

Задача А. Суффиксное дерево

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

Постройте суффиксное дерево для заданной строки s .

Формат входного файла

Первая строка входного файла содержит строку s ($1 \leq |s| \leq 100\,000$). Строка состоит из строчных латинских букв.

Формат выходного файла

В первой строке выходного файла выведите два натуральных числа n и m , разделенных пробелом — число вершин и ребер в суффиксном дереве соответственно. В следующих m строках выведите описания ребер в формате `<родитель> <потомок> <l> <r>`. Эта запись означает, что на ребре написана строка $s[l..r]$, при этом значение l должно быть минимально возможным. Корнем дерева должна быть вершина с номером 1. Вершины должны быть занумерованы натуральными числами, не превышающими n .

Примеры

suffixtree.in	suffixtree.out
ababb	7 6 1 4 1 2 1 6 2 2 4 2 3 5 4 5 5 5 6 3 3 5 6 7 5 5

Задача В. Ненокку

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

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 входного файла

В каждой строчке входного файла записано одна из двух записей.

- ? <слово> (<слово> - это набор не более 50 латинских символов);
- A <текст> (<текст> - это набор не более 10^5 латинских символов).

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

2 означает добавление в произведение <текст>.

Писатель только начал работать над произведением, поэтому он не мог написать более 10^5 символов. А входной файл содержит не более 15 мегабайт информации.

Формат выходного файла

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

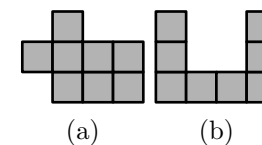
Примеры

nenokku.in	nenokku.out
? love	NO
? is	NO
A Loveis	YES
? love	NO
? WHO	YES
A Whoareyou	
? is	

Задача С. Выпуклые пермутмино

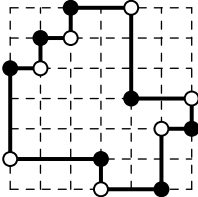
Имя входного файла: `permutominoes.in`
 Имя выходного файла: `permutominoes.out`
 Ограничение по времени: 2 секунды
 Ограничение по памяти: 256 мегабайт

Полимино — это связанное множество клеточек на клетчатой доске. Полимино называется *выпуклым*, если каждая строка и каждый столбец полимино являются связными. Например, полимино на картинке (а) является выпуклым, а полимино, изображенное на картинке (b) не является выпуклым.



Полимино называется *пермутомино*, если оно состоит из $2 \cdot n$ вершин, все его вершины имеют координаты от 1 до n , и в нем нет ни двух вертикальных отрезков с одинаковой

x -координатой, ни двух горизонтальных отрезков с одинаковой y -координатой. Картинка ниже иллюстрирует пермутомино с 14 вершинами. Заметим, что это пермутомино является выпуклым.



Вам дано число n . Ваша задача состоит в том, чтобы посчитать количество различных выпуклых пермутомино с $2 \cdot n$ вершинами. Два пермутомино считаются равными, если их можно наложить одно на другое, при этом повороты и отражения не разрешаются.

Формат входного файла

Единственное число n ($2 \leq n \leq 30$).

Формат выходного файла

Одно число — количество выпуклых пермутомино с $2 \cdot n$ вершинами.

Примеры

permutominoes.in	permutominoes.out
2	1

Note

Помните, что вам следует считать только выпуклые пермутомино.

