

Задача A. Solid Cube

Имя входного файла: `cube.in`
Имя выходного файла: `cube.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 mebibytes

Представьте себе куб $n \times n \times n$, собранный из нескольких деталей различной формы. В случае, если детали соответствующим образом зацеплены, разделить куб на несколько частей можно, только деформировав или разрезав одну из деталей. Назовём этот случай «прочным».

По заданным деталям (каждую из которых можно представить как объединение некоторого числа единичных кубиков) вычислите, является ли собранный из них куб прочным.

Формат входного файла

В первой строке входного файла задано количество тестовых примеров T , $1 \leq T \leq 10$. Далее следуют тестовые примеры, имеющие следующий формат. В первой строке тестового примера задано число n — длина ребра куба ($1 \leq n \leq 10$). Вторая строка является пустой. Далее следуют n блоков, содержащие n строк по n символов в каждом, описывающие детали (по слоям сверху вниз). После каждого из блоков (включая последний, кроме случая последнего блока в последнем тестовом примере) идёт пустая строка. Детали задаются строчными буквами от A до Z. При этом два кубика принадлежат одной детали тогда и только тогда, когда они имеют общую грань и один и тот же цвет (например, в первом тестовом примере из условия куб разрезан на три детали).

Формат выходного файла

Выведите “Yes”, если куб является прочным, и “No” в противном случае.

Пример

<code>cube.in</code>	<code>cube.out</code>
2	No
2	Yes
PQ	
PQ	
QQ	
QP	
3	
XXX	
YUY	
XXX	
XXX	
XUX	
XXX	
XUX	
XUX	
XUX	

Задача В. Expedition

Имя входного файла: `expedition.in`
Имя выходного файла: `expedition.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 mebibytes

Группа энтузиастов, в составе которой присутствует бывалый нумизмат, отправилась на поиски клада из старинных монет. Клад был найден, и возник вопрос о том, как делить монеты. Нумизмат предложил следующий вариант — часть монет разделить поровну, чтобы каждому досталось одинаковое количество монет, а часть (возможно, нулевую) отдать в качестве дара в местный краеведческий музей. При этом сам он готов довольствоваться даже меньшим количеством монет, чем то, достанется каждому из остальных участников экспедиции.

Разумеется, нумизмат определил, какие монеты являются наиболее ценными (для непосвящённых монеты более-менее одинаковы — номинал один и тот же, размер тоже), определил стоимость всех монет по каталогу, так что он хочет выбрать свою долю так, чтобы она имела максимальную стоимость.

Вычислите, какова может быть максимальная стоимость монет, отобранных нумизматом.

Формат входного файла

В первой строке входного файла заданы два целых числа N и M — количество участников экспедиции (кроме нумизмата) и количество найденных монет ($1 \leq N, M \leq 1000$). Во второй строке заданы M целых чисел p_i — стоимость каждой из найденных монет по каталогу ($1 \leq p_i \leq 1000$).

Формат выходного файла

Выведите одно целое число — максимальную стоимость монет, которые достанутся нумизмату при указанном способе дележа.

Пример

<code>expedition.in</code>	<code>expedition.out</code>
1 1 5	0
5 9 3 3 7 7 9 10 13 17 15	17

Пояснение к примеру 2: монеты стоимостью 3, 3 и 7 отправляются в музей, после чего каждому достаётся по одной монете. Нумизмат при этом берёт себе монету стоимостью 17.

Задача С. Experiment

Имя входного файла: `experiment.in`
Имя выходного файла: `experiment.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 mebibytes

Садовод-мичуринец Трофим решил вырастить самый большой арбуз в мире. Для этого он выбрал n сортов арбузов и проводит перекрёстное скрещивание. Трофим выяснил, что если у двух сортов арбуза радиусы равны, соответственно, x и y , где x и y — целые числа (у садоводов-мичуринцев не бывает арбузов с дробными радиусами), то у их гибрида радиус равен наименьшему целому числу, не меньшему $(x + y)/2$. Трофим произвёл первичное скрещивание, получив n^2 разных сортов арбузов (естественно, включая и первоначальные n — при скрещивании одинаковых сортов). Теперь он хочет из них выбрать n сортов с наибольшими радиусами.

Напишите программу, которая по заданному набору первоначальных радиусов арбузов решает столь полезную народному хозяйству задачу. Учтите, что гибриды (a,b) и (b,a) считаются различными сортами.

Формат входного файла

В первой строке входного файла задано целое число n ($1 \leq n \leq 10^4$) — первоначальное количество сортов. Во второй строке заданы n целых чисел r_i ($1 \leq r_i \leq 10^5$) — радиусы каждого из сортов.

Формат выходного файла

Выведите n чисел — радиусы самых крупных из получившихся гибридов. Числа должны быть отсортированы по невозрастанию.

Примеры

<code>experiment.in</code>	<code>experiment.out</code>
3 4 5 6	6 6 6
4 4 9 5 2	9 7 7 7

Задача D. Matrixmaze

Имя входного файла: `matrixmaze.in`
Имя выходного файла: `matrixmaze.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 mebibytes

Лабиринт представляет собой матрицу $N \times M$, заполненную цифрами от 0 до 9. В качестве стартовой точки можно выбрать любую из N клеток в первой строке. Чтобы выйти из лабиринта, необходимо дойти до любой из N клеток в M -й строке.

Перемещаться можно в любую из клеток, имеющую с текущей общую сторону. При этом на всём протяжении пути количество различных цифр, записанных в клетках, через которые проходит путь (так называемых «ключевых» цифр), не должно превосходить 3.

Найдите лексикографически наименьшую тройку «ключевых» цифр, которая позволяет выйти из лабиринта (то есть тройка 0, 0, 0 следует перед тройкой 0, 0, 1, затем идёт 0, 0, 2 и так далее до 9, 9, 8 и 9, 9, 9; в частности, из требования лексикографической минимальности следует, что цифры в тройке надо выводить отсортированными по неубыванию). Если цифр менее трёх, то тройка формируется добавлением любой из цифр. В случае, если из лабиринта выйти нельзя, выведите три числа -1 .

Формат входного файла

В первой строке выходного файла заданы два целых числа N и M — длина и высота матрицы ($1 \leq M, N \leq 100$). Далее следуют M строк, содержащие по N чисел в промежутке от 0 до 9 каждое. j -е число в i -й строке соответствует клетке матрицы с координатами (i, j) .

Формат выходного файла

Выведите три числа — лексикографически наименьшую тройку «ключевых» цифр, или три -1 , если выйти из лабиринта невозможно.

Пример

<code>matrixmaze.in</code>	<code>matrixmaze.out</code>
6 5 0 1 2 3 8 7 0 0 0 1 1 1 0 1 2 3 3 8 7 1 8 1 9 8 9 7 6 2 4 6	0 1 7

Задача E. Byteland Mobile Operators

Имя входного файла: `opsos.in`
Имя выходного файла: `opsos.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 mebibytes

В Байтландии есть некоторое количество городов и дорог, соединяющих эти города. Представители двух крупнейших сотовых операторов решили поделить страну между собой. Изначально в некоторых городах работает только первый оператор, Gigaphone (обозначим его буквой G), в некоторых — только второй, Bearline (обозначим его буквой B). В некоторых городах работают только местные операторы. Обозначим такие города буквой R.

Передел было решено провести в форме следующей игры: представитель каждой компании на своём ходу выбирает один или несколько городов, в которых работают только местные операторы, при этом ни один из выбранных городов не должен быть напрямую соединён дорогой с городом, в котором уже работает данный оператор. В качестве дополнительного «бонуса» представители договорились, что тот из них, кто первым не сможет сделать выбор, оплатит из своего кармана банкет по случаю достижения договорённостей. Считая, что первым выбор делает представитель Gigaphone и что каждый представитель в первую очередь хочет избежать оплаты банкета из собственного кармана, выясните, представитель какой компании всё же вынужден будет оплачивать банкет при оптимальных действиях каждого.

Формат входного файла

В первой строке входного файла описано начальное распределение городов между операторами. Длина этой строки совпадает с количеством городов n . ($1 \leq n \leq 10$).

Символ на позиции i равен 'R', когда i -й город не занят, 'G' — когда город занят Gigaphone, 'B' — когда город занят Bearline.

Далее следуют n строк по n символов в каждой — описание конфигурации дорог. Символ на позиции i в строке под номером j равен '+', когда город i связан с городом j дорогой, и '-' в противном случае. Гарантируется, что если город i связан с j , то и город j связан с i .

Формат выходного файла

Выведите "G", если банкет оплачивает представитель Gigaphone, и "B", если представитель Bearline.

Пример

<code>opsos.in</code>	<code>opsos.out</code>
BBGR --+- ---+ +--- -+--	B
RGRG ---+ --+- -+-+ +--+	G

Задача F. Path through the segments

Имя входного файла: `path.in`
Имя выходного файла: `path.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 mebibytes

На клетчатой бумаге задан квадрат. Вершина $(1, 1)$ расположена в левом верхнем углу квадрата, вершина (n, n) — в правом нижнем. На каждой из n горизонтальных линий, составляющих квадрат, выбраны по две точки с ординатами L_i и R_i ($1 \leq L_i \leq R_i \leq n$). Требуется найти длину кратчайшего пути по сетке, который бы начинался в $(1, 1)$, заканчивался в (n, n) , не содержал бы движений вверх и проходил через все точки (i, L_i) и (i, R_i) . При этом разрешается проходить по одному и тому же ребру дважды.

Формат входного файла

В первой строке входного файла задано целое число n — количество строк и столбцов в квадрате ($1 \leq n \leq 2 \cdot 10^4$). В i -й из последующих n строк заданы два целых числа L_i и R_i ($1 \leq L_i \leq R_i \leq n$).

Формат выходного файла

Выведите одно целое число — длину кратчайшего пути, удовлетворяющего условию задачи.

Пример

<code>path.in</code>	<code>path.out</code>
7 2 7 3 5 1 4 1 3 3 7 4 6 3 5	34