

Задача А. Замощение доминошками 1

Имя входного файла: *стандартный ввод*
 Имя выходного файла: *стандартный вывод*
 Ограничение по времени: 2 секунды
 Ограничение по памяти: 512 мегабайт

Доминошка на клетчатой плоскости — это фигура из двух клеток, имеющих общую сторону. Замощение прямоугольника из $m \times n$ клеток доминошками — это такой набор доминошек, целиком лежащих внутри прямоугольника, что каждая клетка прямоугольника покрыта ровно одной из них. Два замощения P и Q считаются различными, если существуют доминошки $p \in P$ и $q \in Q$ такие, что одна из клеток p и q — общая, а другая клетка p отличается от другой клетки q .

Сколько существует способов замостить прямоугольник $m \times n$ доминошками?

Формат входных данных

Первая строка ввода содержит два целых числа m и n — высота и ширина прямоугольника, соответственно ($1 \leq m, n \leq 12$).

Формат выходных данных

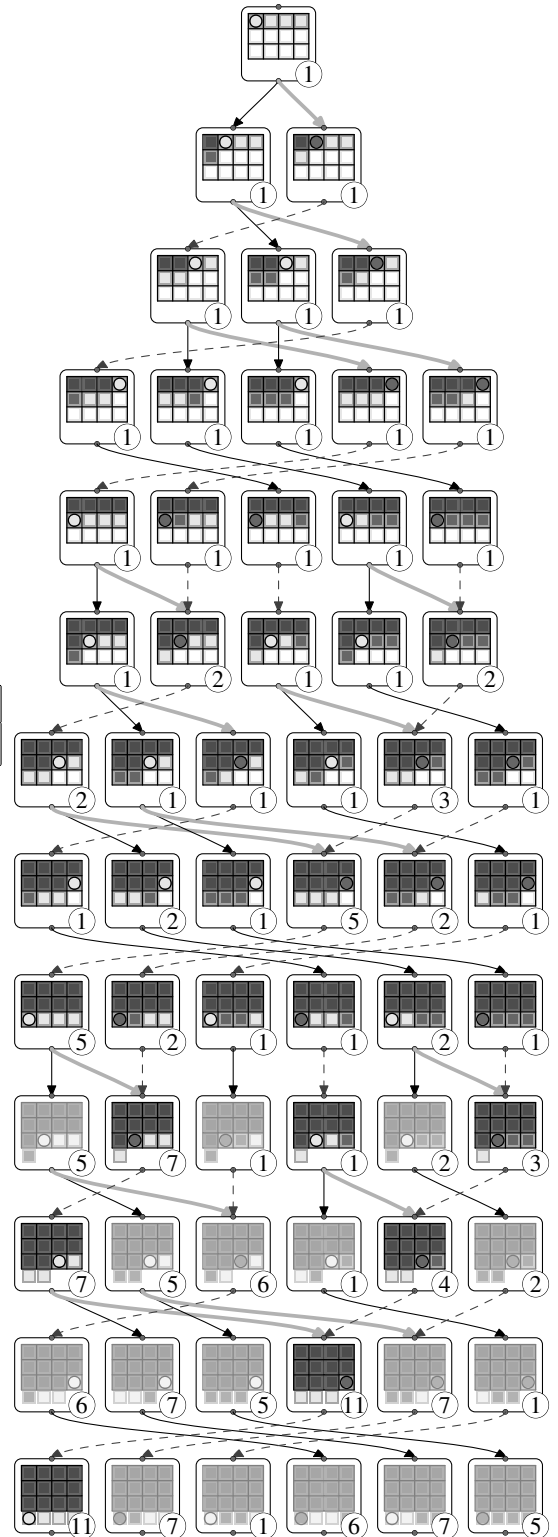
Выведите одно число — количество способов замостить прямоугольник $m \times n$ доминошками.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 4	11

Пояснение к примеру

На диаграмме справа представлен ход решения этой задачи динамическим программированием по изломанному профилю. Профиль состоит из n клеток. В каждом прямоугольнике показаны закрашенные клетки и текущий профиль. Тёмные клетки — заполненные, светлые — пустые. Клетки профиля имеют серые рамки, клетки над профилем — тёмные, а клетки под профилем — светлые. Текущая рассматриваемая клетка — первая клетка профиля — обозначена кружком. Каждому уровню диаграммы соответствует одна и та же текущая клетка. Число в левом нижнем углу — количество способов получить такое состояние. Стрелками обозначены переходы между состояниями. Жирной серой стрелке соответствует укладка горизонтальной доминошки, тонкой чёрной стрелке — укладка вертикальной доминошки. Пунктирная стрелка означает, что первая клетка профиля уже занята, и доминошка в этом переходе не укладывается. Состояния, в которых профиль выходит за границы прямоугольника, затенены.



Задача В. Замощение доминошками 2

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Доминошка на клетчатой плоскости — это фигура из двух клеток, имеющих общую сторону. Замощение прямоугольника из $m \times n$ клеток доминошками — это такой набор доминошек, целиком лежащих внутри прямоугольника, что каждая клетка прямоугольника покрыта ровно одной из них. Два замощения P и Q считаются различными, если существуют доминошки $p \in P$ и $q \in Q$ такие, что одна из клеток p и q — общая, а другая клетка p отличается от другой клетки q .

Сколько существует способов замостить прямоугольник $m \times n$ доминошками? В этой версии задачи некоторые клетки прямоугольника могут быть вырезаны.

Формат входных данных

Первая строка ввода содержит два целых числа m и n — высота и ширина прямоугольника, соответственно ($1 \leq m, n \leq 12$). Следующие m строк содержат по n символов каждая и описывают прямоугольник. Символ '.' (точка) соответствует обычной клетке, а символ '#' (решётка) — вырезанной. Гарантируется, что в этих m строках ввода никакие другие символы не встречаются.

Формат выходных данных

Выведите одно число — количество способов замостить заданный прямоугольник $m \times n$ доминошками.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 4	11
2 2 .. .#	0
2 3 #.. ..#	1

Пояснение к примеру

Первый пример совпадает с примером из первой версии задачи.

Во втором примере три невырезанные клетки невозможно покрыть доминошками.

В третьем примере единственный способ покрытия — положить две доминошки горизонтально.

Задача С. Замощение уголками 1

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Уголок на клетчатой плоскости — это фигура из трёх клеток, получающаяся из квадрата 2×2 клетки удалением одной любой клетки. Замощение прямоугольника из $m \times n$ клеток уголками — это такой набор уголков, целиком лежащих внутри прямоугольника, что каждая клетка прямоугольника покрыта ровно одним из них. Два замощения P и Q считаются различными, если существуют уголки $p \in P$ и $q \in Q$ такие, что хотя бы одна из клеток p и q — общая, но p и q не совпадают полностью.

Сколько существует способов замостить прямоугольник $m \times n$ уголками?

Формат входных данных

Первая строка ввода содержит два целых числа m и n — высота и ширина прямоугольника, соответственно ($1 \leq m, n \leq 14$).

Формат выходных данных

Выведите одно число — количество способов замостить прямоугольник $m \times n$ уголками.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 3	2
3 3	0

Пояснения к примерам

В первом примере существует два способа замощения двумя уголками — назовём их левым и правым. Первый столбец прямоугольника целиком содержится в левом уголке, а третий — в правом. Во втором столбце либо верхняя клетка содержится в левом уголке, а нижняя — в правом, либо наоборот.

Во втором примере замощение невозможно: в девять клеток помещаются только три уголка, однако никакие две угловые клетки квадрата 3×3 не могут быть покрыты одним и тем же уголком.

Задача D. Замощение уголками 2

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	5 секунд
Ограничение по памяти:	512 мегабайт

Уголок на клетчатой плоскости — это фигура из трёх клеток, получающаяся из квадрата 2×2 клетки удалением одной любой клетки. Заметим, что всего существует четыре типа уголков — по количеству различных клеток, которые могут быть вырезаны.

Замощение прямоугольника из $m \times n$ клеток уголками — это такой набор уголков, целиком лежащих внутри прямоугольника, что каждая клетка прямоугольника покрыта ровно одним из них. Два замощения P и Q считаются различными, если существуют уголки $p \in P$ и $q \in Q$ такие, что хотя бы одна из клеток p и q — общая, но p и q не совпадают полностью.

Сколько существует таких способов замостить прямоугольник $m \times n$ уголками, что в этих замощениях не присутствует хотя бы один из типов уголков?

Формат входных данных

Первая строка ввода содержит два целых числа m и n — высота и ширина прямоугольника, соответственно ($1 \leq m, n \leq 15$).

Формат выходных данных

Выведите одно число — количество способов замостить прямоугольник $m \times n$ уголками так, чтобы в этих замощениях не присутствовал хотя бы один из типов уголков.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 3	2
3 3	0

Пояснения к примерам

В первом примере существует два способа замощения двумя уголками — назовём их левым и правым. Первый столбец прямоугольника целиком содержится в левом уголке, а третий — в правом. Во втором столбце либо верхняя клетка содержится в левом уголке, а нижняя — в правом, либо наоборот.

Во втором примере замощение невозможно: в девять клеток помещаются только три уголка, однако никакие две угловые клетки квадрата 3×3 не могут быть покрыты одним и тем же уголком.

Задача Е. Зоопарк

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

Гордость Азиатско-Тихоокеанского региона — новый Большой Круговой Зоопарк. Расположенный на маленьком острове в Тихом океане, он содержит огромное количество вольеров с разными экзотическими животными (в разных вольерах — разные животные).

Вы отвечаете за работу с посетителями зоопарка, то есть ваша задача — делать людей счастливыми. Только что приехал автобус со школьниками, и вы спешите им угодить. Тем не менее, это не простая задача: есть животные, которых некоторые дети любят, и есть животные, которых некоторые дети боятся. К примеру, маленький Алекс любит обезьян и коал, потому что они милые, но боится львов из-за их острых зубов. С другой стороны, Полли любит львов за их красивые гривы, но боится коал, потому что они до ужаса неприятно пахнут. Вы решили перевести часть животных в запасные вольеры на время экскурсии, чтобы не напугать детей. Но вас огорчает, что если вы уберёте из основных вольеров слишком много животных, детям будет не на что смотреть. Ваша задача — решить, каких животных надо перевести в запасные вольеры, чтобы как можно больше детей были счастливы.

Каждый ребёнок стоит снаружи круга из вольеров и видит $2k + 1$ соседних вольеров. Формально, если ребёнок стоит рядом с вольером с номером T , то он видит вольеры с номерами $(T - k, \dots, T - 1, T, T + 1, \dots, T + k)$.

Про каждого ребёнка вы знаете, как он относится к животным, которые находятся в его поле зрения. Ребёнок будет счастлив, если будет верно хотя бы одно из следующих утверждений:

- Хотя бы одно животное, которого он боится, на время экскурсии убрали из вольера.
- Хотя бы одно животное, которое он любит, будет в вольере во время экскурсии.

Например, пусть таблица снизу описывает приехавших детей:

Дети	Видимые вольеры	Страшные животные	Любимые животные
Алекс	2, 3, 4, 5, 6	4	2, 6
Полли	3, 4, 5, 6, 7	6	4
Чайтаня	6, 7, 8, 9, 10	9	6, 8
Хван	8, 9, 10, 11, 12	9	12
Ка-Шу	12, 13, 14, 1, 2	12, 13, 2	—

Предположим, что мы уберём только животных из вольеров 4 и 12. Это сделает Алекса и Ка-Шу счастливыми, потому что из их поля зрения исчезнет страшное животное. Чайтаню это тоже осчастливит, потому что в вольерах 6 и 8 находятся её любимые животные. К сожалению, в этом случае Полли и Хван расстроятся, потому что не увидят любимых животных, но зато будут вынуждены будут смотреть на животных, которые их пугают. Таким образом, эта расстановка животных приносит счастье трём детям.

Теперь предположим, что мы вернём животное обратно в 12-й вольер и удалим животное из 6-го вольера. Алекс и Полли будут счастливы, потому что животные из вольеров 4 и 6, которых они боялись, пропали. Чайтаня будет счастлива, потому что, несмотря на то, что животное из 6-го вольера исчезло, она всё ещё видит своё любимое животное в 8-м вольере. Более того, Хван тоже будет счастлива, потому что она может видеть животное в 12-м вольере, которое она любит. В печали будет только Ка-Шу.

Наконец, вернём всех животных на свои места и уберём животное из 13-го вольера. Теперь Ка-Шу будет рад, так как животное, которого он боится, пропадёт, и Алекс, Полли, Чайтаня и Хван будут счастливы, так как они все видят любимое животное. Таким образом мы осчастливим всех школьников.

Формат входных данных

Первая строка входного файла содержит три целых числа: N ($N \leq 10\,000$) — количество вольеров, C ($C \leq 50\,000$) — количество детей, K ($0 \leq K \leq 2$) — угол обзора ребёнка. Следующие C строк описывают детей. Каждая из этих строк содержит $2k+2$ целых числа: сначала число T ($1 \leq T \leq N$) — вольер, рядом с которым стоит ребёнок, а далее — $2k+1$ число, описывающее отношение ребёнка к животным с номерами от $T-k$ до $T+k$, находящимся в его поле зрения:

- -1 , если он боится этого животного,
- 0 , если он спокойно относится к этому животному, и
- 1 , если он любит это животное.

Вольеры пронумерованы от числами 1 до N по часовой стрелке. Везде в условии, где упоминаются номера вольеров, следует считать, что они зациклены, то есть вольер с номером $N+1$ — это 1-й вольер, вольер с номером 0 — это вольер с номером N .

Формат выходных данных

Выходной файл должен содержать единственное целое число — максимальное количество детей, которых можно сделать счастливыми.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
<pre>14 5 2 4 -1 0 1 0 -1 5 0 -1 0 1 0 8 -1 0 -1 1 0 10 0 1 0 0 -1 14 1 1 0 0 1</pre>	5
<pre>12 7 2 3 1 0 0 0 -1 7 1 0 -1 0 0 7 -1 0 -1 0 -1 9 1 0 -1 0 0 11 1 0 -1 0 0 11 1 0 1 0 1 1 1 0 -1 0 0</pre>	6

Задача F. Игра «Жизнь»

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	6 секунд
Ограничение по памяти:	512 мегабайт

Что наша игра? Жизнь!

Дана клетчатая доска $W \times H$. Каждая клетка исходно является либо чёрной, либо белой. Клетки называются соседними, если у них есть общая сторона.

Повторяется следующая процедура:

- Находится белая клетка, у которой не менее двух чёрных соседей. Если такой клетки нет, процесс завершается.
- Найденная клетка перекрашивается в чёрный цвет.

Напишите программу, которая вычислит количество различных раскрасок доски, которые могут получиться по завершении процесса.

Поскольку ответ может быть большим, достаточно найти его остаток по модулю 10^{18} .

Формат входных данных

В первой строке ввода записаны размеры доски: два целых числа W и H ($1 \leq W, H \leq 13$). В следующих H строках записано по W символов, описывающих саму доску:

- Символ «В» обозначает, что клетка в начальной раскраске была чёрной.
- Символ «.» обозначает, что исходный цвет клетки неизвестен, то есть он может быть как чёрным, так и белым.

Формат выходных данных

Выведите одно целое число: остаток количества различных возможных конечных раскрасок по модулю 10^{18} .

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 4 В.В .В.	3

Пояснение к примеру

В данном примере возможны следующие конечные раскраски (здесь символом «W» обозначаются белые клетки):

```

ВВВ ВВВ ВВВ
ВВВ ВВВ ВВВ
WWW ВВВ ВВВ
WWW WWW ВВВ

```

Задача G. Покраски домино

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 4 секунды
Ограничение по памяти: 512 мегабайт

Алиса хочет стать художницей. Для этого нужно много практики. Но ей нечего рисовать! Она попросила Боба помочь ей с этим, и он кое-что придумал.

Бобу нравится домино. Его доминошки покрашены: одна половина белая, другая чёрная. Он решил замостить прямоугольную доску $n \times m$ доминошками и дать её Алисе, чтобы она её нарисовала.

Рисунок будет выглядеть как прямоугольная решётка $n \times m$, в которой каждая клетка покрашена в чёрный или белый цвет, а как именно лежали доминошки — не будет нарисовано. Боб может замостить доску многими способами, так что Алиса сможет сделать много рисунков. Но иногда рисунки разных замощений выглядят одинаково.

Боб хочет, чтобы Алиса сделала как можно больше рисунков, но он не хочет, чтобы Алиса тратила своё время впустую, рисуя одно и то же. Помогите ему посчитать количество различных рисунков, которые Алиса может нарисовать. Так как ответ может быть очень большим, выведите его по модулю $10^9 + 7$.

Формат входных данных

Первая строка содержит целые числа n и m : размеры доски ($1 \leq n \leq 6$, $1 \leq m \leq 300$).

Формат выходных данных

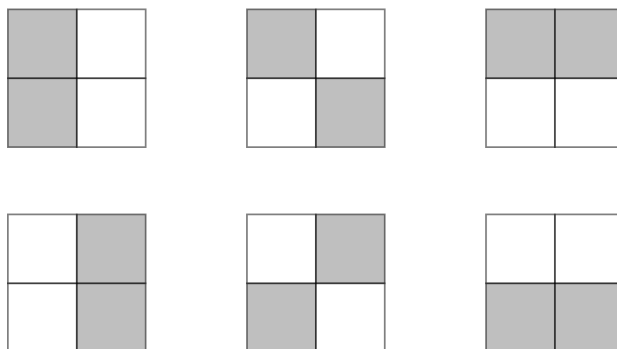
Выведите одно число: количество различных рисунков по модулю $10^9 + 7$.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 2	6
2 3	16
3 3	0

Пояснения к примерам

В первом примере шесть рисунков:



Заметьте, что второй рисунок может быть получен из двух замощений (границы доминошек выделены жирным):



Задача Н. Длинные домино

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Найдите количество способов замостить прямоугольник размера $m \times n$ длинными домино — прямоугольниками размера 3×1 .

Каждое домино должно полностью находиться внутри прямоугольника, домино не должны накладываться.

Формат входных данных

Входной файл содержит m и n ($1 \leq m \leq 9$, $1 \leq n \leq 30$).

Формат выходных данных

Выведите количество способов замостить прямоугольник $m \times n$ длинными домино.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 3	2
3 10	28

Задача I. АВ-таблица по номеру

Имя входного файла: *стандартный ввод*
 Имя выходного файла: *стандартный вывод*
 Ограничение по времени: 2 секунды
 Ограничение по памяти: 512 мегабайт

АВ-таблицей называется прямоугольная таблица, в клетках которой стоят заглавные латинские буквы А и В, обладающая следующими двумя свойствами:

- Для каждой буквы А в таблице длина горизонтального блока, образованного соседними буквами А и содержащего эту букву, чётна;
- Для каждой буквы В в таблице длина вертикального блока, образованного соседними буквами В и содержащего эту букву, чётна.

Вот несколько примеров АВ-таблиц:

А	А
---	---

В	В
В	В

В	В	А	А
В	В	В	В
А	А	В	В

Следующие таблицы, напротив, АВ-таблицами не являются.

В	В
---	---

А	А
В	В

В	А	А	В
В	А	А	В
В	А	А	В

Зачёркнутые буквы — это либо буквы А, лежащие в горизонтальных блоках нечётной длины, либо буквы В, лежащие в вертикальных блоках нечётной длины.

Рассмотрим все АВ-таблицы из m строк и n столбцов. Говорят, что АВ-таблица u *меньше* АВ-таблицы v , если в первой клетке, в которой эти две таблицы различаются, в таблице u стоит буква А, а в таблице v — буква В. Клетки в таблице упорядочены по возрастанию номера строки, а клетки в одной строке — по возрастанию номера столбца — так же, как и обычно при чтении текста.

Отношение «меньше» задаёт порядок на АВ-таблицах размера $m \times n$. К примеру, ниже показаны все три АВ-таблицы 2×3 в этом порядке:

А	А	В
А	А	В

В	А	А
В	А	А

В	В	В
В	В	В

По данным m , n и k найдите k -ю АВ-таблицу размера $m \times n$ в указанном порядке.

Формат входных данных

В первой строке входного файла заданы через пробел три целых числа m , n и k ($1 \leq m, n \leq 12$). Здесь m — высота таблицы, n — её ширина, а k — порядковый номер требуемой АВ-таблицы размера $m \times n$. Таблицы нумеруются, начиная с нуля. Гарантируется, что требуемая таблица существует.

Формат выходных данных

Первые m строк выходного файла должны содержать по n символов каждая и задавать k -ю АВ-таблицу размера $m \times n$.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 3 0	ААВ ААВ
2 3 1	ВАА ВАА
2 3 2	ВВВ ВВВ

Задача J. Номер по АВ-таблице

Имя входного файла: *стандартный ввод*
 Имя выходного файла: *стандартный вывод*
 Ограничение по времени: 2 секунды
 Ограничение по памяти: 512 мегабайт

АВ-таблицей называется прямоугольная таблица, в клетках которой стоят заглавные латинские буквы А и В, обладающая следующими двумя свойствами:

- Для каждой буквы А в таблице длина горизонтального блока, образованного соседними буквами А и содержащего эту букву, чётна;
- Для каждой буквы В в таблице длина вертикального блока, образованного соседними буквами В и содержащего эту букву, чётна.

Вот несколько примеров АВ-таблиц:

А	А
---	---

В	В
В	В

В	В	А	А
В	В	В	В
А	А	В	В

Следующие таблицы, напротив, АВ-таблицами не являются.

В	В
---	---

А	А
В	В

В	А	А	В
В	А	А	В
В	А	А	В

Зачёркнутые буквы — это либо буквы А, лежащие в горизонтальных блоках нечётной длины, либо буквы В, лежащие в вертикальных блоках нечётной длины.

Рассмотрим все АВ-таблицы из m строк и n столбцов. Говорят, что АВ-таблица u *меньше* АВ-таблицы v , если в первой клетке, в которой эти две таблицы различаются, в таблице u стоит буква А, а в таблице v — буква В. Клетки в таблице упорядочены по возрастанию номера строки, а клетки в одной строке — по возрастанию номера столбца — так же, как и обычно при чтении текста.

Отношение «меньше» задаёт порядок на АВ-таблицах размера $m \times n$. К примеру, ниже показаны все три АВ-таблицы 2×3 в этом порядке:

А	А	В
А	А	В

В	А	А
В	А	А

В	В	В
В	В	В

По данным m , n и АВ-таблице размера $m \times n$ найдите её номер в указанном порядке.

Формат входных данных

В первой строке входного файла заданы через пробел два целых числа m и n ($1 \leq m, n \leq 12$). Здесь m — высота таблицы, а n — её ширина. Следующие m строк входного файла содержат по n символов каждая и корректно задают АВ-таблицу размера $m \times n$.

Формат выходных данных

Выведите порядковый номер данной АВ-таблицы среди всех АВ-таблиц размера $m \times n$. Таблицы нумеруются, начиная с нуля.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 3 AAB AAB	0
2 3 BAA BAA	1
2 3 BBB BBB	2