

Задача А «Вырубка деревьев» (В)

| | |
|---|--------------|
| Имя входного файла: | trees.in |
| Имя выходного файла: | trees.out |
| Максимальное время работы на одном тесте: | 1 секунда |
| Максимальный объем используемой памяти: | 64 Мегабайта |

Король Флатландии решил вырубить некоторые деревья, растущие перед его дворцом. Деревья перед дворцом короля посажены в ряд, всего там растет n деревьев, расстояния между соседними деревьями одинаковы.

После вырубки перед дворцом должно остаться m деревьев, и расстояния между соседними деревьями должны быть одинаковыми. Помогите королю выяснить, сколько существует способов вырубки деревьев.

Требуется написать программу, которая по заданным числам n и m определит, сколько существует способов вырубки некоторых из n деревьев так, чтобы после вырубки осталось m деревьев и соседние деревья находились на равном расстоянии друг от друга.

Формат входных данных

Входной файл содержит два целых числа n и m ($0 \leq m \leq n \leq 1000$).

Формат выходных данных

Выведите в выходной файл одно число — искомое количество способов.

Примеры входного и выходного файлов

| trees.in | trees.out |
|----------|-----------|
| 5 3 | 4 |

Пояснение к примерам

Если обозначить условно исходное расположение деревьев перед дворцом как «ТТТТТ», то возможные результаты после вырубки следующие:

«ТТТ. .», «.ТТТ.», «. .ТТТ», «Т.Т.Т».

Задача В «Перепутанные диски» (Е)

| | |
|---|--------------|
| Имя входного файла: | discs.in |
| Имя выходного файла: | discs.out |
| Максимальное время работы на одном тесте: | 1 секунда |
| Максимальный объем используемой памяти: | 64 Мегабайта |

Вася — страстный любитель компьютерных игр. Его коллекция насчитывает десятки компакт-дисков с играми. Однако он очень неаккуратный мальчик. Коробки с дисками в полном беспорядке раскиданы по его столу, и поэтому найти что-либо на столе практически невозможно.

Когда Вася хочет поиграть в очередную игру, он действует следующим образом: берет произвольную коробку с диском со стола и вставляет диск из этой коробки в CD-привод своего компьютера. Если в CD-приводе уже есть какой-нибудь диск, то вместо того, чтобы найти коробку от этого диска и убрать его туда, Вася убирает диск в коробку, из которой он только что достал очередной диск.

Например, пусть у Васи есть три компакт-диска с играми — «Цивилизация», «Тетрис» и «Сапер». Пусть Вася сначала начал играть в «Цивилизацию», а затем решил поиграть в «Тетрис». Тогда после этого диск с «Цивилизацией» окажется в коробке от «Тетриса». Пусть затем он решил поиграть в «Сапера». Тогда диск от «Тетриса» окажется в коробке от «Сапера». Если после этого он снова решит поиграть в «Цивилизацию» (заметим, что для этого он достанет ее из коробки от «Тетриса»), то игра «Сапер» окажется в коробке от «Тетриса», а «Цивилизация» — в CD-приводе Васиного компьютера.

Предполагая, что исходно все диски с играми находятся в своих коробках, напишите программу, которая по заданной последовательности игр, в которые играл Вася, определит, в какой коробке окажется после этого каждый из дисков с играми.

Формат входных данных

Первая строка входного файла содержит число n — количество игр, в которые играл Вася ($1 \leq n \leq 1000$), при этом Вася мог играть в одну и ту же игру несколько раз. Следующие n строк содержат названия игр в том порядке, в котором играл Вася. Все названия состоят из латинских букв, цифр и пробелов, длина названия не превышает 50 символов.

Формат выходных данных

Выведите в выходной файл k строк, где k — количество различных игр, в которые играл Вася. Каждая строка должна иметь вид «<game> - <box>», где <game> — название игры, а <box> — название игры, в коробке от которой лежит игра <game>. Если соответствующая игра лежит в CD-приводе компьютера, вместо <box> выведите «*» (звездочку). Выводите игры в произвольном порядке

Примеры входного и выходного файлов

| discs.in | discs.out |
|--------------|----------------------|
| 4 | Civilization - * |
| Civilization | Tetris - Minesweeper |
| Tetris | Minesweeper - Tetris |
| Minesweeper | |
| Civilization | |

Задача С «Разрезание торта»

| | |
|---|--------------|
| Имя входного файла: | cut.in |
| Имя выходного файла: | cut.out |
| Максимальное время работы на одном тесте: | 1 секунда |
| Максимальный объем используемой памяти: | 64 Мегабайта |

Мама испекла Мише на день рождения торт. Торт имеет форму выпуклого многоугольника с n вершинами. Вместе с гостями и родственниками у Миши на празднике оказалось k человек. В нужный момент Миша планирует разрезать торт на k частей. Каждый разрез должен представлять собой диагональ исходного многоугольника. Чтобы торт не развалился, разрезы не должны пересекаться нигде, кроме как в вершинах торта.

Как юного математика, Мишу заинтересовал вопрос — сколько существует способов разрезания торта на k частей указанным выше способом. Порядок выполнения разрезов не важен. Помогите Мише найти ответ на интересующий его вопрос.

Требуется написать программу, которая по заданным значениям чисел n и k определяет количество способов разрезания на k частей торта с n вершинами, как указано выше.

Формат входных данных

Входной файл содержит два целых числа n и k ($1 \leq k \leq n \leq 50$, $n \geq 3$).

Формат выходных данных

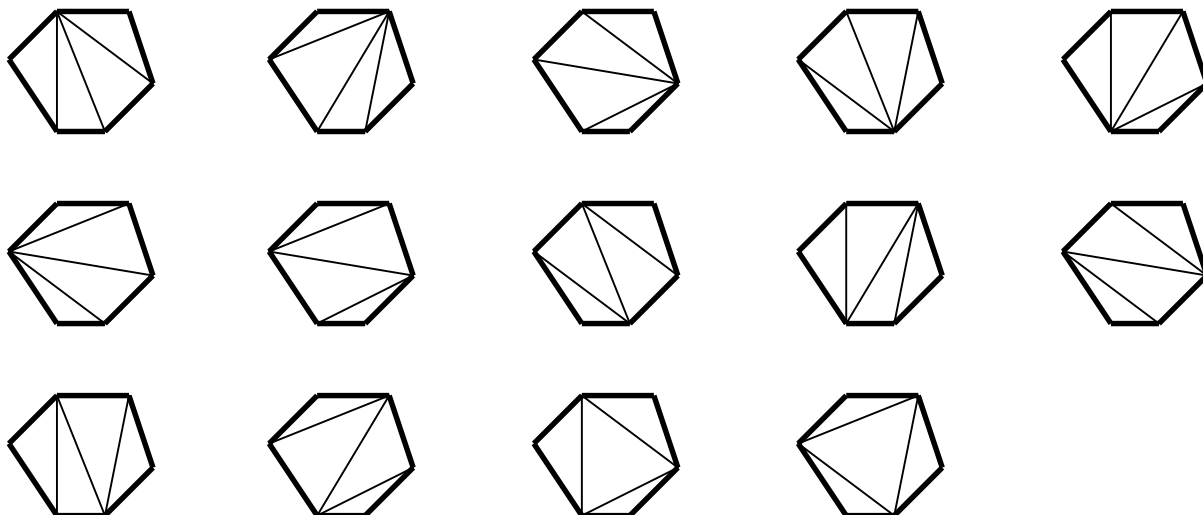
Выведите в выходной файл одно число — искомое количество способов разрезания торта.

Примеры входных и выходных файлов

| cut.in | cut.out |
|--------|---------|
| 4 2 | 2 |
| 6 4 | 14 |
| 3 2 | 0 |

Пояснение к примерам

Все способы разрезания торта с шестью вершинами на четыре части приведены на следующем рисунке.



Задача D «Алгоритм Евклида»

| | |
|---|--------------|
| Имя входного файла: | euclid.in |
| Имя выходного файла: | euclid.out |
| Максимальное время работы на одном тесте: | 2 секунды |
| Максимальный объем используемой памяти: | 64 мегабайта |

Дима недавно начал изучать информатику. Одним из первых алгоритмов, который он изучил, был алгоритм Евклида для нахождения наибольшего общего делителя (НОД) двух чисел. Напомним, что наибольшим общим делителем двух чисел a и b называется наибольшее натуральное число x , такое, что и число a , и число b делится на него без остатка.

Алгоритм Евклида заключается в следующем:

1. Пусть a, b — числа, НОД которых надо найти.
2. Если $b = 0$, то число a — искомым НОД.
3. Если $b > a$, то необходимо поменять местами числа a и b .
4. Присвоить числу a значение $a - b$.
5. Вернуться к шагу 2.

Дима достаточно быстро освоил алгоритм Евклида и вычислил с его помощью много наибольших общих делителей. Поняв, что надо дальше совершенствоваться, ему пришла идея решить новую задачу. Пусть заданы числа a, b, c и d . Требуется узнать, наступит ли в процессе реализации алгоритма Евклида для заданной пары чисел (a, b) такой момент, когда перед исполнением шага 2 число a будет равно c , а число b будет равно d .

Требуется написать программу, которая решает эту задачу.

Формат входных данных

Первая строка входного файла содержит количество наборов входных данных k ($1 \leq k \leq 100$). Далее идут описания этих наборов. Каждое описание состоит из двух строк. Первая из них содержит два целых числа: a, b ($1 \leq a, b \leq 10^{18}$). Вторая строка – два целых числа: c, d ($1 \leq c, d \leq 10^{18}$).

Все числа в строках разделены пробелом.

Формат выходных данных

Для каждого набора входных данных выведите в отдельной строке выходного файла слово «YES», если в процессе применения алгоритма Евклида к паре чисел (a, b) в какой-то момент получается пара (c, d) , или слово «NO» – в противном случае.

Примеры входных и выходных файлов

| euclid.in | euclid.out |
|-----------|------------|
| 2 | YES |
| 20 10 | NO |
| 10 10 | |
| 10 7 | |
| 2 4 | |

Задача Е «Скобочки»

| | |
|---|--------------|
| Имя входного файла: | brackets.in |
| Имя выходного файла: | brackets.out |
| Максимальное время работы на одном тесте: | 2 секунды |
| Максимальный объем используемой памяти: | 64 мегабайта |

Строка, состоящая из символов «(» и «)», называется *скобочной последовательностью*. Скобочная последовательность называется *правильной*, если она может быть получена из некоторого корректного арифметического выражения удалением всех символов, кроме скобок. Например, правильная скобочная последовательность «(()) ()» может быть получена из выражения «(2 - (3 + 4) * 6) * (1 + 1)».

Другое определение правильной скобочной последовательности можно дать следующим образом:

1. Пустая строка является правильной скобочной последовательностью.
2. Если S — правильная скобочная последовательность, то (S) — тоже правильная скобочная последовательность.
3. Если S и T — правильные скобочные последовательности, то ST — правильная скобочная последовательность.

Глубиной правильной скобочной последовательности называется максимальная разность между количеством открывающихся и закрывающихся скобок в префиксе последовательности (префиксом строки S называется строка, которую можно получить из S удалением некоторого количества последних символов, например, префиксами строки «ABCAB» являются строки «», «A», «AB», «ABC», «ABCA» и «ABCAB»). Например, глубина последовательности «() (())» равна двум (префикс «() ((» имеет 4 открывающиеся и 2 закрывающиеся скобки).

Требуется написать программу, определяющую по заданным значениям n и k количество правильных скобочных последовательностей с n открывающимися скобками, которые имеют глубину, равную k .

Формат входных данных

Входной файл содержит в одной строке целые числа n и k ($1 \leq k \leq n \leq 50$), разделенные пробелом.

Формат выходных данных

Выходной файл должен содержать одно число — количество правильных скобочных последовательностей с n открывающимися скобками, которые имеют глубину k .

Примеры входных и выходных файлов

| brackets.in | brackets.out |
|-------------|-----------------|
| 3 2 | 3 |
| 37 23 | 203685956218528 |

Задача F «Строки в книге»

Имя входного файла: a.in
Имя выходного файла: a.out
Максимальное время работы на одном тесте: 3 секунды
Максимальный объем используемой памяти: 64 мегабайта

В книге на одной странице помещается K строк. Таким образом, на 1-й странице печатаются строки с 1-й по K -ю, на второй — с $(K+1)$ -й по $(2 \cdot K)$ -ю и т.д. Напишите программу, которая по номеру строки в тексте определяет номер страницы, на которой будет напечатана эта строка, и порядковый номер этой строки на странице.

Формат входных данных

Входной файл содержит число K — количество строк, которое печатается на странице, и число N — номер строки ($1 \leq K \leq 200$, $1 \leq N \leq 20000$).

Формат выходных данных

В выходной файл выведите два числа — номер страницы, на которой будет напечатана эта строка и номер строки на странице

Примеры

| a.in | a.out |
|-------|-------|
| 50 1 | 1 1 |
| 20 25 | 2 5 |
| 15 43 | 3 13 |

Задача G Оттепель

Максимальное время работы на одном тесте: 1 секунда
 Максимальный объем используемой памяти: 64 мегабайт
 Имя входного файла: bin.in
 Имя выходного файла: bin.out

Уставшие от необычно теплой зимы, парижане решили узнать, действительно ли это самая длинная оттепель за всю историю наблюдений за погодой. Они обратились к синоптикам, а те, в свою очередь, занялись исследованиями статистики за прошлые годы. Их интересует, сколько дней длилась самая длинная оттепель.

Оттепелью они называют период, в который среднесуточная температура ежедневно превышала 0 градусов Цельсия. Напишите программу, помогающую синоптикам в работе.

Формат входных данных

Сначала вводится число N – общее количество рассматриваемых дней ($1 \leq N \leq 100$). В следующей строке вводится N целых чисел, разделенных пробелами. Каждое число – среднесуточная температура в соответствующий день. Температуры – целые числа и лежат в диапазоне от -50 до 50 .

Формат выходных данных

Требуется вывести одно число – длину самой продолжительной оттепели, то есть наибольшее количество последовательных дней, на протяжении которых среднесуточная температура превышала 0 градусов. Если температура в каждый из дней была неположительной, выведите 0.

Примеры

| Входные данные | Выходные данные | Комментарий |
|---------------------------|-----------------|---|
| 6 -20 30 -40 50 10 -10 | 2 | Рассматриваются 6 дней. Самая продолжительная оттепель была на 4-й и 5-й день (50 и 10 градусов соответственно) |
| 8 10 20 30 1 -10 1 2 3 | 4 | Самая продолжительная оттепель была в первые 4 дня |
| 5 -10 0 -10 0 -10 | 0 | Дней с положительной температурой не было |

Задача Н «Шахматы»

| | |
|---|--------------|
| Имя входного файла: | chess.in |
| Имя выходного файла: | chess.out |
| Максимальное время работы на одном тесте: | 1 секунда |
| Максимальный объем используемой памяти: | 64 Мегабайта |

На шахматной доске стоит несколько офицеров и ладей. Требуется посчитать количество свободных клеток, которые не находятся под боем ни одной из фигур.

Замечание для тех, кто не умеет играть в шахматы:

Шахматная доска имеет размеры 8*8. Ладья бьет все клетки горизонтали и вертикали, проходящих через клетку, где она стоит, до первой встретившейся фигуры. Офицер бьет все клетки обеих диагоналей, проходящих через клетку, где он стоит, до первой встретившейся фигуры.

Формат входных данных

В первых восьми строках входного файла описывается шахматная доска. Первые восемь символов каждой из этих строк описывают состояние соответствующей горизонтали: символ В (заглавная латинская буква) означает, что в клетке стоит офицер, символ R — ладья, символ * — что клетка пуста. После описания горизонтали в строке могут идти пробелы, однако длина каждой строки не превышает 250 символов. После описания доски в файле могут быть пустые строки.

Формат выходных данных

В выходной файл выведите количество пустых клеток, которые не бьются ни одной из фигур.

Примеры входного и выходного файлов

| chess.in | chess.out | chess.in | chess.out | chess.in | chess.out |
|---|-----------|---|-----------|--|-----------|
| ***** *RB***** ***** ***** ***** ***** ***** ***** | 47 | RRRRRRRR BBBBBBBB RRRRRRRR BBBBBBBB RRRRRRRR BBBBBBBB RRRRRRRR BBBBBBBB* | 0 | ***** ***** ***** ***** ***** ***** ***** ***** | 64 |