

Содержание

Must have	2
Задача 9А. Сумма всего подряд [0.8 sec, 256 mb]	2
Задача 9В. Симпатичные узоры [1 sec, 256 mb]	3
Обязательные задачи	4
Задача 9С. Крышки [2 sec, 256 mb]	4
Задача 9D. Быстрое пересечение множеств [1 sec, 256 mb]	5
Задача 9Е. Симпатичные узоры 2 [2 sec, 256 mb]	6
Задача 9F. Покрытие доминошками [2 sec, 256 mb]	7
Дополнительные задачи	8
Задача 9G. Игра «Жизнь» [3 sec, 256 mb]	8
Задача 9H. Симпатичные узоры 3 [2 sec, 256 mb]	9
Задача 9I. Кони на гриде [2 sec, 256 mb]	10
Задача 9J. Длинные домино [1 sec, 256 mb]	11

Вы не умеете читать/выводить данные, открывать файлы? Воспользуйтесь **примерами**.

В некоторых задачах большой ввод и вывод. Пользуйтесь **быстрым вводом-выводом**.

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) **переопределение стандартного аллокатора** ускорит вашу программу.

Обратите внимание на компилятор GNU C++11 5.1.0 (TDM-GCC-64) inc, который позволяет пользоваться **дополнительной библиотекой**. Под ним можно сдать **вот это**.

Must have

Задача 9А. Сумма всего подряд [0.8 sec, 256 mb]

Дан случайный граф. Нужно для каждого множества вершин A посчитать $f(A)$, количество независимых подмножеств вершин $B: B \subseteq A$. Множество вершин B называется независимым, если в графе нет ребра, оба конца которого лежат в множестве B .

Формат входных данных

На первой строке число вершин n ($1 \leq n \leq 23$) и число ребер $m \geq 1$.

Следующие m строк содержат пары чисел от 1 до n — ребра графа.

В графе нет ни петель, ни кратных ребер.

Формат выходных данных

Каждому множеству A можно сопоставить целое число $b(A)$, двоичная запись которого соответствует наличию элементов в множестве A . Пример: $n = 5, A = \{1, 2, 5\}, b(A) = 2^0 + 2^1 + 2^4 = 19$. Выведите $\sum_A f(A)2^{b(A)} \bmod (10^9 + 7)$

Примеры

subsetsum.in	subsetsum.out
3 1 1 2	1221

Пояснение к примеру

Независимыми являются множества вершин $\{\}, \{1\}, \{2\}, \{3\}, \{1, 3\}, \{2, 3\}$.

$$A = \{\} \quad f(A) = 1 \quad b(A) = 0$$

$$A = \{1\} \quad f(A) = 2 \quad b(A) = 2^0 = 1$$

$$A = \{2\} \quad f(A) = 2 \quad b(A) = 2^1 = 2$$

$$A = \{1, 2\} \quad f(A) = 3 \quad b(A) = 2^0 + 2^1 = 3$$

$$A = \{3\} \quad f(A) = 2 \quad b(A) = 2^2 = 4$$

$$A = \{1, 3\} \quad f(A) = 4 \quad b(A) = 2^0 + 2^2 = 5$$

$$A = \{2, 3\} \quad f(A) = 4 \quad b(A) = 2^1 + 2^2 = 6$$

$$A = \{1, 2, 3\} \quad f(A) = 6 \quad b(A) = 2^0 + 2^1 + 2^2 = 7$$

$$1 \cdot 2^0 + 2 \cdot 2^1 + 2 \cdot 2^2 + 3 \cdot 2^3 + 2 \cdot 2^4 + 4 \cdot 2^5 + 4 \cdot 2^6 + 6 \cdot 2^7 = 1221$$

Задача 9B. Симпатичные узоры [1 sec, 256 mb]

Компания *BrokenTiles* планирует заняться выкладыванием во дворах у состоятельных клиентов узор из черных и белых плиток, каждая из которых имеет размер 1×1 метр. Известно, что дворы всех состоятельных людей имеют наиболее модную на сегодня форму прямоугольника $n \times m$ метров.

Однако при составлении финансового плана у директора этой организации появилось целых две серьезных проблемы: во первых, каждый новый клиент очевидно захочет, чтобы узор, выложенный у него во дворе, отличался от узоров всех остальных клиентов этой фирмы, а во вторых, этот узор должен быть симпатичным.

Как показало исследование, узор является симпатичным, если в нем нигде не встречается квадрата 2×2 метра, полностью покрытого плитками одного цвета.

Для составления финансового плана директору необходимо узнать, сколько клиентов он сможет обслужить, прежде чем симпатичные узоры данного размера закончатся. Помогите ему!

Формат входных данных

На первой строке входного файла находятся два натуральных числа n и m . $1 \leq n \cdot m \leq 30$.

Формат выходных данных

Выведите в выходной файл единственное число — количество различных симпатичных узоров, которые можно выложить во дворе размера $n \times m$. Узоры, получающиеся друг из друга сдвигом, поворотом или отражением считаются различными.

Пример

nice.in	nice.out
2 2	14
3 3	322

Обязательные задачи

Задача 9С. Крышки [2 sec, 256 mb]

У программиста Петрова есть хобби — собирать крышки от пивных бутылок. Ничего странного, он знает еще с сотню программистов, которые очень уважают пиво. Да, они тоже собирают крышки. Не все из них, конечно, но некоторые. Если честно, то часть своих крышек он просто купил, уже без бутылок. Да, это не совсем спортивно, зато коллекция теперь более солидная. Одна вот беда — не хватает ему для полноты коллекции еще нескольких редких крышек. Он даже нашел в Интернете программистов, которые согласны продать их ему. Некоторые даже продают крышки сразу наборами, с большой скидкой. Почему продают, да еще со скидкой? А вы попробуйте объяснить жене, какая польза от пивных крышек. Она же не программист. Осталось выбрать оптимальные предложения. Если объяснить жене зачем надо хранить крышки еще возможно, то почему на них надо тратить деньги — точно не поймет. Поэтому надо купить как можно дешевле. Петров выписал на бумажку все варианты и задумался. Купить сразу все не получится, никаких денег не хватит. Тогда надо купить самые необходимые, но подешевле. Да уж, без программы тут не обойдешься...

Формат входных данных

В первой строке записано число N — количество недостающих Петрову крышек ($1 \leq N \leq 20$). Далее идет N строк — цена, за которую можно купить эту крышку, если покупать ее отдельно. В следующей строке стоит число M ($0 \leq M \leq 100$) — количество предложений по продаже наборов, содержащих нужные ему крышки. Далее идет M строк описывающих эти наборы. В каждой строке первое число — цена набора, второе — количество крышек в наборе, далее перечислены номера крышек (каждый номер от 1 до N), которые в этот набор входят. Номера в наборе не повторяются. Все цены — положительные числа, не превосходящие 2000. В последней строке перечислен минимальный набор крышек, который Петров намерен купить в любом случае.

Формат выходных данных

Выведите минимальную сумму, необходимую Петрову, чтобы купить все крышки из приведенного в последней строке набора.

Примеры

bottletaps.in	bottletaps.out
4	25
10	
11	
12	
13	
3	
17 2 1 3	
25 3 2 3 4	
15 2 3 4	
3 1 3 4	

Задача 9D. Быстрое пересечение множеств [1 сек, 256 mb]

Даны N множеств. Множества занумерованы целыми числами от 1 до N . Для каждого множества $i = 1..N$ нужно найти такое множество $j = 1..N, j \neq i$, что их непохожесть минимальна. Непохожестью двух множеств A и B называется количество элементов, присутствующих ровно в одном из множеств A и B .

Формат входных данных

На первой строке целое число N от 2 до 10^4 — количество множеств. Далее собственно множества. Каждое множество задается следующим образом: сперва целое число k от 0 до 32 — размер множества, далее k целых чисел от 0 до 31 — элементы множества. Все элементы множества различны.

Формат выходных данных

Выведите N строк, в i -й строке выведите номер j — номер множества, которое вы считаете наименее непохожим на i -е), и собственно “непохожесть” данных множеств. Если для некоторого i существует несколько оптимальных j , выведите любое.

Пример

intersectsets.in	intersectsets.out
6	2 2
6 1 2 3 4 5 6	3 0
4 1 2 3 4	2 0
4 1 2 3 4	1 2
6 0 1 2 3 4 5	6 3
4 31 30 29 28	5 3
3 1 30 31	

Замечание

Есть такая стандартная функция `__builtin_popcount(x)`.

Но код, написанный руками за $\mathcal{O}(1)$, работает быстрее.

Как посчитать количество бит в числе за $\mathcal{O}(1)$? Предподсчёт поможет.

Задача 9Е. Симпатичные узоры 2 [2 sec, 256 mb]

Компания *BrokenTiles* планирует заняться выкладыванием во дворах у состоятельных клиентов узор из черных и белых плиток, каждая из которых имеет размер 1×1 метр. Известно, что дворы всех состоятельных людей имеют наиболее модную на сегодня форму прямоугольника $n \times m$ метров.

Однако при составлении финансового плана у директора этой организации появилось целых две серьезных проблемы: во первых, каждый новый клиент очевидно захочет, чтобы узор, выложенный у него во дворе, отличался от узоров всех остальных клиентов этой фирмы, а во вторых, этот узор должен быть симпатичным.

Как показало исследование, узор является симпатичным, если в нем нигде не встречается квадрата 2×2 метра, полностью покрытого плитками одного цвета.

Для составления финансового плана директору необходимо узнать, сколько клиентов он сможет обслужить, прежде чем симпатичные узоры данного размера закончатся. Помогите ему!

Формат входных данных

На первой строке входного файла находятся два натуральных числа n и m . $1 \leq n \cdot m \leq 300$.

Формат выходных данных

Выведите в выходной файл единственное число — количество различных симпатичных узоров, которые можно выложить во дворе размера $n \times m$ по модулю $2^{30} + 1$. Узоры, получающиеся друг из друга сдвигом, поворотом или отражением считаются различными.

Пример

nice2.in	nice2.out
2 2	14
3 3	322

Задача 9F. Покрытие доминошками [2 sec, 256 mb]

Сколько способов покрыть прямоугольник $W \times H$ доминошками?
Каждая клетка должна быть покрыта ровно одной доминошкой.

Формат входных данных

Числа W и H . Ограничения: $W \cdot H \leq 100$, кроме этого $1 \leq W, H$.

Формат выходных данных

Число способов покрыть прямоугольник доминошками.

Пример

dominoes2.in	dominoes2.out
3 2	3

Дополнительные задачи

Задача 9G. Игра «Жизнь» [3 sec, 256 mb]

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

life.in	life.out
3 4	3
В.В	
.В.	
...	
...	

Пояснение к примеру

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

```
BBB BBB BBB
BBB BBB BBB
WWW BBB BBB
WWW WWW BBB
```

Замечание

Нерекурсивная реализация по скошенному профилю точно получает ок.

Задача 9Н. Симпатичные узоры 3 [2 sec, 256 mb]

Компания *BrokenTiles* планирует заняться выкладыванием во дворах у состоятельных клиентов узор из черных и белых плиток, каждая из которых имеет размер 1×1 метр. Известно, что дворы всех состоятельных людей имеют наиболее модную на сегодня форму прямоугольника $n \times m$ метров.

Однако при составлении финансового плана у директора этой организации появилось целых две серьезных проблемы: во первых, каждый новый клиент очевидно захочет, чтобы узор, выложенный у него во дворе, отличался от узоров всех остальных клиентов этой фирмы, а во вторых, этот узор должен быть симпатичным.

Как показало исследование, узор является симпатичным, если в нем нигде не встречается квадрата 2×2 метра, полностью покрытого плитками одного цвета.

Для составления финансового плана директору необходимо узнать, сколько клиентов он сможет обслужить, прежде чем симпатичные узоры данного размера закончатся. Помогите ему!

Формат входных данных

На первой строке входного файла находятся три натуральных числа n , m , mod . ($1 \leq n \leq 10^{100}$, $1 \leq m \leq 6$, $1 \leq mod \leq 10\,000$).

n, m — размеры двора. mod — модуль, по которому нужно посчитать ответ.

Формат выходных данных

Выведите в выходной файл единственное число — количество различных симпатичных узоров, которые можно выложить во дворе размера $n \times m$ по модулю mod . Узоры, получающиеся друг из друга сдвигом, поворотом или отражением считаются различными.

Пример

nice3.in	nice3.out
2 2 5	4
3 3 23	0

Задача 91. Кони на гриде [2 сек, 256 mb]

Вопрос: сколькими способами можно расставить шахматных коней на доске $w \times h$ так, чтобы они не били друг друга?

Ответ требуется посчитать по модулю $10^9 + 7$.

Напомним, что шахматный конь ходит «буквой Г». Т.е. из клетки (x, y) он бьет следующие 8 клеток: $(x - 1, y - 2)$, $(x - 2, y - 1)$, $(x + 1, y - 2)$, $(x + 2, y - 1)$, $(x - 1, y + 2)$, $(x - 2, y + 1)$, $(x + 1, y + 2)$, $(x + 2, y + 1)$.

$w \leq 9, h \leq 100$

Формат входных данных

Два целых числа — w и h ($1 \leq w, h$).

Формат выходных данных

Одно целое число — количество способов по модулю $10^9 + 7$.

Примеры

knight.in		knight.out	
1	1	2	
2	3	36	

Задача 9J. Длинные домино [1 сек, 256 mb]

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.

Пример

dominoes.in	dominoes.out
3 3	2
3 10	28