

Задача 5. «Парк аттракционов»

Имя входного файла:	<code>attract.in</code>
Имя выходного файла:	<code>attract.out</code>
Максимальное время работы на одном тесте:	1 секунда
Максимальный объем используемой памяти:	256 мегабайт
Максимальная оценка	100 баллов

В городе π недавно построили парк аттракционов, в котором есть павильон игровых автоматов. Каждый из автоматов рассчитан на одного человека. В программе Всероссийской олимпиады планируется посещение этого павильона.

Перед организаторами встала сложная задача — составить расписание игры участников олимпиады на автоматах таким образом, чтобы каждый из N участников олимпиады смог поиграть на каждом из автоматов, и при этом автобус, увозящий участников из парка олимпиады, смог бы отправиться к месту проживания как можно раньше.

Время перемещения участников между автоматами, а также между автобусом и павильоном считается равным нулю. Каждый из участников в любой момент времени может как играть на автомате, так и ждать своей очереди, например, гуляя по парку. Для каждого из M ($M \leq N$) автоматов известно время игры на нём t_i ($1 \leq i \leq M$). Прервать начатую игру на автомате невозможно. Автобус привозит всех участников олимпиады в парк одновременно в нулевой момент времени.

Требуется написать программу, которая по заданным числам N , M и t_i определяет оптимальное расписание игры на автоматах для каждого из участников.

Формат входных данных

В первой строке входного файла содержатся два числа: N и M ($1 \leq M \leq N \leq 100$). Во второй строке заданы M целых чисел t_i ($1 \leq t_i \leq 100$), каждое из которых задаёт время игры на i -м автомате ($1 \leq i \leq M$). Числа в строке разделяются одиночными пробелами.

Формат выходных данных

В первой строке необходимо вывести одно число — минимально возможное время отправления автобуса из парка аттракционов. Далее необходимо вывести N расписаний игр на автоматах, по одному для каждого из участников. Каждое расписание описывается в $(M + 1)$ строках, первая из которых — пустая, а далее следуют M строк, описывающих автоматы в порядке их посещения этим участником. Посещение автомата описывается двумя целыми числами: номером автомата j ($1 \leq j \leq M$) и временем начала игры участника на этом автомате.

Примеры

<code>attract.in</code>	<code>attract.out</code>
2 1 2	4 1 0 1 2
3 2 2 1	6 1 0 2 2 1 2 2 4 2 0 1 4

Подзадачи и система оценки

Данная задача содержит пять подзадач. Для оценки каждой подзадачи используется своя группа тестов. Баллы за подзадачу начисляются только в том случае, если все тесты из этой группы пройдены.

Подзадача 1 (20 баллов)

$M = 1$, $1 \leq N \leq 100$, t_1 лежит в пределах от 1 до 100.

Подзадача 2 (20 баллов)

Все t_i равны 1, $N = M$.

Подзадача 3 (20 баллов)

Все t_i равны 1, $N > M$.

Подзадача 4 (20 баллов)

Числа t_i лежат в пределах от 1 до 100, $N = M$.

Подзадача 5 (20 баллов)

Числа t_i лежат в пределах от 1 до 100, $N > M$.

Обратная связь

В течение тура можно не более 10 раз запросить результаты работы своей программы на тестах жюри. Запрос можно делать не чаще одного раза в 5 минут. Для каждого теста сообщается результат запуска программы на этом тесте.

В этой задаче можно выбрать, какое решение будет оцениваться. В этом случае баллы начисляются за лучшее решение из следующих:

- выбранного явно;
- последнего принятого на проверку решения.

Если выбор не сделан, то будет оцениваться лучшее решение из следующих:

- решений, по которым просмотрены баллы;
- последнего принятого на проверку решения.

Задача 6. «Велогонка»

Имя входного файла:	bicycle.in
Имя выходного файла:	bicycle.out
Максимальное время работы на одном тесте:	2 секунды
Максимальный объем используемой памяти:	256 мегабайт
Максимальная оценка	100 баллов

Велосипедисты, участвующие в шоссейной гонке, в некоторый момент времени, который называется начальным, оказались в точках, удалённых от места старта на x_1, x_2, \dots, x_n метров (n – общее количество велосипедистов). Каждый велосипедист движется со своей постоянной скоростью v_1, v_2, \dots, v_n метров в секунду. Все велосипедисты движутся в одну и ту же сторону.

Репортёр, освещающий ход соревнований, хочет определить момент времени, в который расстояние между лидирующим в гонке велосипедистом и замыкающим гонку велосипедистом станет минимальным, чтобы с вертолётá сфотографировать сразу всех участников велогонки.

Требуется написать программу, которая по заданному количеству велосипедистов n , заданным начальным положениям велосипедистов x_1, x_2, \dots, x_n и их скоростям v_1, v_2, \dots, v_n , вычислит момент времени t , в который расстояние l между лидирующим и замыкающим велосипедистом будет минимальным.

Формат входных данных

Первая строка входного файла содержит целое число n – количество велосипедистов.

В последующих n строках указаны по два целых числа: x_i – расстояние от старта до i -го велосипедиста в начальный момент времени ($0 \leq x_i \leq 10^7$) и v_i – его скорость ($0 \leq v_i \leq 10^7$).

Формат выходных данных

В выходной файл необходимо вывести два вещественных числа: t – время в секундах, прошедшее от начального момента времени до момента, когда расстояние в метрах между лидером и замыкающим будет минимальным, l – искомое расстояние.

Числа t и l должны иметь абсолютную или относительную погрешность не более 10^{-6} , что означает следующее. Пусть выведенное число равно x , а в правильном ответе оно равно y . Ответ будет считаться правильным, если значение выражения $|x - y| / \max\{1, |y|\}$ не превышает 10^{-6} .

Примеры

bicycle.in	bicycle.out
3 0 40 30 10 40 30	1 30
5 90 100 100 70 100 70 110 60 120 35	0.5 5.000000000000

Подзадачи и система оценки

Данная задача содержит четыре подзадачи. Для оценки каждой подзадачи используется своя группа тестов. Баллы за подзадачу начисляются только в том случае, если все тесты из этой группы пройдены.

Подзадача 1 (20 баллов)

$2 \leq n \leq 50$, $0 \leq x_i \leq 1000$, $0 \leq v_i \leq 1000$. Гарантируется, что существует ответ, в котором t – целое число, не превышающее 1000.

Подзадача 2 (20 баллов)

$2 \leq n \leq 200$.

Подзадача 3 (30 баллов)

$$2 \leq n \leq 2000.$$

Подзадача 4 (30 баллов)

$$2 \leq n \leq 10^5.$$

Обратная связь

В течение тура можно не более 10 раз запросить результаты работы своей программы на тестах жюри. Запрос можно делать не чаще одного раза в 5 минут. Для каждого теста сообщается результат запуска программы на этом тесте.

В этой задаче можно выбрать, какое решение будет оцениваться. В этом случае баллы начисляются за лучшее решение из следующих:

- выбранного явно;
- последнего принятого на проверку решения.

Если выбор не сделан, то будет оцениваться лучшее решение из следующих:

- решений, по которым просмотрены баллы;
- последнего принятого на проверку решения.

Задача 7. «Сад пермского периода»

Имя входного файла:

garden.in

Имя выходного файла:

garden.out

Максимальное время работы на одном тесте:

1 секунда

Максимальный объем используемой памяти:

256 мегабайт

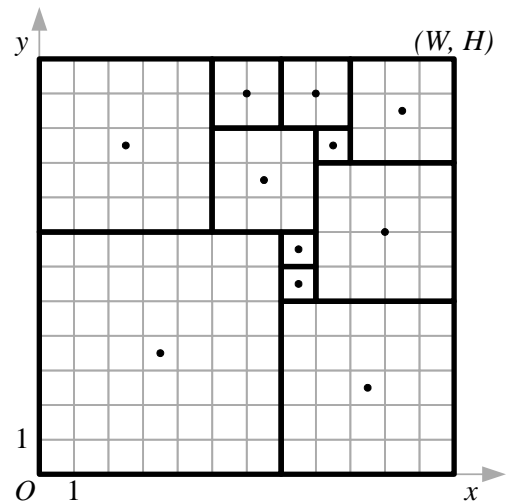
Максимальная оценка

100 баллов

Оранжерея «Сад пермского периода» представляет собой прямоугольный участок для выращивания растений пермского периода. Оранжерея была разбита дорожками на квадраты. В центре каждого квадрата посажено одно растение. Размер квадрата зависит от корневой системы растения.

За год дорожки заросли травой, что затруднило уход за оранжереями. Чтобы при садовых работах не повредить корневую систему какого-либо растения, по имеющемуся расположению растений необходимо восстановить размеры соответствующих им квадратов.

Введем декартову прямоугольную систему координат, начало которой совмещено с левым нижним углом оранжереи. Ось Ox направлена вдоль нижней границы участка, ось Oy – вдоль левой. Изначально дорожки были проложены параллельно осям координат. Единичный отрезок удалось выбрать так, что координаты углов каждого из квадратов оказались *целыми*.



Требуется написать программу, которая по размеру оранжереи и координатам растений определит размеры соответствующих им квадратов.

Формат входных данных

В первой строке входного файла записаны три натуральных числа: W – ширина оранжереи, H – длина оранжереи и N – количество посаженных растений. В каждой из следующих N строк расположены по два числа: x_i, y_i – координаты i -го растения ($0 < x_i < W, 0 < y_i < H$). Гарантируется, что соответствующие растениям квадраты имеют целую длину стороны и покрывают всю оранжерею.

Формат выходных данных

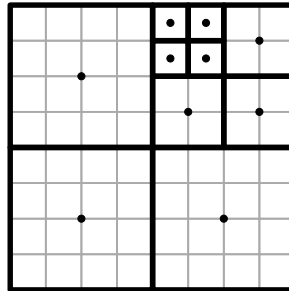
В выходной файл необходимо вывести N целых чисел – размеры квадратов, соответствующих растениям. Числа требуется вывести в порядке описания растений во входном файле.

Примеры

garden.in	garden.out
4 6 3	2
1 1	2
3 1	4
2 4	
8 8 10	1
4.5 7.5	1
5.5 7.5	4
2 6	1
4.5 6.5	2
7 7	2
5 5	4
6 2	2
7 5	4
2 2	1
5.5 6.5	

Комментарий

Оранжерея во втором примере соответствует следующему рисунку:



Подзадачи и система оценки

Данная задача содержит пять подзадач.

Подзадача 1 (10 баллов)

$W = H \leq 10$; $N \leq 6$. Для оценки данной подзадачи используется соответствующая группа тестов. Баллы за подзадачу начисляются только в том случае, если все тесты из этой группы пройдены.

Подзадача 2 (20 баллов)

$W, H \leq 20$; $N \leq 20$.

Для оценки данной подзадачи используется соответствующая группа тестов. Баллы за подзадачу начисляются только в том случае, если все тесты из этой группы пройдены.

Подзадача 3 (30 баллов)

$W, H \leq 1000$; $N \leq 1000$.

Для оценки данной подзадачи используется соответствующая группа тестов. Баллы за подзадачу начисляются только в том случае, если все тесты из этой группы пройдены.

Подзадача 4 (20 баллов)

$W, H \leq 2 \times 10^5$; $N \leq 2 \times 10^5$. Каждый тест для данной подзадачи оценивается отдельно.

Подзадача 5 (20 баллов)

$W, H \leq 10^{12}$; $N \leq 2 \times 10^5$. Каждый тест для данной подзадачи оценивается отдельно.

Обратная связь

В течение тура можно не более 10 раз запросить баллы, которые набирает программа на тестах жюри. Запрос можно делать не чаще одного раза в 5 минут. Для каждой подзадачи сообщаются результаты проверки на каждом тесте.

В этой задаче можно выбрать, какое решение будет оцениваться. В этом случае баллы начисляются за лучшее решение из следующих:

- выбранного явно;
- последнего принятого на проверку решения.

Если выбор не сделан, то будет оцениваться лучшее решение из следующих:

- тех решений, по которым просмотрены баллы;
- последнего принятого на проверку решения.

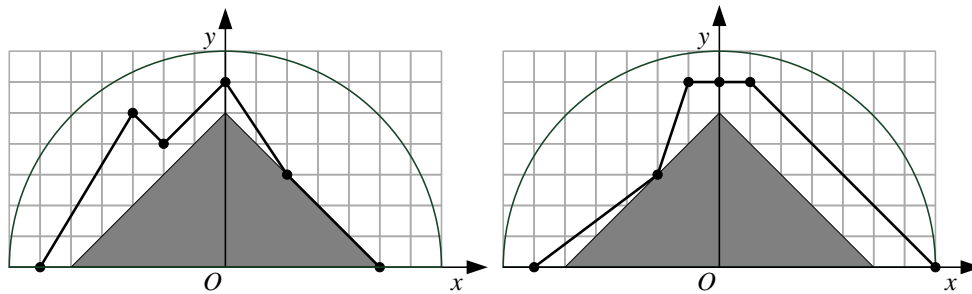
Задача 8. «Распил бревен»

Имя входного файла:	raspil.in
Имя выходного файла:	raspil.out
Максимальное время работы на одном тесте:	2 секунды
Максимальный объем используемой памяти:	256 мегабайт
Максимальная оценка	100 баллов

Лесопильный комбинат выполняет заказ на распил брусьев для строительства детского городка. Все готовые брусья должны иметь форму треугольных призм, основаниями которых являются равнобедренные треугольники. Для изготовления брусьев закуплены заготовки в виде половинок продольно распиленных бревен. Заготовки не являются идеальными половинками цилиндров, поэтому при изготовлении бруса необходимо учитывать форму заготовок. Комбинат заинтересован в изготовлении бруса с наибольшей возможной площадью поперечного сечения.

Для каждой заготовки измеряется несколько сечений. Каждое из них задано в виде ломаной, представленной координатами ее вершин $(x_0, y_0), (x_1, y_1), \dots, (x_N, y_N)$ в порядке их следования. Координаты вершин ломанной удовлетворяют следующим условиям:

- $x_0 < x_1 < x_2 < \dots < x_N$;
- $x_i = 0$ для некоторого $0 < i < N$;
- $y_0 = y_N = 0$
- для всех i от 1 до $(N - 1)$ выполнено условие $y_i > 0$.



С учетом описанных требований необходимо найти максимально возможную площадь равнобедренного треугольника, удовлетворяющего следующим условиям:

- основание треугольника лежит на оси абсцисс;
- основание симметрично относительно начала координат;
- треугольник полностью лежит внутри каждого из измеренных сечений заготовки.

Требуется написать программу, которая по заданным сечениям заготовки вычислит максимально возможную площадь искомого равнобедренного треугольника.

Формат входных данных

Первая строка входного файла содержит целое число K – количество измеренных сечений.

Далее следуют описания каждого из K сечений. В первой строке описания сечения содержится число N_k – количество звеньев ломаной. За ней следуют $(N_k + 1)$ строк, каждая из которых содержит пару целых чисел x_i и y_i – координаты вершин ломаной сечения в порядке их следования.

Формат выходных данных

Выходной файл должен содержать одно вещественное число – наибольшую возможную площадь треугольника. Эта площадь должна иметь абсолютную или относительную погрешность не более 10^{-6} , что означает следующее. Пусть выведенное число равно x , а в правильном ответе оно равно y . Ответ будет считаться правильным, если значение выражения $|x - y| / \max\{1, |y|\}$ не превышает 10^{-6} .

Обратная связь

В течение тура можно не более 10 раз запросить результаты работы своей программы на тестах жюри. Запрос можно делать не чаще одного раза в 5 минут. Для каждого теста сообщается результат запуска программы на этом тесте.

В этой задаче можно выбрать, какое решение будет оцениваться. В этом случае баллы начисляются за лучшее решение из следующих:

- выбранного явно;
- последнего принятого на проверку решения.

Если выбор не сделан, то будет оцениваться лучшее решение из следующих:

- решений, по которым просмотрены баллы;
- последнего принятого на проверку решения.

Пример

<code>raspil.in</code>	<code>raspil.out</code>
2	25.0
5	
-6 0	
-3 5	
-2 4	
0 6	
2 3	
5 0	
5	
-6 0	
-2 3	
-1 6	
0 6	
1 6	
7 0	

Подзадачи и система оценки

Данная задача содержит пять подзадач.

Подзадача 1 (20 баллов)

$K = 1$, $N_1 \leq 15$, координаты вершин по модулю не превышают 20.

Для оценки данной подзадачи используется соответствующая группа тестов. Баллы за подзадачу начисляются только в том случае, если все тесты из этой группы пройдены.

Подзадача 2 (10 баллов)

$1 \leq K \leq 20$, сумма $N_i \leq 2000$, координаты вершин по модулю не превышают 10^4 . Гарантируется, что полученный в качестве ответа треугольник является прямоугольным.

Для оценки данной подзадачи используется соответствующая группа тестов. Баллы за подзадачу начисляются только в том случае, если все тесты из этой группы пройдены.

Подзадача 3 (20 баллов)

$1 \leq K \leq 20$, сумма $N_i \leq 2000$, координаты вершин по модулю не превышают 10^4 .

Для оценки данной подзадачи используется соответствующая группа тестов. Баллы за подзадачу начисляются только в том случае, если все тесты из этой группы пройдены.

Подзадача 4 (10 баллов)

$1 \leq K \leq 1000$, сумма $N_i \leq 10^5$, координаты вершин по модулю не превышают 10^9 . Гарантируется, что полученный в качестве ответа треугольник является прямоугольным.

Для оценки данной подзадачи используется соответствующая группа тестов. Баллы за подзадачу начисляются только в том случае, если все тесты из этой группы пройдены.

Подзадача 5 (40 баллов)

$1 \leq K \leq 1000$, сумма $N_i \leq 10^5$, координаты вершин по модулю не превышают 10^9 .

Каждый тест для данной подзадачи оценивается отдельно.