Букеты

Будем покупать цветы в порядке неубывания их стоимости. При решении задачи можно придерживаться следующей схемы:

```
Сформировать множество букетов S – множество кандидатов быть первым купленным букетом.
Для i от 1 до N
  Выбрать из S самый дешевый букет - это и будет i-й купленный букет.
  Удалить выбранный букет из S
  Добавить в S букеты, которые могут оказаться (i+1)-ым дешевым букетом.
```

Очевидно, что самый 1-й букет, который будет куплен, состоит из одного цветка. Поэтому в начале алгоритма мы инициализируем множество S, поместив в него все возможные букеты, состоящие из одного цветка. Аналогично, когда мы на очередном шаге алгоритма выбрали самый дешевый букет из S (обозначим его P), нам имеет смысл добавить во множество S только букеты, полученные добавлением к P ровно одного цветка (понятно, что букеты, полученные добавлением к P двух и более цветков будут заведомо хуже).

S можно организовать при помощи обычного массива.
Обратите также внимание на следующий тонкий момент. Если не принимать никаких мер, то один и тот же букет может попасть во множество S несколько раз. К примеру, букет, состоящий из одного цветка 1-го типа и одного цветка 2-го типа, может попасть в S двумя способами: после того, как к букету из одного цветка 1-го типа добавлен цветок 2-го типа, и после того, как к букету из одного цветка 2-го типа добавлен цветок 1-го типа. Довольно простой способ борьбы с этой проблемой заключается в следующем: можно добавлять к букету P не все возможные цветки, а только цветки с номером, большим или равным Q, где Q - это цветок с максимальным номером в букете P. Тогда у любого букета будет только один способ попасть во множество S, так как цветки добавляются в букет в порядке неубывания номеров.

Стены

Произведем сортировку точек в порядке возрастания их абсциссы и обозначим их (x_i, y_i). Допустим, мы уже соединили первые k точек и пытаемся соединить k+1 точку:

- с правой стороной дома. Так как все предыдущие K точек имеют координату x не большую, чем данная, то мы всегда можем это сделать.
- с левой стороной дома. Некоторые из предыдущих точек уже могли соединиться с верхней или нижней стеной и это может воспрепятствовать Коле построить стену влево. Из всех точек соединенных с верхней стеной выберем ту, координата y которой – минимальна (u_{min}). Также, из всех точек соединенных с нижней стеной выберем ту, координата y которой – максимальна (d_{max}). Соединение возможно, если $u_{min} < y_{k+1} < d_{max}$ и при этом $y_i \neq y_{k+1}$, $i \leq k$.
- с верхней или нижней стороной дома. Некоторые из предыдущих точек уже могли соединиться с правой стеной. Выберем точку с максимальной координатой y, соединенную с правой стороной (r_{max}). Выберем точку с минимальной координатой y, соединенную с правой стеной (r_{min}). Соединение возможно с верхней стеной возможно, если $y_{k+1} > r_{max}$ и при этом $y_i \neq y_{k+1}$, $i \leq k$. Соединение возможно с нижней стеной возможно, если $r_{min} > y_{k+1}$ и при этом $y_i \neq y_{k+1}$, $i \leq k$.

Эти 5 параметров (K; точка с минимальной y-координатой, соединенная с верхней стеной; точка с максимальной y-координатой, соединенная с нижней стеной; точка с минимальной y-координатой соединенная с правой стеной; точка с максимальной y-координатой, соединенная с правой стеной) определяют все возможные состояния, с которыми мы можем встретиться при построении стен. Эти состояния, вместе с правильно построенными переходами между собой, образуют ориентированный связный ацикличный граф. Следовательно, задача сводится к обходу этого графа вариантов и сохранению в массиве промежуточных решений, необходимых для нахождения ответа. Также эту задачу можно рассматривать как задачу динамического программирования. Сложность алгоритма O(N*A^4).

Перестановки

Обозначим длину строки W через l. Программа должна хранить два списка (WList, SList), строки которых соответствуют возможным символам в строках W и S. В списке WList должно храниться количество вхождений каждого символа в строку W. В списке SList должно храниться количество вхождений каждого символа в подстроку строки S длины l. Последовательно перебирая подстроки строки S длины l, мы сравниваем списки WList и SList. Если эти списки совпадают, то данная подстрока строки S эквивалентна W.
При таком подходе естественно строить список SList не каждый раз заново для каждой новой подстроки S, а используя тот факт, что список SList для последовательно расположенных подстрок одной длины отличается не более чем в двух позициях. Сравнение списков WList и SList необходимо проводить используя тот же факт, что описан выше.

Пирамида

Вычислим сумму высот всех прямоугольников размера $a \times b$ и $c \times d$, которые помещаются на поле боя и занесем информацию о них в два списка Monuments и Tombs. Затем будем перебирать пары прямоугольников такие, что один прямоугольник принадлежит Monuments, а второй Tombs, причем второй содержится в первом и не касается его границ. Для описанных выше пар прямоугольников посчитать разность между сумой высот первого и второго. Та пара прямоугольников, которая даст максимальное значение этой разнице и будет являться решением, т.к. искомая средняя высота основания получается из полученной оценки делением на количество клеток ($a*b-c*d$), которое величина постоянная для всех пар прямоугольников.

Букеты

На дворе весна. 8 марта. У школьника Васи N подружек и он хочет подарить каждой из них по букету цветов. Когда Вася обошел весь цветочный рынок, он узнал, что:

- Продается равно M видов цветов.
- Цена одного цветка i-го вида - F[i] рублей. (0 <= I <= M-1).

Вася хочет, чтобы каждый букет был непохож на другие. Два букета A и B идентичны, если для каждого i от 0 до M-1 включительно количество цветов i-го вида в букете A равно количеству цветов i-го вида в букете B. Под букетом Вася понимает непустое конечное множество цветов (неважно каких типов).
Стоимость одного букета равна сумме стоимостей всех цветов в наборе.

Требования.

Вася хочет минимизировать затраты на покупку букетов. Поэтому он просит тебя написать программу, помогающую сделать наиболее оптимальный выбор. Если способов подобрать букеты с минимальной стоимостью несколько, то указать любой из этих способов.
Количество баллов за полное решение задачи: **20 баллов.**

Формат входных данных (input.txt).

В первой строке содержится число N ($1 \le N \le 100$) и число M ($1 \le M \le 100$) разделенные пробелом. Во второй строке указываются стоимости цветов каждого вида: F[0], F[1]… F[N-1]. Каждый элемент массива F от 1 до 100000 включительно.

Формат выходных данных (output.txt).

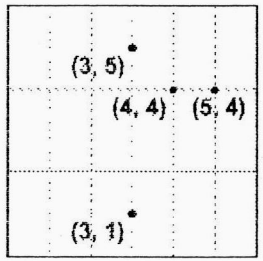
Вывести целое число, равное минимальной суммарной стоимости букетов.

Примеры входного и выходного файлов:

input.txt	output.txt
100 1 1	5050
1 1 100	100
10 2 1 1	24

Стены

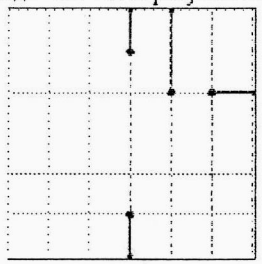
Инженер Коля решил сделать ремонт своего дома. Дело в том, что жизненного пространства для нормальной жизни ему стало не хватать – он завел семью. Так как места для дополнительной пристройки нет, он решил перепланировать архитектуру всех стен (кроме внешних) так, чтобы как можно больше уменьшить их суммарную длину.
Сначала Коля схематически представил дом в виде квадрата (т.к. дом у него квадратный). Горизонтальные и вертикальные прямые в этом квадрате - это места, где могли бы располагаться новые стены.
Далее прикинул, на какие места приходится основное давление, и представил их в виде точек в узлах пересечения прямых. Координаты отсчитываются от левого, нижнего угла дома



Проделав всю предыдущую работу, Коля стал прикидывать новое положение стен, удовлетворяющее следующим условиям:

1. Стена – это горизонтальный или вертикальный отрезок, берущий начало в одной из точек с наибольшим давлением, и заканчивающийся впритык с внешними стенами дома.
2. Стен должно получиться ровно столько, сколько точек с наибольшим давлением.
3. Стены не должны пересекаться, перекрываться, касаться.

В данном случае ответ для приведенного выше рисунка можно проиллюстрировать так:



После нескольких неудачных попыток, Коля понял, что он сильно рискует. Если суммарная длина стен будет хоть немного отличаться от намеченной (т.е. минимальной) его дом может быть небезопасен для жизни. Поэтому он просит тебя написать программу, которая поможет нарисовать схему дома с минимальной суммарной длиной стен.

Требования.

Найти минимальную возможную сумму длин всех стен, а также расположение этих стен в доме. Если расположений стен с минимальной суммой несколько, то указать любое.
Количество баллов за полное решение задачи: **30 баллов.**

Формат входных данных (input.txt).

Первая строка содержит целое число A (2 <= A <= 30) – длина стороны дома.
Вторая строка содержит целое число N (1 <= N <= 50) – количество точек с максимальным давлением.
Далее идут N строк, содержащих 2 целых числа X и Y (1<=X, Y<=A-1) – координаты точек с максимальным напряжением. Точек с одинаковыми координатами нет.
Вы можете полагать, что решение существует для любого входного файла.

Формат выходных данных (output.txt).

Первая строка содержит минимальную суммарную длину стен.
Остальные N строк должны описывать положения стен. Для каждой точки из входного файла в отдельной строке напишите "up", "down", "left" или "right" в зависимости от ориентации стены.
Если есть несколько решений, выведите любое из них.

Примеры входного и выходного файлов:

input.txt	output.txt
6 4 3 1 3 5 5 4 4 4	5 down up right up
10 4 5 1 5 2 4 3 6 3	13 down right down right

Пирамида

После победы в великой битве, Великий Ягуар король Ацтеков, решил построить пирамиду, которая будет сразу и памятником его победе и склепом для погибших героев. Пирамида должна быть построена на поле боя.
Королевские архитекторы разметили поле боя на квадраты, которых получилось n – строк и m – столбцов. Каждый из квадратов имеет высоту над уровнем поля боя, представленную целым числом.
Король хочет, чтобы пирамида была размером $a \times b$ квадратов, а внутри нее была прямоугольное помещение под склеп размером $c \times d$ квадратов. При этом склеп должен остаться на том уровне, на котором он есть сейчас, а остальная поверхность пирамиды должна быть выровнена. Поскольку привозить землю издалека не имеет смысла, король решил, что землю с более высоко расположенных квадратов надо перенести на те которые расположены ниже. Таким образом, высота основания пирамиды получиться в результате усреднения высот всех квадратов на которых расположена пирамида, за исключением тех квадратов, где будет располагаться склеп.

	1	2	3	4	5	6	7	8
1	1	5	10	3	7	1	2	5
2	6	12	4	4	3	3	1	5
3	2	4	3	1	6	6	19	8
4	1	1	1	3	4	2	4	5
5	6	6	3	3	3	2	2	2

Требования.

Помогите архитекторам выбрать место для построения пирамиды и акрополя в ней, так чтобы высота основания (пирамида без учета акрополя) была максимальна. Если решений несколько, вывести любое.
Количество баллов за полное решение задачи: **30 баллов.**

Формат входных данных (input.txt).

Первая строка содержит шесть целых чисел, записанных через пробел: *n, m, a, b, c, d* (все эти числа больше 0 и меньше 100). Следующие *n* строк содержат по *m* целых чисел. Число в *i*-й строке и *j*-м столбце представляет относительную высоту квадрата (i,j) на поле боя.

Формат выходных данных (output.txt).

Первая строка содержит координаты левого верхнего угла пирамиды. Вторая строка содержит координаты левого верхнего угла акрополя.

Примеры входного и выходного файлов:

input.txt	output.txt
5 8 3 5 1 2 1 5 10 3 7 1 2 5 6 12 4 4 3 3 1 5 2 4 3 1 6 6 19 8 1 1 1 3 4 2 4 5 6 6 3 3 3 2 2 2	1 4 2 6
4 4 3 3 1 1 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4	1 1 2 2

Перестановки

Будем считать, что две строки символов эквивалентны, если вторую строку можно получить из первой переставив символы местами. Например строка 'string' эквивалентна строке 'trinsg', а строка 'abc' эквивалентна строке 'bac'. Даны строки S и W, причем S длиннее W.

Требования.

Найти в S все подстроки (из подряд идущих символов) которые эквивалентны W.
Количество баллов за полное решение задачи: **20 баллов.**

Формат входных данных (input.txt).

Первая строка содержит два целых числа, записанных через пробел, определяющие длины W и S соответственно (оба числа меньше 200). Вторая строка содержит W. Третья строка содержит S.

Формат выходных данных (output.txt).

На выходе должно быть одно число – количество подстрок.

Примеры входного и выходного файлов:

input.txt	output.txt
3 10 abc htbcabhpac	2
3 6 klm abcdef	0