

64 megabytes

Задача А. Компьютерная игра (платформы)

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 0.5 second
Ограничение по памяти: 64 мегабайта

Вы можете вспомнить хоть одного своего знакомого до двадцатилетнего возраста, который в детстве не играл в компьютерные игры? Если да, то может быть вы и сами не знакомы с этим развлечением? Впрочем, трудностей при решении этой задачи это создать не должно.

Во многих старых играх с двумерной графикой можно столкнуться с подобной ситуацией. Какой-нибудь герой прыгает по платформам (или островкам), которые висят в воздухе. Он должен перебраться от одного края экрана до другого. При этом при прыжке с одной платформы на соседнюю, у героя уходит $|y_2 - y_1|$ единиц энергии, где y_1 и y_2 — высоты, на которых расположены эти платформы. Кроме того, у героя есть суперприём, который позволяет перескочить через платформу, но на это затрачивается $3 \cdot |y_3 - y_1|$ единиц энергии. Конечно же, энергию следует расходовать максимально экономно.

Предположим, что вам известны координаты всех платформ в порядке от левого края до правого. Сможете ли вы найти, какое минимальное количество энергии потребуется герою, чтобы добраться с первой платформы до последней?

Формат входного файла

В первой строке записано количество платформ n ($1 \leq n \leq 30000$). Вторая строка содержит n натуральных чисел, не превосходящих 30000 — высоты, на которых располагаются платформы.

Формат выходного файла

Выведите единственное число — минимальное количество энергии, которую должен потратить игрок на преодоление платформ (конечно же в предположении, что cheat-коды использовать нельзя).

Примеры

stdin	stdout
3 1 5 10	9
3 1 5 2	3

64 мегабайта

Задача В. Платформы — дополнительный критерий оптимизации

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Во многих старых играх с двумерной графикой можно столкнуться с подобной ситуацией. Какой-нибудь герой прыгает по платформам (или островкам), которые висят в воздухе. Он должен перебраться от одного края экрана до другого. При этом при прыжке с одной платформы на соседнюю, у героя уходит $|y_2 - y_1|$ единиц энергии, где y_1 и y_2 — высоты, на которых расположены эти платформы. Кроме того, у героя есть суперприём, который позволяет перескочить через платформу, но на это затрачивается $3 \cdot |y_3 - y_1|$ единиц энергии. Конечно же, энергию следует расходовать максимально экономно. Но, по возможности, хотелось бы уменьшить также и количество прыжков.

Предположим, что вам известны координаты всех платформ в порядке от левого края до правого. Во-первых, найдите минимальное количество энергии, которое потребуется герою, чтобы добраться с первой платформы до последней. Если существуют разные способы добраться с первой платформы до последней с *одинаковыми минимальными* затратами энергии, то среди них выберите способ с минимальным количеством прыжков.

Формат входного файла

Первая строка содержит количество платформ N ($2 \leq N \leq 100000$), вторая — N целых чисел, значения которых не превышают по модулю 4000 — высоты платформ.

Формат выходного файла

Выведите в одной строке два числа через пробел: сначала минимальное количество энергии, потом минимальное (среди тех, при которых возможно минимальное количество энергии) количество прыжков.

Примеры

stdin	stdout
4 1 2 3 30	29 3
5 1 1 1 1 1	0 2

Примечание

Второй ответ первого примера всё-таки 3 (а не 2), т. к. просят искать не вообще минимальное количество прыжков, а минимальное среди тех способов, при которых достигается минимальное количество затраченной энергии. В данном конкретном случае минимальное количество затраченной энергии (29) достигается только одним способом, и в нём 3 прыжка.

64 megabytes

Задача С. Компьютерная игра (платформы) с восстановлением пути

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 second
Ограничение по памяти: 64 мегабайта

В старых играх можно столкнуться с такой ситуацией. Герой прыгает по платформам, висящим в воздухе. Он должен перебраться от одного края экрана до другого. При прыжке с платформы на соседнюю, у героя уходит $|y_2 - y_1|$ энергии, где y_1 и y_2 — высоты, на которых расположены эти платформы. Кроме того, есть суперприём, позволяющий перескочить через платформу, но на это затрачивается $3 \cdot |y_3 - y_1|$ энергии.

Известны высоты платформ в порядке от левого края до правого. Найдите минимальное количество энергии, достаточное, чтобы добраться с 1-й платформы до n -й (последней) и список (последовательность) платформ, по которым нужно пройти.

Формат входного файла

Первая строка содержит количество платформ N ($2 \leq N \leq 100000$), вторая — N целых чисел, значения которых не превышают по модулю 4000 — высоты платформ.

Формат выходного файла

В первой строке выведите минимальное количество энергии. Во второй — количество платформ, по которым нужно пройти, а в третьей выведите последовательность этих платформ.

Примеры

stdin	stdout
4 1 2 3 30	29 4 1 2 3 4
10 1 100 1 100 1 100 1 100 1 100	99 6 1 2 4 6 8 10

Примечание

Разумеется, в этой задаче для некоторых входных данных возможны разные правильные последовательности платформ. Ваша программа должна выводить любую одну из них.

64 мегабайта

Задача D. Компьютерная игра (платформы) — квадратичные прыжки

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Вы можете вспомнить хоть одного своего знакомого до двадцатилетнего возраста, который в детстве не играл в компьютерные игры? Если да, то может быть вы и сами не знакомы с этим развлечением? Впрочем, трудностей при решении этой задачи это создать не должно.

Во многих старых играх с двумерной графикой можно столкнуться с подобной ситуацией. Какой-нибудь герой прыгает по платформам (или островкам), которые висят в воздухе. Он должен перебраться от одного края экрана до другого. При этом при прыжке с одной платформы на соседнюю, у героя уходит $|y_2 - y_1|^2$ единиц энергии, где y_1 и y_2 — высоты, на которых расположены эти платформы. Кроме того, у героя есть суперприём, который позволяет перескочить через платформу, но на это затрачивается $3 \cdot |y_3 - y_1|^2$ единиц энергии. Конечно же, энергию следует расходовать максимально экономно.

Предположим, что вам известны координаты всех платформ в порядке от левого края до правого. Сможете ли вы найти, какое минимальное количество энергии потребуется герою, чтобы добраться с первой платформы до последней?

Формат входного файла

Первая строка содержит количество платформ N ($2 \leq N \leq 100000$), вторая — N целых чисел, значения которых не превышают по модулю 4000 — высоты платформ.

Формат выходного файла

Выведите единственное число — минимальное количество энергии, которую должен потратить игрок на преодоление платформ (конечно же в предположении, что cheat-коды использовать нельзя).

Примеры

stdin	stdout
4 1 2 3 30	731

256 мегабайт

Задача Е. Математические платформы

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Вы можете вспомнить хоть одного своего знакомого до двадцатилетнего возраста, который в детстве не играл в компьютерные игры? Если да, то может быть вы и сами не знакомы с этим развлечением? Впрочем, трудностей при решении этой задачи это создать не должно.

Во многих старых играх с двумерной графикой можно столкнуться с подобной ситуацией. Какой-нибудь герой прыгает по платформам (или островкам), которые висят в воздухе. Он должен перебраться от одного края экрана до другого. Игрок может прыгнуть с любой платформы i на любую платформу k , затратив при этом $(i - k)^2 \cdot (y_i - y_k)^2$ единиц энергии, где y_i и y_k — высоты, на которых расположены эти платформы. Конечно же, энергию следует расходовать максимально экономно.

Предположим, что вам известны координаты всех платформ в порядке от левого края до правого. Сможете ли вы найти, какое минимальное количество энергии потребуется герою, чтобы добраться с первой платформы до последней?

Формат входного файла

В первой строке записано количество платформ n ($1 \leq n \leq 4000$). Вторая строка содержит n целых чисел, не превосходящих по модулю 200000 — высоты, на которых располагаются платформы.

Формат выходного файла

Выведите единственное число — минимальное количество энергии, которую должен потратить игрок на преодоление платформ (конечно же в предположении, что cheat-коды использовать нельзя).

Примеры

<code>stdin</code>	<code>stdout</code>
4 1 2 3 30	731

64 мегабайта

Задача F. Путь через горы (малые ограничения)

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Поверхность Земли в горной местности можно представить в виде ломаной линии. Вершины ломаной расположены в точках (x_1, y_1) , (x_2, y_2) , \dots , (x_N, y_N) , при этом $x_i < x_{i+1}$. Обычный горный маг находится в точке (x_1, y_1) и очень хочет попасть в точку (x_N, y_N) . При этом он может перемещаться только пешком. Он может ходить по поверхности Земли (т. е. вдоль ломаной). А может сотворить в воздухе мост и пройти по нему. Мост может соединять две вершины ломаной: мост не может начинаться и заканчиваться не в вершине ломаной, и мост не может проходить под землей (в т. ч. не может быть туннелем в горе), но мост может каким-то своим участком проходить по поверхности земли. Длина моста не может быть больше R . Суммарно маг может построить не более K мостов. После прохождения моста, он (мост) растворяется в воздухе. Какое наименьшее расстояние придётся пройти магу, чтобы оказаться в точке (x_N, y_N) ?

Формат входного файла

Программа должна прочитать сначала натуральное число N ($2 \leq N \leq 42$); затем натуральное число K ($1 \leq K \leq 23$) — максимальное количество мостов; далее целое число R ($0 \leq R \leq 10000$) — максимальную возможную длину моста. Далее координаты (x_1, y_1) , (x_2, y_2) , \dots , (x_N, y_N) . Все координаты — целые числа, не превышающие по модулю 10000, для всех i от 1 до $N-1$ выполняется $x_i < x_{i+1}$.

Формат выходного файла

Программа должна вывести одно число — минимальную длину пути, которую придётся пройти магу (как по земле, так и по мостам). Ответ выведите с точностью 6 цифр после десятичной точки.

Примеры

stdin	stdout
5 2 5 0 0 2 2 3 -1 4 1 5 0	6.4787086646190746

64 мегабайта

Задача G. Путь через горы

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Поверхность Земли в горной местности можно представить в виде ломаной линии. Вершины ломаной расположены в точках $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$, при этом $x_i < x_{i+1}$. Обычный горный маг находится в точке (x_1, y_1) и очень хочет попасть в точку (x_N, y_N) . При этом он может перемещаться только пешком. Он может ходить по поверхности Земли (т. е. вдоль ломаной). А может сотворить в воздухе мост и пройти по нему. Мост может соединять две вершины ломаной: мост не может начинаться и заканчиваться не в вершине ломаной, и мост не может проходить под землей (в т. ч. не может быть туннелем в горе), но мост может каким-то своим участком проходить по поверхности земли. Длина моста не может быть больше R . Суммарно маг может построить не более K мостов. После прохождения моста, он (мост) растворяется в воздухе. Какое наименьшее расстояние придётся пройти магу, чтобы оказаться в точке (x_N, y_N) ?

Формат входного файла

Программа должна прочитать сначала натуральное число N ($2 \leq N \leq 555$); затем натуральное число K ($1 \leq K \leq 256$) — максимальное количество мостов; далее целое число R ($0 \leq R \leq 10000$) — максимальную возможную длину моста. Далее координаты $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$. Все координаты — целые числа, не превышающие по модулю 10000, для всех i от 1 до $N-1$ выполняется $x_i < x_{i+1}$.

Формат выходного файла

Программа должна вывести одно число — минимальную длину пути, которую придётся пройти магу (как по земле, так и по мостам). Ответ выведите с точностью 6 цифр после десятичной точки.

Примеры

stdin	stdout
5 2 5 0 0 2 2 3 -1 4 1 5 0	6.4787086646190746

256 мегабайт
Задача Н. Шахматы

Имя входного файла: **stdin**
Имя выходного файла: **stdout**
Ограничение по времени: **3 секунды**
Ограничение по памяти: **64 мегабайта**

Напомним некоторые существенные для данной задачи стандартные правила игры в шахматы. Играют два игрока, один играет белыми фигурами, другой чёрными. Игра происходит на доске 8×8 , столбики обозначаются буквами от “a” до “h” слева направо, строки — цифрами от 1 до 8 снизу вверх. Каждая клетка доски или пустая, или содержит одну фигуру. Если фигура A (не пешка) может походить согласно правилам на клетку, занятую чужой фигурой B , то вследствие такого хода фигуру B бьют, т. е. снимают с доски. Поэтому обо всех клетках, куда некоторая фигура может походить, говорят, что они находятся «под боем» данной фигуры. Королю запрещено ходить в клетку, которая находится под боем любой чужой фигуры. Если один из игроков сделал такой ход, что король противника оказался под боем (это называют «шах»), противник обязан ответить таким ходом, чтобы его король уже не был под боем. Если такого хода не существует, то это называют «мат».

Король может ходить на одну клетку в любом из 8-ми направлений (влево, вправо, вперед, назад, в любом направлении по любой диагонали). Ферзь может ходить в любом из 8-и направлений на любое количество клеток, но не пересекая клеток, занятых фигурами.

Пусть на шахматной доске находится три фигуры: белый король, белый ферзь и чёрный король. Сейчас ход белых. За какое минимальное количество ходов они гарантированно смогут поставить мат? Чёрные будут делать всё, допустимое правилами игры, чтобы избежать мата.

Формат входного файла

Программа должна прочитать число ($TEST_NUM \leq 70000$) — количество тестовых блоков, потом сами блоки. Каждый блок является отдельной строкой, в которой записаны обозначения трёх клеток, где находятся белый король, белый ферзь и чёрный король. Обозначение клетки состоит из записанных слитно буквы вертикали и номера горизонтали, обозначения клеток внутри строки разделены единичным пробелом.

Все заданные позиции гарантированно допустимы с точки зрения шахматных правил (например, чёрный король не под боем).

Формат выходного файла

Ваша программа должна вывести для каждого теста единственное число — минимальное количество ходов.

Примеры

stdin	stdout
2	1
a3 b3 a1	2
a3 e3 b1	

64 мегабайта

Задача I. MaxSum (базовая)

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Есть прямоугольная таблица размером N строк на M столбиков. В каждой клетке записано целое число. По ней нужно пройти сверху вниз, начиная из любой клетки верхней строки, дальше каждый раз переходя в одну из “нижних соседних” клеток (иными словами, из клетки с номером (i, j) можно перейти или на $(i+1, j-1)$, или на $(i+1, j)$, или на $(i+1, j+1)$; в случае $j = M$ последний из трёх описанных вариантов становится невозможным, а в случае $j = 1$ — первый) и закончить маршрут в какой-нибудь клетке нижней строки.

Напишите программу, которая будет находить максимально возможную сумму значений пройденных клеток среди всех допустимых путей.

Формат входного файла

В первой строке записаны N и M — количество строчек и количество столбиков ($1 \leq N, M \leq 200$), дальше в каждой из следующих N строк записано ровно по M разделённых пробелами целых чисел (каждое не превышает по модулю 10^6) — значения клеток таблицы.

Формат выходного файла

Вывести единственное целое число — найденную максимальную сумму.

Примеры

stdin	stdout
4 3 1 15 2 9 7 5 9 2 4 6 9 -1	42

64 мегабайта

Задача J. MaxSum (с количеством путей)

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Есть прямоугольная таблица размером N строк на M столбиков. В каждой клетке записано целое число. По ней нужно пройти сверху вниз, начиная из любой клетки верхней строки, дальше каждый раз переходя в одну из “нижних соседних” клеток (иными словами, из клетки с номером (i, j) можно перейти или на $(i+1, j-1)$, или на $(i+1, j)$, или на $(i+1, j+1)$; в случае $j = M$ последний из трёх описанных вариантов становится невозможным, а в случае $j = 1$ — первый) и закончить маршрут в какой-нибудь клетке нижней строки.

Напишите программу, которая будет находить максимально возможную сумму значений пройденных клеток среди всех допустимых путей и количество путей, на которых эта сумма достигается.

Формат входного файла

В первой строке записаны N и M — количество строчек и количество столбиков ($1 \leq N, M \leq 200$), дальше в каждой из следующих N строк записано ровно по M разделённых пробелами целых чисел (каждое не превышает по модулю 10^6) — значения клеток таблицы.

Гарантировано, что при проверке будут использованы только такие входные данные, для которых искомое количество путей с максимальной суммой не превышает 10^9 .

Формат выходного файла

Вывести два целых числа, разделенные пробелом — максимальную сумму и количество путей.

Примеры

stdin	stdout
4 3 1 15 2 9 7 5 9 2 4 6 9 -1	42 1
3 3 1 1 100 1 1 10 10 1 1	111 3

Примечание

В первом тесте, максимальное значение 42 можно набрать лишь по одному пути ($15 + 9 + 9 + 9$). А во втором, максимальное значение 111 можно набрать тремя способами: либо $a[1][3]=100$, $a[2][2]=1$, $a[3][1]=10$, либо $a[1][3]=100$, $a[2][3]=10$, $a[3][2]=1$, либо $a[1][3]=100$, $a[2][3]=10$, $a[3][3]=1$.

64 мегабайта

Задача К. MaxSum (прыжки в любой столбик)

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Есть прямоугольная таблица размером N строк на M столбиков. В каждой клетке записано целое число. По ней нужно пройти сверху вниз, начиная из любой клетки верхней строки, дальше каждый раз перепрыгивая в *любую* клетку следующего ряда (иными словами, из клетки с номером (i, j) можно перепрыгнуть в любую из клеток $(i + 1, 1), (i + 1, 2), \dots (i + 1, M)$) и закончить маршрут в какой-нибудь клетке нижней строки.

Напишите программу, которая будет находить максимально возможную сумму значений пройденных клеток среди всех допустимых путей.

Формат входного файла

В первой строке записаны N и M — количество строчек и количество столбиков ($1 \leq N, M \leq 200$), дальше в каждой из следующих N строк записано ровно по M разделённых пробелами целых чисел (каждое не превышает по модулю 10^6) — значения клеток таблицы.

Формат выходного файла

Вывести единственное целое число — найденную максимальную сумму.

Примеры

stdin	stdout
4 3 1 15 2 9 7 5 9 2 4 6 9 -1	42

64 мегабайта

Задача L. MaxSum (нечётная сумма)

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Есть прямоугольная таблица размером N строк на M столбиков. В каждой клетке записано целое число. По ней нужно пройти сверху вниз, начиная из любой клетки верхней строки, дальше каждый раз переходя в одну из “нижних соседних” клеток (иными словами, из клетки с номером (i, j) можно перейти или на $(i+1, j-1)$, или на $(i+1, j)$, или на $(i+1, j+1)$; в случае $j = M$ последний из трёх описанных вариантов становится невозможным, а в случае $j = 1$ — первый) и закончить маршрут в какой-нибудь клетке нижней строки.

Напишите программу, которая будет находить максимально возможную *нечётную сумму* значений пройденных клеток среди всех допустимых путей. Обратите внимание, что нечётной должна быть именно сумма; никаких ограничений на чётность/нечётность отдельных слагаемых нет.

Формат входного файла

В первой строке записаны N и M — количество строчек и количество столбиков ($1 \leq N, M \leq 200$), дальше в каждой из следующих N строк записано ровно по M разделённых пробелами целых чисел (каждое не превышает по модулю 10^6) — значения клеток таблицы.

Формат выходного файла

Вывести либо единственное целое число (найденную максимальную среди нечётных сумм), либо строку “impossible” (без кавычек, маленькими латинскими буквами). Строка “impossible” должна выводиться только в случае, когда абсолютно все маршруты указанного вида имеют чётные суммы.

Примеры

stdin	stdout
4 3 1 15 2 9 7 5 9 2 4 6 9 -1	39

Примечание

Вообще-то максимально возможная сумма — $42 = 15 + 9 + 9 + 9$, но число 42 чётное. Поэтому ответом будет максимальная среди нечётных сумм $39 = 15 + 9 + 9 + 6$, которая достигается по маршруту $a[1][2], a[2][1], a[3][1], a[4][1]$.

64 мегабайта

Задача M. MaxSum (посетить все столбики)

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Есть прямоугольная таблица размером N строк на M столбиков. В каждой клетке записано целое число. По ней нужно пройти сверху вниз, начиная из любой клетки верхней строки, дальше каждый раз переходя в одну из “нижних соседних” клеток (иными словами, из клетки с номером (i, j) можно перейти или на $(i+1, j-1)$, или на $(i+1, j)$, или на $(i+1, j+1)$; в случае $j = M$ последний из трёх описанных вариантов становится невозможным, а в случае $j = 1$ — первый) и закончить маршрут в какой-нибудь клетке нижней строки.

Напишите программу, которая будет находить максимально возможную сумму значений пройденных клеток *среди всех допустимых путей, проходящих хотя бы по одному разу через каждый из столбиков*.

Формат входного файла

В первой строке входа записаны N и M — количество строк и количество столбиков ($2 \leq N \leq 1024$, $2 \leq M \leq 768$, $N \geq M$), дальше в каждой из следующих N строк записано ровно по M разделённых пробелами целых чисел, не превышающих по модулю 10^6 — значения клеток таблицы.

Формат выходного файла

Вывести единственное целое число — найденную максимальную (среди путей, проходящих через все столбики) сумму.

Примеры

stdin	stdout
4 3 1 15 2 9 7 5 9 2 4 6 9 -1	28

Примечание

Ответ равен $28 = 15 + 5 + 2 + 6$, потому что все пути с большей суммой проходят не через все столбики.

Обратите внимание, что в задаче большой размер входного файла. На C++ не рекомендуется использовать `cin` со включенной синхронизацией, на java не рекомендуется использовать `Scanner` — это может привести к тому, что программа просто не успеет прочитать входные данные.

64 мегабайта

Задача N. MaxSum (посетить все столбики ходами коня)

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Есть прямоугольная таблица размером N строк на M столбиков. В каждой клетке записано целое число. По ней нужно пройти сверху вниз, начиная из любой клетки верхней строки, дальше двигаясь вниз ходами коня и закончить маршрут в какой-нибудь клетке нижней строки. То есть, из клетки (i, j) можно перейти в $(i + 1, j - 2)$, или в $(i + 2, j - 1)$, или в $(i + 2, j + 1)$, или в $(i + 1, j + 2)$, исключая варианты, выходящие за пределы доски.

Напишите программу, которая будет находить максимально возможную сумму значений пройденных клеток *среди всех допустимых путей ходами коня, проходящих хотя бы по одному разу через каждый из столбиков*.

Формат входного файла

В первой строке записаны N и M — количество строчек и количество столбиков ($1 \leq N \leq 42$, $1 \leq M \leq 17$); дальше в каждой из следующих N строк записано ровно по M разделённых пробелами целых чисел (каждое не превышает по модулю 10^6) — значения клеток таблицы.

Формат выходного файла

Вывести либо единственное целое число (найденную максимальную среди сумм по маршрутам указанного вида), либо строку `impossible` (без кавычек, маленькими латинскими буквами). Строка `impossible` должна выводиться только в случае, когда не существует ни одного маршрута ходами коня, проходящего через все столбики хотя бы по одному разу.

Примеры

stdin	stdout
4 3 1 15 2 9 7 5 9 2 4 6 9 -1	25
3 3 1 15 2 9 7 5 9 2 4	impossible

Примечание

Для поля 4×3 есть ровно четыре способа спуститься ходами коня, посетив каждый столбик:

первый — $a[1][1] \rightarrow a[2][3] \rightarrow a[4][2]$;

второй — $a[1][2] \rightarrow a[3][1] \rightarrow a[4][3]$;

третий — $a[1][2] \rightarrow a[3][3] \rightarrow a[4][1]$;

четвёртый — $a[1][3] \rightarrow a[2][1] \rightarrow a[4][2]$.

Максимальная возможная сумма $25 = 15 + 4 + 6$ достигается на третьем из них.

Для поля 3×3 таких способов вообще нет.

64 мегабайта

Задача О. MaxSum (счастливая сумма—1)

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Есть прямоугольная таблица размером N строк на M столбиков. В каждой клетке записано целое число. По ней нужно пройти сверху вниз, начиная из любой клетки верхней строки, дальше каждый раз переходя в одну из “нижних соседних” клеток (иными словами, из клетки с номером (i, j) можно перейти или на $(i+1, j-1)$, или на $(i+1, j)$, или на $(i+1, j+1)$; в случае $j = M$ последний из трёх описанных вариантов становится невозможным, а в случае $j = 1$ — первый) и закончить маршрут в какой-нибудь клетке нижней строки.

Напишите программу, которая будет находить максимально возможную *счастливую сумму* значений пройденных клеток среди всех допустимых путей. Всем известно, что счастливыми являются натуральные числа, в десятичной записи которых содержатся только счастливые цифры 4 и 7. Например, числа 47, 744, 4 являются счастливыми, а 0, 5, 17, 467 — не являются. Обратите внимание, что счастливой должна быть именно сумма, а не отдельные слагаемые.

Формат входного файла

В первой строке записаны N и M — количество строчек и количество столбиков ($1 \leq N, M \leq 77$, дальше в каждой из следующих N строк записано ровно по M разделённых пробелами целых чисел (каждое принадлежит диапазону $0 \leq a_{ij} \leq 77$) — значения клеток таблицы.

Формат выходного файла

Вывести либо единственное натуральное число (найденную максимальную среди счастливых сумм), либо строку “impossible” (без кавычек, маленькими латинскими буквами). Строка “impossible” должна выводиться только в случае, когда среди маршрутов указанного вида нет ни одного со счастливой суммой.

Примеры

stdin	stdout
3 4 8 2 10 14 22 2 15 25 1 14 9 1	44

64 мегабайта

Задача Р. MaxSum (счастливая сумма—2)

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Есть прямоугольная таблица размером N строк на M столбиков. В каждой клетке записано целое число. По ней нужно пройти сверху вниз, начиная из любой клетки верхней строки, дальше каждый раз переходя в одну из “нижних соседних” клеток (иными словами, из клетки с номером (i, j) можно перейти или на $(i+1, j-1)$, или на $(i+1, j)$, или на $(i+1, j+1)$; в случае $j = M$ последний из трёх описанных вариантов становится невозможным, а в случае $j = 1$ — первый) и закончить маршрут в какой-нибудь клетке нижней строки.

Напишите программу, которая будет находить максимально возможную *счастливую сумму* значений пройденных клеток среди всех допустимых путей. Всем известно, что счастливыми являются натуральные числа, в десятичной записи которых содержатся только счастливые цифры 4 и 7. Например, числа 47, 744, 4 являются счастливыми, а 0, 5, 17, 467 — не являются. Обратите внимание, что счастливой должна быть именно сумма, а не отдельные слагаемые.

Формат входного файла

В первой строке записаны N и M — количество строчек и количество столбиков ($1 \leq N, M \leq 12$, дальше в каждой из следующих N строк записано ровно по M разделённых пробелами целых неотрицательных чисел (каждое содержит в десятичной записи не более 12 цифр) — значения клеток таблицы.

Формат выходного файла

Вывести либо единственное натуральное число (найденную максимальную среди счастливых сумм), либо строку “impossible” (без кавычек, маленькими латинскими буквами). Строка “impossible” должна выводиться только в случае, когда среди маршрутов указанного вида нет ни одного со счастливой суммой.

Примеры

stdin	stdout
3 4 3 0 10 10 5 0 7 4 4 10 5 4	7

256 мегабайт

Задача Q. Пасьянс (малые ограничения)

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

«N-T пасьянс» — карточная игра для одного игрока. В игре используется $4N$ ($3 \leq N \leq 15$) карт, причем каждой карте соответствует уникальная пара её значения (целое число в диапазоне $1..N$) и масти (\spadesuit , \clubsuit , \heartsuit или \diamondsuit). В начальном положении все карты разложены в T ($4 \leq T \leq 8$) стопок; при этом каждая из первых $(4N)\%T$ стопок содержит по $(4N/T) + 1$ карт, остальные — по $4N/T$ карт (здесь “/” и “%” — целочисленное деление и остаток от деления соответственно). Если сумма значений верхних карт двух стопок равна $N + 1$, то эти две карты можно переместить в отбой (независимо от их мастей). Это единственный способ перемещать карты.

Напишите программу, которая будет определять, какое максимальное количество карт можно переместить в отбой.

Формат входного файла

Первая строка содержит два целых числа N и T , далее идут T строк с описаниями карт соответствующей стопки. Каждая карта описывается её значением (целое число) и мастью (символ с ASCII-кодом 03(\heartsuit), 04(\diamondsuit), 05(\spadesuit), или 06(\clubsuit)) без пробела между ними. Описания разных карт одной стопки разделены ровно одним пробелом, направление описания слева направо соответствует порядку карт снизу вверх.

Формат выходного файла

Ваша программа должна вывести единственное целое число — максимально возможное количество карт, которые можно переместить в отбой.

Примеры

stdin	stdout
3 5 2 \spadesuit 2 \clubsuit 2 \heartsuit 2 \diamondsuit 3 \diamondsuit 1 \heartsuit 3 \clubsuit 1 \spadesuit 1 \clubsuit 3 \heartsuit 1 \diamondsuit 3 \spadesuit	10

128 мегабайт
Задача R. Пасьянс

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

«N-T пасьянс» — карточная игра для одного игрока. В игре используется $4N$ ($3 \leq N \leq 15$) карт, причем каждой карте соответствует уникальная пара её значения (целое число в диапазоне $1..N$) и масти (\spadesuit , \clubsuit , \heartsuit или \diamondsuit). В начальном положении все карты разложены в T ($4 \leq T \leq 12$) стопок; при этом каждая из первых $(4N)\%T$ стопок содержит по $(4N/T) + 1$ карт, остальные — по $4N/T$ карт (здесь “/” и “%” — целочисленное деление и остаток от деления соответственно). Если сумма значений верхних карт двух стопок равна $N + 1$, то эти две карты можно переместить в отбой (независимо от их мастей). Это единственный способ перемещать карты.

Напишите программу, которая будет определять, какое максимальное количество карт можно переместить в отбой.

Формат входного файла

Первая строка содержит два целых числа N и T , далее идут T строк с описаниями карт соответствующей стопки. Каждая карта описывается её значением (целое число) и мастью (символ с ASCII-кодом 03(\heartsuit), 04(\diamondsuit), 05(\spadesuit), или 06(\clubsuit)) без пробела между ними. Описания разных карт одной стопки разделены ровно одним пробелом, направление описания слева направо соответствует порядку карт снизу вверх.

Формат выходного файла

Ваша программа должна вывести единственное целое число — максимально возможное количество карт, которые можно переместить в отбой.

Примеры

stdin	stdout
3 5 2 \spadesuit 2 \clubsuit 2 \heartsuit 2 \diamondsuit 3 \diamondsuit 1 \heartsuit 3 \clubsuit 1 \spadesuit 1 \clubsuit 3 \heartsuit 1 \diamondsuit 3 \spadesuit	10

64 мегабайта

Задача S. Раскладывание Пасьянса

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

«N-T пасьянс» — карточная игра для одного игрока. В игре используется $4N$ ($3 \leq N \leq 15$) карт, причем каждой карте соответствует уникальная пара её значения (целое число в диапазоне $1..N$) и масти (\spadesuit , \clubsuit , \heartsuit или \diamondsuit). В начальном положении все карты разложены в T ($4 \leq T \leq 12$) стопок; при этом каждая из первых $(4N)\%T$ стопок содержит по $(4N/T) + 1$ карт, остальные — по $4N/T$ карт (здесь “/” и “%” — целочисленное деление и остаток от деления соответственно). Если сумма значений верхних карт двух стопок равна $N + 1$, то эти две карты можно переместить в отбой (независимо от их мастей). Это единственный способ перемещать карты.

Напишите программу, которая будет определять, какое максимальное количество карт можно переместить в отбой, и как это сделать.

Формат входного файла

Первая строка содержит два целых числа N и T , далее идут T строк с описаниями карт соответствующей стопки. Каждая карта описывается её значением (целое число) и мастью (символ с ASCII-кодом 03(\heartsuit), 04(\diamondsuit), 05(\spadesuit), или 06(\clubsuit)) без пробела между ними. Описания разных карт одной стопки разделены ровно одним пробелом, направление описания слева направо соответствует порядку карт снизу вверх.

Формат выходного файла

Ваша программа должна вывести целое число S — максимально возможное количество карт, которые можно переместить в отбой. Затем $S/2$ пар чисел, по паре в строке: номера стопок, с которых надо снимать карты на очередном ходу.

Примеры

stdin	stdout
3 5	10
2 \spadesuit 2 \clubsuit 2 \heartsuit	2 4
2 \diamondsuit 3 \diamondsuit 1 \heartsuit	2 3
3 \clubsuit 1 \spadesuit	1 2
1 \clubsuit 3 \heartsuit	4 5
1 \diamondsuit 3 \spadesuit	3 5

Примечание

Если существует несколько способов раскладывания (с одинаковым максимальным суммарным количеством снятых карт), выведите любой.

64 мегабайта

Задача Т. Игра на максимум суммы

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

N карточек выложены в ряд слева направо. На каждой карточке написано целое число. Два игрока поочерёдно забирают по одной карточке, причём забирать можно либо крайнюю левую, либо крайнюю правую. Заканчивается игра, когда забраны все карточки (пока карточки есть, игрок обязан делать один из возможных ходов). Цель игры — получить как можно большую сумму (чисел, записанных на забранных карточках).

Какую максимальную сумму гарантированно сможет набрать первый игрок?

Формат входного файла

В первой строке указано количество карточек N ($1 \leq N \leq 2013$). Во второй строке через пробелы заданы N целых чисел (не превосходящих по модулю 10^3) — значения, записанные на карточках.

Формат выходного файла

Выведите единственное целое число — максимальную сумму, которую гарантированно сможет набрать первый игрок.

Примеры

<code>stdin</code>	<code>stdout</code>
4 1 2 9 3	10

Примечание

Если на первом ходу забрать 1, то соперник в ответ будет вынужден забрать либо 3, либо 2; в любом из этих случаев, первый игрок сможет забрать себе 9, и, таким образом, гарантированно получить сумму 10 (после чего второй игрок забирает последнюю карточку, и игра заканчивается).

Если бы первый игрок на первом ходу забирал 3, то второй мог бы в ответ забрать 9, и в итоге первый получил бы всего лишь $3 + 2 = 5$. При большой глупости, второй игрок мог бы ответить на ход “3” и ходом “1”; в этом случае первый игрок мог бы получить и сумму $3 + 9 = 12$. Но первый игрок не может гарантировать, что второй сделает столь глупый ход, потому и ответ не 12, а 10.

64 мегабайта

Задача U. «Игра» на максимум суммы — подконтрольный «соперник»

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

N карточек выложены в ряд слева направо. На каждой карточке написано целое число. Два игрока поочерёдно забирают по одной карточке, причём забирать можно либо крайнюю левую, либо крайнюю правую. Заканчивается игра, когда забраны все карточки (пока карточки есть, игрок обязан делать один из возможных ходов). Цель игры — получить как можно большую сумму (чисел, записанных на забранных карточках).

Только вот «играют» в эту игру непонятно зачем начальник-самодур и подчинённый-подлиза. Начальник-самодур может целиком и полностью контролировать не только свои ходы, но и ходы подчинённого-подлизы. Какую максимальную сумму сможет набрать начальник-самодур (который, разумеется, ходит первый)?

Формат входного файла

В первой строке указано количество карточек N ($1 \leq N \leq 2013$). Во второй строке через пробелы заданы N целых чисел (не превосходящих по модулю 10^3) — значения, записанные на карточках.

Формат выходного файла

Выведите единственное целое число — максимальную сумму, которую сможет набрать первый игрок (начальник-самодур).

Примеры

<code>stdin</code>	<code>stdout</code>
4 1 2 9 3	12

Примечание

Поскольку начальник-самодур полностью контролирует ходы подчинённого-подлизы, он может забрать первым ходом 3, приказать «сопернику» забирать 1, после чего забрать 9 (получив $3 + 9 = 12$), после чего подчинённый забирает 2 и «игра» заканчивается.

64 мегабайта

Задача V. Разложение в сумму кубов

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Дано натуральное число N . Необходимо представить его в виде суммы минимального количества положительных точных кубов. Программа должна вывести это минимальное количество.

Формат входного файла

Программа получает на вход натуральное число N ($1 \leq N \leq 10^6$).

Формат выходного файла

Программа должна вывести единственное натуральное число — минимальное количество кубов.

Примеры

stdin	stdout
33	5

Примечание

$33 = 2^3 + 2^3 + 2^3 + 2^3 + 1^3$, и 33 не раскладывается в сумму менее чем пяти положительных кубов.