

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:	school.in
Имя выходного файла:	school.out
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 вывести любой из них.

Примеры

school.in	school.out
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.

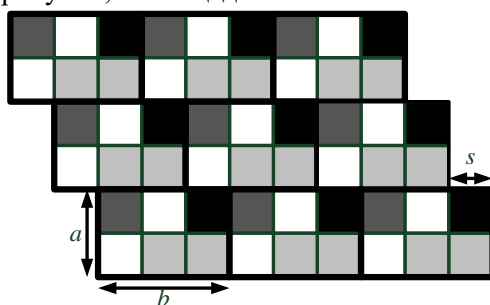
Задача 3. «Снова в космос»

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 BBABBAB 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 — развезти всю корреспонденцию по почтовым отделениям города.

IT-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.