

Задача А. Пути на доске

Имя входного файла: `paths.in`
Имя выходного файла: `paths.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Рассмотрим бесконечную клетчатую доску.

Назовём *путём* из одной клетки в другую последовательность клеток, в которой каждые две идущие подряд клетки являются соседними по стороне. Длина пути — это количество клеток в нём, не считая начальную.

Назовём путь *простым*, если в нём не встречается двух одинаковых клеток.

Зафиксируем какую-то клетку на доске. Сколько существует простых путей заданной длины, начинающихся в этой клетке?

Формат входного файла

В первой строке входного файла задано целое число n ($1 \leq n \leq 10$).

Формат выходного файла

В первой строке выходного файла выведите одно число — количество путей длины n из этой клетки.

Примеры

<code>paths.in</code>	<code>paths.out</code>
1	4
2	12

Задача В. Самый длинный путь

Имя входного файла: `path.in`
Имя выходного файла: `path.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

В данном ориентированном графе найдите самый длинный путь такой, что каждая вершина графа встречается в нём не более одного раза.

Формат входного файла

В первой строке входного файла заданы через пробел два целых числа n и m ($1 \leq n \leq 22$, $0 \leq m \leq 1000$). В следующих m строках заданы рёбра графа в формате $u_i v_i$ — номера начальной и конечной вершин ребра i , соответственно. Граф может содержать петли и кратные рёбра.

Формат выходного файла

В первой строке выходного файла выведите длину искомого пути l . Во второй строке выведите $l + 1$ число через пробел — вершины пути в порядке обхода. Если оптимальных ответов несколько, можно вывести любой из них.

Примеры

<code>path.in</code>	<code>path.out</code>
3 3 1 2 2 3 3 1	2 1 2 3
4 6 1 2 2 1 2 3 2 4 3 2 4 2	2 1 2 4
5 3 3 2 2 2 1 5	1 3 2

Задача С. String Decomposition

Имя входного файла: `strdec.in`
Имя выходного файла: `strdec.out`
Ограничение по времени: 3 seconds
Ограничение по памяти: 256 megabytes

As you remember, Vasya works in RIGS. And now... Yes, he faced another problem with strings. The building of RIGS is very big, and Vasya wants to send a string to his friend Petr who works at the other end of the building. However, the sending equipment is rather strange — it cannot deal with strings with lengths greater than k . So Vasya has to divide his string into small blocks.

But there is another problem — the sending equipment asks one coin per string. Of course, Vasya doesn't want to pay too much. Luckily, he knows that the string sending equipment software contains a bug: if Vasya sends two or more equal strings, it will ask for money only for the first time when such a string was sent. This works even if Vasya sends some other strings between the equal ones. Now, Vasya wants to send his

string, and he wants to pay as little as possible. The coins are considered equal.

Write a program which will find the minimal number of coins required to send a given string.

Формат входного файла

The first line contains a non-empty string S (length of S is not greater than 40). The second line contains one integer k ($1 \leq k \leq 20$). It is guaranteed that S contains only lowercase Latin letters.

Формат выходного файла

Write only one integer — the minimal cost of sending S to Petr.

Примеры

<code>strdec.in</code>	<code>strdec.out</code>
<code>cabbaabc</code> <code>3</code>	<code>3</code>
<code>tatata</code> <code>2</code>	<code>1</code>