

Содержание

| | | |
|---|---------------------------------|---|
| 1 | Задача А. Любители Кошек | 2 |
| 2 | Задача В. Ненокку (простая) | 3 |
| 3 | Задача С. Больше чёрного! | 4 |
| 4 | Задача D. Раскраска в три цвета | 5 |

1 Задача А. Любители Кошек

Имя входного файла: `catlover.in`
Имя выходного файла: `catlover.out`
Ограничение по времени: 0.2 секунды
Ограничение по памяти: 256 мегабайт

В университетском клубе любителей кошек зарегистрировано n членов. Естественно, что некоторые из членов клуба знакомы друг с другом. Нужно сосчитать, сколькими способами можно выбрать из них троих, которые могли бы свободно общаться (то есть, любые два из которых знакомы между собой).

Формат входных данных

В первой строке входного файла заданы числа n и m ($1 \leq n \leq 1000$, $1 \leq m \leq 30\,000$), где m обозначает общее число знакомств. В последующих m строках идут пары чисел $a_i b_i$, обозначающие, что a_i знаком с b_i . Информация об одном знакомстве может быть записана несколько раз, причем даже в разном порядке (как (x, y) , так и (y, x)).

Формат выходных данных

В выходной файл необходимо вывести количество способов выбрать троих попарно знакомых друг с другом людей из клуба.

Пример

| <code>catlover.in</code> | <code>catlover.out</code> |
|--------------------------|---------------------------|
| 3 3 1 2 2 3 3 1 | 1 |

2 Задача В. Ненокку (простая)

Имя входного файла: `nenokku_easy.in`
Имя выходного файла: `nenokku_easy.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Очень известный автор не менее известной книги решил написать продолжение своего произведения. Он писал все свои книги на компьютере, подключенном к интернету. Из-за такой неосторожности мальчику Ненокку удалось получить доступ к еще ненаписанной книге. Каждый вечер мальчик залазил на компьютер писателя и записывал на свой компьютер новые записи. Ненокку, записав на свой компьютер очередную главу, заинтересовался, а использовал ли хоть раз писатель слово “книга”. Но он не любит читать книги (он лучше ползает в интернете), и поэтому он просит вас узнать есть ли то или иное слово в тексте произведения. Но естественно его интересует не только одно слово, а достаточно много.

Формат входных данных

В каждой строчке входного файла записано одна из двух записей.

1. ? <слово> (<слово> - это набор не более 50 латинских символов);
2. A <текст> (<текст> - это набор не более 1024 латинских символов).

1 означает просьбу проверить существование подстроки <слово> в произведении.

2 означает добавление в произведение <текст>.

Число запросов — не более 30. Входной файл содержит не более 1 килобайта.

Формат выходных данных

Выведите на каждую строчку типа 1 “YES”, если существует подстрока <слово>, и “NO” в противном случае. Не следует различать регистр букв.

Пример

| <code>nenokku_easy.in</code> | <code>nenokku_easy.out</code> |
|------------------------------|-------------------------------|
| ? love | NO |
| ? is | NO |
| A Loveis | YES |
| ? love | NO |
| ? WHO | YES |
| A Whoareyou | |
| ? is | |

3 Задача С. Больше чёрного!

Имя входного файла: `black.in`
Имя выходного файла: `black.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дана прямоугольная доска $w \times h$, состоящая из квадратов 1×1 . Некоторые из квадратов отсутствуют. Нужно покрасить оставшиеся квадраты в чёрный и белый так, чтобы число чёрных было как можно больше, и чтобы никакие два одноцветных квадрата не имели общей стороны.

Формат входных данных

В первой строке ввода записаны числа w и h — размеры доски ($1 \leq w, h \leq 50$). В следующих h строках записано по w символов: «.» означает, что соответствующий квадрат отсутствует, «#» — обратное.

Формат выходных данных

Выведите доску, раскрашенную указанным образом. Среди всех оптимальных решений выведите лексикографически минимальное. Чёрный квадрат обозначается символом «b», белый — «w».

Примеры

| <code>black.in</code> | <code>black.out</code> |
|--|--|
| <code>3 3 .#. ### .#.</code> | <code>.b. bwb .b.</code> |
| <code>6 6 #.#.#. .#.#.# #.#.#. .#.#.# #.#.#. .#.#.#</code> | <code>b.b.b. .b.b.b b.b.b. .b.b.b b.b.b. .b.b.b</code> |
| <code>6 1 #####</code> | <code>bwbwbw</code> |
| <code>3 4 .#. .#. ### .#.</code> | <code>.w. .b. bwb .b.</code> |

4 Задача D. Раскраска в три цвета

Имя входного файла: `coloring.in`
Имя выходного файла: `coloring.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Сколько способов покрасить n домов в три цвета — c_1 домов в красный, c_2 в синий и c_3 в зелёный?

Формат входных данных

В первой строке входного файла заданы через пробел три целых числа c_1 , c_2 и c_3 ($0 \leq c_1, c_2, c_3 \leq 20$, кроме того, их сумма $n = c_1 + c_2 + c_3 \leq 20$).

Формат выходных данных

В первой строке выходного файла выведите искомое количество способов.

Примеры

| <code>coloring.in</code> | <code>coloring.out</code> |
|--------------------------|---------------------------|
| 1 1 1 | 6 |
| 2 1 1 | 12 |
| 0 1 2 | 3 |