



## Задача В. Треугольник в прямоугольнике

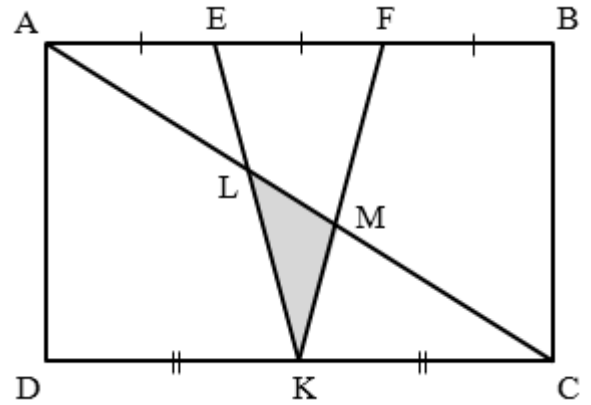
Ограничение по времени: 1 секунда

Ограничение по памяти: 32 Мб

Дан прямоугольник ABCD со сторонами  $a$  и  $b$  ( $a = AD = BC$ ,  $b = AB = DC$ ). Сторону AB разбили на три равные части точками E и F. Сторону DC разбили на две равные части точкой K.

Затем провели диагональ AC и отрезки EK и FK, которые образовали пересечение с диагональю AC в точках L и M.

По длинам сторон прямоугольника ABCD требуется определить значение площади треугольника LMK.



### Входные данные

В первых двух строках входного файла INPUT.TXT записаны два целых числа  $a$  и  $b$  – длины сторон прямоугольника ABCD ( $1 \leq a, b \leq 1000$ ).

### Выходные данные

В выходной файл OUTPUT.TXT выведите искомую площадь треугольника LMK с точностью не хуже, чем  $10^{-3}$ .

### Пример

№	INPUT.TXT	OUTPUT.TXT
1	7 10	3

## Задача С. Весёлые качели

Ограничение по времени: 1 секунда

Ограничение по памяти: 32 Мб

Однажды во дворе многоэтажки родители для своих  $n$  детей построили большие и прочные качели типа «качалка-балансир» в виде длинной балки, подвешенной в центре тяжести на шарнире так, что она может качаться в вертикальной плоскости.

Оказалось, что качаться на этих качелях интересно, когда качели уравновешены, то есть если модуль суммы моментов детей слева от центра качелей совпадает с модулем суммы моментов детей справа от центра. Напомним, что момент ребёнка равен произведению его массы на расстояние от него до центра качелей.

По заданной массе детей требуется выбрать им места на качелях так, чтобы все они сидели в различных позициях на целочисленном расстоянии от центра качелей и могли интересно качаться, то есть качели должны быть уравновешены. При этом никакой ребенок не должен сидеть в центре качелей.

Более формально: для каждого ребёнка номер  $i$  с массой  $m_i$  необходимо определить целое число  $d_i$  ( $d_i \neq 0$ ) – его позицию на качелях так, что  $\sum_{i=1}^n m_i d_i = 0$  и все  $d_i$  различны.

### Входные данные

Первая строка входного файла INPUT.TXT содержит целое число  $n$  ( $1 \leq n \leq 1000$ ) – количество детей.

Во второй строке заданы  $n$  целых чисел  $m_1, m_2, \dots, m_n$  ( $1 \leq m_i \leq 100$ ), где  $m_i$  обозначает массу  $i$ -го ребёнка в килограммах.

### Выходные данные

В выходной файл OUTPUT.TXT выведите «No», если решения не существует. В противном случае в первой строке выведите «Yes», а во второй строке – последовательность различных целых чисел  $d_i$  ( $1 \leq |d_i| \leq 10^9$ ). Если существует несколько решений, выведите любое.

Гарантируется, что если решение существует, то найдётся и решение для  $|d_i| \leq 10^9$  при всех  $i$ .

### Примеры

№	INPUT.TXT	OUTPUT.TXT	Пояснение
1	1 1	No	Нет решения
2	2 3 8	Yes -8 3	$3 \cdot (-8) + 8 \cdot (-3) = -24 + 24 = 0$
3	5 1 2 3 4 2	Yes 5 -10 13 -8 4	$1 \cdot 5 + 2 \cdot (-10) + 3 \cdot 13 + 4 \cdot (-8) + 2 \cdot 4 = 0$

### Система оценки

Решения, работающие только для случаев, когда присутствует хотя бы один ребёнок с массой в 1 килограмм, будут оцениваться в 50 баллов.

## Задача D. Башенки из кубиков

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 Мб

Маленький Рома любит играть в кубики и строить из них башенки. Всего у него  $N$  кубиков, каждый из которых определяется целочисленным размером  $A_i$ , который определяет длину ребра  $i$ -го кубика. Если поставить несколько кубиков последовательно друг на друга, то получается башенка.

Рома считает башенку *красивой*, если каждый последующий сверху вниз кубик имеет размер на единицу больше, чем предыдущий. Высота башенки – это количество кубиков в ней. То есть для красивой башенки высоты  $M$ , состоящей из кубиков с размерами  $S_1, S_2, \dots, S_M$ , должно выполняться условие:  $S_{i-1} = S_i - 1$  для  $i$  от 2 до  $M$ .

Рома хочет для построения башенок использовать все кубики. Он заметил, что чем меньше в результате получается красивых башенок при одном и том же наборе кубиков, тем выше в среднем получаются башенки. Однако, кубиков у Ромы очень много и у него не получается минимизировать число красивых башенок так, чтобы в результате построения были использованы все кубики.

Помогите Роме с этой задачей!

### Входные данные

Первая строка входного файла INPUT.TXT содержит число  $N$  – количество кубиков у Ромы ( $1 \leq N \leq 2 \times 10^5$ ). Во второй строке содержатся  $N$  целых чисел  $A_1, A_2, \dots, A_N$ , где  $A_i$  – размер  $i$ -го кубика ( $1 \leq A_i \leq 2 \times 10^5$ ).

### Выходные данные

В первой строке выходного файла OUTPUT.TXT выведите целое число  $K$  – минимальное количество красивых башенок, которые можно построить из имеющихся кубиков.

В следующих  $K$  строках выведите описание башенок по одному описанию башенки в каждой отдельной строке. Описание башенки состоит из двух целых чисел, где первое число – размер самого маленького (верхнего) кубика в башенке, а второе число – размер самого большого (нижнего) кубика в башенке.

Если существует несколько решений задачи, выведите любое из них.

### Примеры

№	INPUT.TXT	OUTPUT.TXT
1	3 1 2 3	1 1 3
2	5 6 2 3 2 1	3 1 3 2 2 6 6

### Система оценки

Решения, работающие только для  $N \leq 100$ , будут оцениваться в 40 баллов.

Решения, работающие только в тех случаях, когда размеры всех кубиков различны, будут оцениваться в 40 баллов.

## Задача Е. Площадь многоугольника

*Ограничение по времени: 1 секунда*

*Ограничение по памяти: 32 Мб*

По длинам сторон многоугольника требуется определить его максимально возможную площадь.

### Входные данные

Первая строка входного файла INPUT.TXT содержит целое число  $N$  – количество сторон многоугольника ( $3 \leq N \leq 100$ ). Вторая строка входных данных содержит  $N$  целых чисел  $L_i$  – длины сторон многоугольника ( $1 \leq L_i \leq 100$ ).

### Выходные данные

В выходной файл OUTPUT.TXT выведите значение максимально возможной площади многоугольника с точностью не хуже, чем  $10^{-2}$ . Если составить многоугольник невозможно, выведите «0».

### Примеры

№	INPUT.TXT	OUTPUT.TXT
1	3 5 5 5	10.825318
2	6 9 9 9 9 9 9	210.444173
3	4 15 12 15 30	252
4	3 1 2 5	0

### Система оценки

Решения, работающие только для  $N=3$ , будут оцениваться в 20 баллов.

Решения, работающие только для равносторонних многоугольников, будут оцениваться в 20 баллов.

## Задача F. Поход за грибами

Ограничение по времени: 5 секунд

Ограничение по памяти: 128 Мб

Однажды грибник Макар попал в трёхмерный лес, в котором растут деревья и грибы. Лес состоит из  $N$  уровней, расположенных друг над другом. Каждый уровень – это прямоугольная площадка, разбитая на  $N \times M$  участков. Макар может свободно перемещаться по свободным участкам в шести направлениях: вверх, вниз, влево, вправо, вперед и назад. Оказавшись на одном участке с грибом, Макар может его сорвать. Однако сквозь деревья и за пределы леса Макар ходить не может. При любом перемещении из одного участка в смежный Макар делает один шаг.

У Макара есть карта леса, включающая расположение деревьев и грибов в нём, а также текущее положение Макара. Некоторые грибы (возможно, все) могут быть недостижимы.

Требуется определить: какое максимальное число грибов сможет собрать Макар и какое минимальное количество шагов ему на это потребуется.

### Входные данные

Первая строка входного файла INPUT.TXT содержит целые числа  $H$ ,  $N$ ,  $M$  – размеры леса ( $2 \leq H, N, M \leq 50$ ). Далее идет описание  $N$  уровней снизу вверх. Каждый уровень описывается  $N$  строками по  $M$  символов в каждой: «.» обозначает свободный участок, «#» – дерево, заглавная буква английского алфавита «M» – гриб и заглавная буква английского алфавита «P» – положение Макара. Описание каждого уровня завершается пустой строкой. Гарантируется, что количество грибов на карте не превосходит 20.

### Выходные данные

В выходной файл OUTPUT.TXT выведите два целых числа: максимальное число грибов, которые может собрать Макар, и минимальное количество шагов, которые Макар должен сделать, чтобы собрать эти грибы.

### Пример

INPUT.TXT	OUTPUT.TXT
3 3 3 . #. .M  ### P.# .#  M## #. #.M	2 10

### Система оценки

Решения, работающие только для  $H, N, M \leq 10$ , будут оцениваться в 40 баллов.

Решения, работающие только для решений с не более чем 10 грибами, будут оцениваться в 80 баллов.