ . Каждую строку можно представить в виде линейного дерева, у которого символы, составляющие строку, написаны на ребрах. Например, для строки  $abcd$  получится такое линейное дерево:



Черным цветом отмечена вершина дерева. Теперь можно построить линейные деревья для всех строк и слить их в одно таким образом, чтобы вершины исходных линейных деревьев перешли в одну вершину нового дерева и из одной вершины выходило не более одной вершины, помеченной некоторой буквой. Такая структура называется *бор* (trie). Бор для строк  $abcd$ ,  $abce$ ,  $adab$ ,  $b$  показан ниже.



На стандартном потоке ввода на вход программе подаются символьные строки. Длина каждой строки не превышает 1023 символов. Строки разделены произвольным количеством пробельных символов. Постройте бор для введённых строк и выведите его на стандартный поток вывода. Вывод бора заключается в перечислении дуг бора. Каждая дуга печатается на отдельной строке. Дуга описывается одним символом, тем, который на ней находится, и двумя целыми числами: номером исходной вершины и номером целевой вершины. Корень дерева имеет номер 0, все остальные вершины нумеруются произвольным образом положительными числами. Номер каждой вершины должен быть уникален.

**Пример ввода:**

```

    abcd
abce adab
b

```

**Вывод для данного примера:**

```

a 0 1
b 0 2
b 1 3
d 1 4
c 3 5
a 4 6
d 5 7
c 5 8
b 6 9

```