

Задача X. «Сумма»

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

Заданы два целых числа. Требуется найти их сумму.

Формат входных данных

Входной файл содержит два целых числа a и b , по одному на строке. Файл не содержит пробелов.

Формат выходных данных

Выведите в выходной файл сумму чисел, заданных во входном файле.

Пример

sum.in	sum.out
2	5
3	

Подзадачи и система оценки

В данной задаче четыре подзадачи. Для оценки каждой подзадачи используется группа тестов. Вы получите баллы за подзадачу только в том случае, если все тесты из соответствующей группы пройдены.

Подзадача 1 (25 баллов)

Заданные числа a и b неотрицательные и не превышают 10^9 .

Подзадача 2 (25 баллов)

Заданные числа a и b по модулю не превышают 10^9 .

Подзадача 3 (25 баллов)

Заданные числа a и b неотрицательные и не превышают 10^{100} .

Подзадача 4 (25 баллов)

Заданные числа a и b по модулю не превышают 10^{100} .

Обратная связь

Вы можете запросить баллы, которые получает ваша программа на тестах жюри для этой задачи не более 10 раз за тур. Вы можете делать запрос не чаще одного раза в 5 минут. Для каждой подзадачи вам сообщаются баллы за эту подзадачу. Детальные результаты на тестах не сообщаются.

В этой задаче вы можете вручную выбрать, какое решение будет оцениваться. В этом случае вы получите баллы за лучшее решение из:

- выбранного вами;
- последнего принятого на проверку решения.

Если вы не сделаете свой выбор, то будет оцениваться лучшее решение из:

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

Задача Y. «Угадай число»

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Максимальное время работы на одном тесте:	<i>2 секунды</i>
Максимальный объем используемой памяти:	<i>256 мегабайт</i>
Максимальная оценка	<i>100 баллов</i>

Это интерактивная задача. В процессе тестирования ваша программа будет взаимодействовать с программой жюри с использованием стандартных потоков ввода/вывода.

Программа жюри загадала число от 1 до n , цель вашей программы — отгадать его. Для этого ваша программа сообщает свои догадки программе жюри, а программа жюри отвечает, является ли загаданное число большим, меньшим или равным сделанной догадке.

Протокол взаимодействия с программой жюри

Сначала ваша программа должна прочитать из стандартного потока ввода число n . Затем протокол общения следующий: ваша программа выводит в стандартный поток вывода одну строку, содержащую число — свою догадку о загаданном числе. Делайте сброс буфера потока вывода после каждой догадки. Для этого используйте

- `flush(output)` в паскале или Delphi;
- `fflush(stdout)` или `cout.flush()` в C/C++;
- `Console.out.flush()` в Visual Basic.

После этого программа должна считать из стандартного потока ввода одно число: ответ программы жюри. Возможны следующие ответы:

- 1 — загаданное число больше последней догадки;
- -1 — загаданное число меньше последней догадки;
- 0 — последняя догадка верна. Считав 0, ваша программа должна завершиться.

Пример взаимодействия

стандартный ввод	стандартный вывод
5	3
-1	1
1	2
0	

Подзадачи и система оценки

В данной задаче две подзадачи. Для оценки каждой подзадачи используется группа тестов. Вы получите баллы за подзадачу только в том случае, если все тесты из соответствующей группы пройдены.

Подзадача 1 (50 баллов)

Выполнено неравенство $1 \leq n \leq 100$. Ваша программа должна сделать не более n догадок.

Подзадача 2 (50 баллов)

Выполнено неравенство $100 \leq n \leq 10^9$. Ваша программа должна сделать не более 30 догадок.

Обратная связь

Вы можете запросить результаты работы вашей программы на тестах жюри для этой задачи 10 раз за тур. Вы можете делать запрос не чаще одного раза в 5 минут. По каждому тесту вам сообщается результат запуска вашей программы на этом тесте.

В этой задаче вы можете вручную выбрать, какое решение будет оцениваться. В этом случае вы получите баллы за лучшее решение из:

- выбранного вами;
- последнего принятого на проверку решения.

Если вы не сделаете свой выбор, то будет оцениваться лучшее решение из:

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

Задача Z. «Разворот последовательности»

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

Задана последовательность целых чисел. Требуется вывести эту последовательность в обратном порядке.

Формат входных данных

Входной файл содержит последовательность целых чисел: $a_1, a_2, \dots, a_n$ ($-10^{18} \leq a_i \leq 10^{18}$). Количество элементов последовательности не превышает 100 000.

Формат выходных данных

Выведите в выходной файл последовательность чисел из входного файла в обратном порядке.

Примеры

<code>sequence.in</code>	<code>sequence.out</code>
1 2 3	3 2 1
-3 4 6 -8 9	9 -8 6 4 -3

Система оценки

Для оценки этой задачи используется 20 тестов. Каждый тест оценивается по 5 баллов. Тесты оцениваются независимо.

Обратная связь

В этой задаче вам не сообщаются результаты работы вашей программы на тестах жюри во время тура.

В этой задаче вы можете вручную выбрать, какое решение будет оцениваться. В этом случае вы получите баллы за лучшее решение из:

- выбранного вами;
- последнего принятого на проверку решения.

Если вы не сделаете свой выбор, то будет оцениваться последнее принятое на проверку решение.

Задача 1. «Скобки»

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

Юная программистка Агнесса недавно узнала на уроке информатики об арифметических выражениях. Она заинтересовалась вопросом, что случится, если из арифметического выражения удалить всё, кроме скобок. Введя запрос в своём любимом поисковике, она выяснила, что математики называют последовательности скобок, которые могли бы встречаться в некотором арифметическом выражении, *правильными скобочными последовательностями*.

Так, последовательность $()()$ является правильной скобочной последовательностью, потому что она может, например, встречаться в выражении $(2+2):(3-(5-2)+4)$, а последовательности $()()$ и $()()$ не являются таковыми. Легко видеть, что существует пять правильных скобочных последовательностей, состоящих ровно из шести скобок (по три скобки каждого типа — открывающих и закрывающих): $((()))$, $((())())$, $(())(())$, $(())(())$ и $(())(())$.

Агнесса заинтересовалась простейшими преобразованиями правильных скобочных последовательностей. Для начала Агнесса решила ограничиться добавлением скобок в последовательность. Она очень быстро выяснила, что после добавления одной скобки последовательность перестаёт быть правильной, а вот добавление двух скобок иногда сохраняет свойство правильности. Например, при добавлении двух скобок в различные места последовательности $()()$ можно получить последовательности $((())())$, $((())())$, $(())(())$ и $(())(())$. Легко видеть, что при любом способе добавления двух скобок с сохранением свойства правильности одна из новых скобок должна быть открывающей, а другая — закрывающей.

Агнесса хочет подсчитать количество различных способов добавления двух скобок в заданную правильную скобочную последовательность так, чтобы снова получилась правильная скобочная последовательность. К сожалению, выяснилось, что это количество может быть в некоторых случаях очень большим. Агнесса различает способы получения последовательности по позициям добавленных скобок в полученной последовательности. Например, даже при добавлении скобок в простейшую последовательность $()$ можно получить другую правильную скобочную последовательность семью способами: $((()))$, $((())())$, $((())())$, $((())())$, $((())())$, $((())())$, $((())())$. Здесь добавленные скобки выделены жирным шрифтом.

Таким образом, если в полученной последовательности добавленная открывающая скобка стоит в позиции i , а добавленная закрывающая — в позиции j , то два способа, соответствующие парам (i_1, j_1) и (i_2, j_2) , считаются различными, если $i_1 \neq i_2$ или $j_1 \neq j_2$.

Требуется написать программу, которая по заданной правильной скобочной последовательности определяет количество различных описанных выше способов добавления двух скобок.

Формат входных данных

Входной файл состоит из одной непустой строки, содержащей ровно $2n$ символов: n открывающих и n закрывающих круглых скобок. Гарантируется, что эта строка является правильной скобочной последовательностью.

Формат выходных данных

Выведите в выходной файл количество различных способов добавления в заданную последовательность двух скобок таким образом, чтобы получилась другая правильная скобочная последовательность.

Примеры

brackets.in	brackets.out
$()$	7
$()()$	17
$((()))$	21

Подзадачи и система оценки

Данная задача содержит три 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 (40 баллов)

Величина n (количество скобок каждого типа) не превосходит 50.

Подзадача 2 (30 баллов)

Величина n (количество скобок каждого типа) не превосходит 2500.

Подзадача 3 (30 баллов)

Величина n (количество скобок каждого типа) не превосходит 50 000.

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 2. «Школа олимпийского резерва»

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

Для подготовки к чемпионату мира по футболу 2018 года создается школа олимпийского резерва. В нее нужно зачислить M юношей 1994–1996 годов рождения. По результатам тестирования каждому из N претендентов был выставлен определенный балл, характеризующий его мастерство. Все претенденты набрали различные баллы. В составе школы олимпийского резерва хотелось бы иметь A учащихся 1994 г.р., B – 1995 г.р. и C – 1996 г.р. ($A + B + C = M$). При этом минимальный балл зачисленного юноши 1994 г.р. должен быть больше, чем минимальный балл зачисленного 1995 г.р., а минимальный балл зачисленного 1995 г.р. должен быть больше, чем минимальный балл зачисленного 1996 г.р. Все претенденты, набравшие балл больше минимального балла для юношей своего года рождения, также должны быть зачислены.

В базе данных для каждого претендента записаны год его рождения и тестовый балл. Требуется определить, сколько нужно зачислить юношей каждого года рождения M_{94} , M_{95} и M_{96} ($M_{94} + M_{95} + M_{96} = M$), чтобы значение величины $F = |M_{94} - A| + |M_{95} - B| + |M_{96} - C|$ было минимально, все правила, касающиеся минимальных баллов зачисленных, были соблюдены, и должен быть зачислен хотя бы один юноша каждого требуемого года рождения.

Формат входных данных

В первой строке входного файла находится число K – количество наборов входных данных. Далее следуют описания каждого из наборов. В начале каждого набора расположены три натуральных числа A , B , C . Во второй строке описания находится число N – количество претендентов (гарантируется, что $N \geq A + B + C$). В каждой из следующих N строк набора содержатся два натуральных числа – год рождения (число 1994, 1995 или 1996 соответственно) и тестовый балл очередного претендента.

Формат выходных данных

Ответ на каждый тестовый набор выводится в отдельной строке. Если хотя бы одно из требований выполнить невозможно, то в качестве ответа следует вывести только число -1 . В противном случае соответствующая строка сначала должна содержать минимальное значение величины F , а затем три числа M_{94} , M_{95} и M_{96} , на которых это минимальное значение достигается, удовлетворяющие всем требованиям отбора. Если искомым вариантов несколько, то разрешается выводить любой из них.

Примеры

<code>school.in</code>	<code>school.out</code>
3	-1
1 1 1	0 1 1 1
4	-1
1994 3	
1994 4	
1996 1	
1996 2	
1 1 1	
3	
1995 2	
1994 3	
1996 1	
1 1 1	
3	
1994 1	
1995 2	
1996 3	

1	2 3 2 1
2 3 1	
7	
1996 2	
1994 7	
1994 4	
1996 1	
1995 3	
1994 5	
1995 6	

Комментарий

В первом примере на первом наборе ответ не существует, потому что нельзя пригласить хотя бы одного юношу 1995 г.р. Во втором наборе ответ существует и единственный, в третьем – нельзя выполнить правило относительно минимальных баллов.

Во втором примере правильным является также ответ 2 2 2 2.

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 (25 баллов)

$K = 1$; $N \leq 100$; каждый претендент характеризуется своим баллом от 1 до N .

Подзадача 2 (25 баллов)

Сумма значений N по всем тестовым наборам не превосходит 10 000, каждый претендент характеризуется своим баллом от 1 до 10^9 .

Подзадача 3 (25 баллов)

Сумма значений N по всем тестовым наборам не превосходит 100 000, каждый претендент характеризуется своим баллом от 1 до N .

Подзадача 4 (25 баллов)

Сумма значений N по всем тестовым наборам не превосходит 300 000, каждый претендент характеризуется своим баллом в диапазоне от 1 до 10^9 .

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 баллы за эту подзадачу. Детальные результаты на тестах не сообщаются.

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.

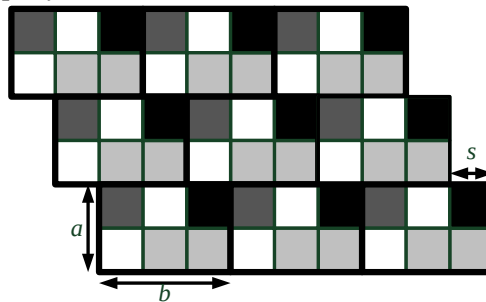
Задача 3. «Снова в космос»

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

К 50-летию первого пилотируемого полета в космос решено создать новый тип космического корабля многоразового использования “Восторг”. Прямоугольная часть его корпуса (далее *прямоугольник*) должна быть облицована квадратными термозащитными плитками разных цветов одного и того же размера. Прямоугольник состоит из r рядов по c плиток в каждом. Плитки должны образовывать заданный рисунок.

Облицовка космического корабля отдельными плитками очень трудоемка, поэтому для выкладывания заданного рисунка используются одинаковые прямоугольные панели, состоящие из плиток. Панели крепятся на корпусе одна за другой, заполняя ряд за рядом сверху вниз. Каждый ряд панелей может быть сдвинут относительно предыдущего на одно и то же число плиток. При этом панели могут выходить за пределы прямоугольника. Панели должны быть одинаково ориентированы, то есть при параллельном переносе одной панели на место другой цвета образующих эти панели плиток должны совпадать.

Главный конструктор хочет выбрать такой размер панели $a \times b$ и сдвиг s , чтобы этими панелями можно было выложить заданный рисунок, и площадь панели была минимальна.



Пример панелей с $a = 2$, $b = 3$, $s = 1$.

Требуется написать программу, которая по заданному расположению плиток в прямоугольнике рассчитывает размеры минимальной по площади панели, которую можно использовать при его облицовке, а также величину сдвига вправо ($0 \leq s < b$) каждого следующего ряда относительно предыдущего.

Формат входных данных

Первая строка входного файла содержит два целых числа: r и c – размеры прямоугольника в плитках. В последующих r строках указаны цвета плиток фрагмента. Каждый из $k \leq 26$ цветов обозначен одной из первых k прописных букв латинского алфавита. Гарантируется, что для этого прямоугольника можно подобрать панель размера $a \times b$, такую, что $2a \leq r$ и $2b \leq c$.

Формат выходных данных

В выходной файл необходимо вывести три целых числа a , b и s , удовлетворяющих условиям задачи. Если решений несколько, разрешается вывести любое из них.

Пример

shuttle.in	shuttle.out
2 4 ABAB ABAB	1 2 0
5 7 DCADCAD BABBBAB ADCADCA BBABBBAB CADCADC	2 3 1

Комментарий

Во втором примере облицовка прямоугольника соответствуют следующему рисунку (выступающие за границы прямоугольника части панелей не показаны):

D	C	A	D	C	A	D
B	A	B	B	A	B	B
A	D	C	A	D	C	A
B	B	A	B	B	A	B
C	A	D	C	A	D	C

Подзадачи и система оценки

Данная задача содержит семь 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 (10 баллов)

В правильном ответе величина сдвига s равна нулю, r и c не превосходят 20.

Подзадача 2 (15 баллов)

В правильном ответе величина сдвига s равна нулю, r и c не превосходят 200.

Подзадача 3 (20 баллов)

В правильном ответе величина сдвига s равна нулю, r и c не превосходят 1961.

Подзадача 4 (10 баллов)

Величина сдвига s произвольна, r и c не превосходят 20.

Подзадача 5 (15 баллов)

Величина сдвига s произвольна, r и c не превосходят 200.

Подзадача 6 (15 баллов)

Величина сдвига s произвольна, r и c не превосходят 500.

Подзадача 7 (15 баллов)

Величина сдвига s произвольна, r и c не превосходят 1961.

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 баллы за эту подзадачу. Детальные результаты на тестах не сообщаются.

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 4. «Почта»

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

В городе π планируется проведение реформы доставки почты. Вместо большого количества устаревших автомобилей приобретён специализированный Е-мобиль, задача которого — развозить всю корреспонденцию по почтовым отделениям города.

ИТ-отдел городской почты разработал кольцевой маршрут, проходящий через все почтовые отделения. Все улицы в городе π с односторонним движением. Необходимо выбрать почтовое отделение для размещения логистического центра, куда будет поступать вся городская корреспонденция перед её отправкой на Е-мобиле по маршруту. Из-за пробок скорость движения на участках маршрута между почтовыми отделениями зависит от времени суток. Размещение логистического центра считается оптимальным, если после отъезда из него в нулевой момент времени Е-мобиль развезёт корреспонденцию и возвратится в логистический центр как можно раньше. Время разгрузки корреспонденции пренебрежимо мало.

Требуется написать программу, которая по заданному маршруту с учетом скорости движения Е-мобиля определяет оптимальное расположение логистического центра и наиболее ранний возможный момент возврата в логистический центр.

Формат входных данных

В первой строке задаётся целое положительное число N — количество почтовых отделений в городе π . Почтовые отделения нумеруются в порядке их следования по маршруту, начиная с единицы.

В следующих строках описаны N участков маршрута между почтовыми отделениями. Каждое описание содержит три строки:

- в первой строке описания задаётся целое положительное число d_i — длина данного участка ($d_i \leq 10^9$), а также целое неотрицательное число E_i — количество отрезков времени, в течение каждого из которых скорость движения Е-мобиля постоянна;
- во второй строке даны целые положительные числа $t_{i,j}$ ($1 \leq j \leq E_i$, $0 < t_{i,1} < \dots < t_{i,E_i} \leq 10^9$) — значения моментов времени, в которые изменяется скорость движения. Если соответствующее E_i равно нулю, то эта строка — пустая;
- в третьей строке целые положительные числа v_j — скорости на полуинтервале времени $[t_{i,j-1}, t_{i,j})$, где $1 \leq j \leq E_i + 1$, считается, что $t_{i,0} = 0$, $t_{i,E_i+1} = +\infty$.

Все числа в строках разделяются одним пробелом.

Формат выходных данных

В единственной строке необходимо вывести номер почтового отделения, в котором нужно разместить логистический центр, а также время проезда Е-мобиля по маршруту.

Ответ должен иметь абсолютную или относительную погрешность не более 10^{-9} , что означает следующее. Пусть максимальное расстояние от выведенной точки до некоторой трассы равно x , а в правильном ответе оно равно y . Ответ будет засчитан, если значение выражения $|x - y| / \max\{1, |y|\}$ не превышает 10^{-9} .

Примеры

post.in	post.out
2 3 2 1 1 2 4 2 2 3 1	2 2.833333
2 2 1 2 2 1 2	1 2.000000

Подзадачи и система оценки

В данной задаче три подзадачи.

Подзадача 1 (20 баллов)

В тестах первого набора количество почтовых отделений $N \leq 1000$, $N \leq \sum_{i=1}^N E_i \leq 1000$. Баллы за первую подзадачу начисляются только в том случае, если все тесты для этой подзадачи пройдены.

Подзадача 2 (20 баллов)

В тестах второго набора количество почтовых отделений $N \leq 4000$, $N \leq \sum_{i=1}^N E_i \leq 10^5$. Баллы за вторую подзадачу начисляются только в том случае, если все тесты для этой подзадачи пройдены.

Подзадача 3 (60 баллов)

В тестах третьего набора количество почтовых отделений $N \leq 10^5$, $N \leq \sum_{i=1}^N E_i \leq 3 \cdot 10^5$.

Каждый тест для третьей 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 баллы за эту подзадачу. Детальные результаты на тестах не сообщаются.

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 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

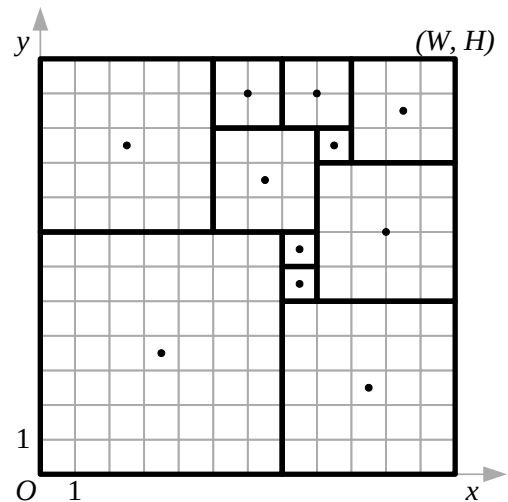
Максимальная оценка

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 оранжереей. 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

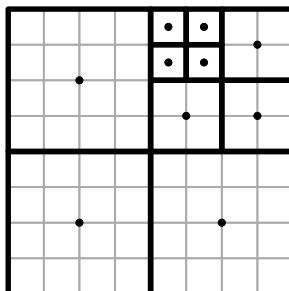
В выходной файл необходимо вывести N целых чисел – размеры квадратов, соответствующих растениям. Числа требуется вывести в порядке описания растений во входном файле.

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.

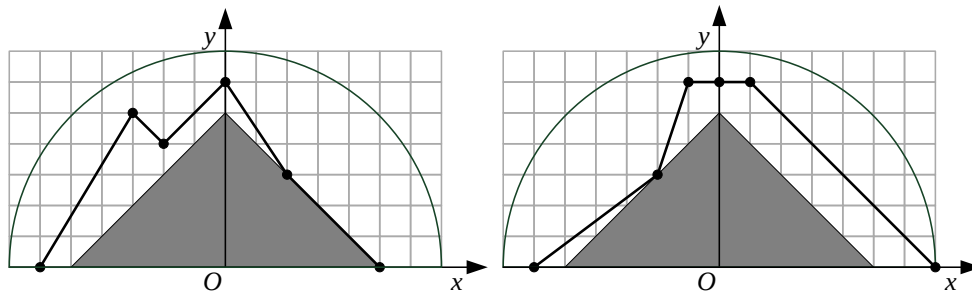
Задача 8. «Распил бревен»

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

raspil.in	raspil.out
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.