

<i>Предмет</i>	<i>Класс</i>	<i>Дата</i>	<i>Время начала</i>	<i>Время окончания</i>
<i>Информатика и ИКТ</i>	<i>9-11</i>	<i>9.12.2022</i>	<i>11-00</i>	<i>14-55</i>

Задача 1. Почти анаграмма

<i>Входные данные:</i>	<i>стандартный ввод</i>
<i>Выходные данные:</i>	<i>стандартный вывод</i>
<i>Ограничение по памяти:</i>	<i>256 МБ</i>
<i>Ограничение по времени:</i>	<i>1 секунда на тест</i>
<i>Максимальная оценка за задачу:</i>	<i>100 баллов</i>

Слово является «почти анаграммой» другого слова, если в одном из слов добавить или убрать ровно одну букву, то слова станут анаграммами. Анаграмма – слово, полученное перестановкой букв, из первоначального слова.

Напишите программу, которая проверяет являются ли слова «почти анаграммами».

Входные данные

В первой строке записано первое слово.

Во второй строке записано второе слово.

Слова состоят из маленьких букв латинского алфавита. Длина каждого слова не превышает 256 символов.

Выходные данные

В первой строке выведите YES или NO. Во второй строке, что нужно сделать с первым словом, если ответ YES:

Во второй строке, что нужно сделать с первым словом, если ответ YES:

«+» новая буква – если в первое слово нужно добавить букву;

«-» удаляемая буква – если из первого слова нужно убрать букву;

«0» – если два слова уже анаграммы друг друга.

В третьей строке нужно вывести добавляемую или удаляемую букву.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
sbsbc ssbb	YES - c
bbs bbb	NO

Комментарий

В первом тесте из примера можно убрать в первой строке букву «с», и она станет анаграммой второй строки.

Оценивание

Решения, корректно работающие для случаев, когда количество различных букв в словах не более трех, будут набирать не менее 30 баллов.

<i>Предмет</i>	<i>Класс</i>	<i>Дата</i>	<i>Время начала</i>	<i>Время окончания</i>
<i>Информатика и ИКТ</i>	<i>9-11</i>	<i>9.12.2022</i>	<i>11-00</i>	<i>14-55</i>

Задача 2. Прогулка

<i>Входные данные:</i>	<i>стандартный ввод</i>
<i>Выходные данные:</i>	<i>стандартный вывод</i>
<i>Ограничение по памяти:</i>	<i>256 МБ</i>
<i>Ограничение по времени:</i>	<i>1 секунда на тест</i>
<i>Максимальная оценка за задачу:</i>	<i>100 баллов</i>

Пешеход идет по дорожке, перешагивая с одной плитки на следующую. На дорожке есть места, где вместо плиток – глубокие ямы. Пешеход один раз может сделать прыжок фиксированной длины и дальше опять идет мелкими шажками. Прыжок через последовательность только целых плиток пешеход делать не будет. За последней плиткой дорожки находится очень широкая лужа, которую пешеход не может никак перепрыгнуть и попадать в неё он тоже не хочет.

Пешеходу стало интересно узнать, сколько есть способов добраться из первой плитки дорожки на последнюю без падений в ямы и лужу. Способы отличаются последовательностью шагов и прыжка. Гарантируется, что на первой плитке ямы нет.

Напишите программу, которая подсчитывает количество способов.

Входные данные

В первой строке записано одно целое число N ($0 < N \leq 100$), N — длина дорожки в плитках. Плитки нумеруются от 1 до N . Номер первой плитки на дорожке – 1, последней – N .

Во второй строке записано одно целое число d ($0 < d \leq 100$), d — длина прыжка, через сколько плиток можно перепрыгнуть. (Если пешеход с плитки с номером 1 сделает прыжок, то он приземлится на плитку с номером $d+2$.)

В третьей строке записано одно целое число M ($0 \leq M \leq N$), M — количество ям на дорожке.

В каждой из следующих M строк содержится по одной цифре — номера плиток с ямами.

Выходные данные

Необходимо вывести целое число S — количество способов добраться из начальной точки дорожки в конечную.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
6 3 2 4 3	2

Оценивание

Решения, корректно работающие для случаев, когда ям не более одной, будут набирать не менее 15 баллов.

Решения, корректно работающие для случаев, когда расстояние от плиток с ямами до последней плитки более d , будут набирать не менее 40 баллов.

<i>Предмет</i>	<i>Класс</i>	<i>Дата</i>	<i>Время начала</i>	<i>Время окончания</i>
<i>Информатика и ИКТ</i>	<i>9-11</i>	<i>9.12.2022</i>	<i>11-00</i>	<i>14-55</i>

Задача 3. Ахиллес, черепаха и муха

<i>Входные данные:</i>	<i>стандартный ввод</i>
<i>Выходные данные:</i>	<i>стандартный вывод</i>
<i>Ограничение по памяти:</i>	<i>256 МБ</i>
<i>Ограничение по времени:</i>	<i>1 секунда на тест</i>
<i>Максимальная оценка за задачу:</i>	<i>100 баллов</i>

Однажды древнегреческий герой Ахиллес увидел черепаху на расстоянии d метров и решил подойти к ней поближе. Каждую минуту он сокращает расстояние в 2 раза до тех пор, пока черепаха удалена от него более чем на метр.

Древняя Греция – удивительное место, и существа, населяющие ее, так же удивительны. Одно из таких существ – муха, сидящая рядом с Ахиллесом. Она решила посостязаться с ним в скорости. Для этого муха решила летать от героя к черепахе и обратно до тех пор, пока Ахиллес не закончит свой путь.

Напишите программу, которая подсчитывает расстояние, которое пролетела муха.

Входные данные

В единственной строке записаны два целых числа: d – расстояние в метрах между черепахой и Ахиллесом и v – скорость мухи в метрах в секунду ($1 \leq d \leq 10^5$, $1 \leq v \leq 10^{12}$). Скорость мухи выше скорости Ахиллеса в любой момент времени.

Выходные данные

Выведите единственное число – расстояние, которое пролетела муха. Число вывести с точностью до 10^{-6} .

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 7	840.0

Замечание

Обратите внимание, что числа могут не помещаться в 32-битный тип данных, необходимо использовать 64-битный тип данных (например, **long long** в C++, **int64** в паскале, **long** в Java). При вычислениях лучше использовать вещественные типы.

Задача 4. Защитные башни

<i>Входные данные:</i>	<i>стандартный ввод</i>
<i>Выходные данные:</i>	<i>стандартный вывод</i>
<i>Ограничение по памяти:</i>	<i>256 МБ</i>
<i>Ограничение по времени:</i>	<i>2 секунды на тест</i>
<i>Максимальная оценка за задачу:</i>	<i>100 баллов</i>

Археологи по космическим снимкам нашли следы древних руин и русло засохшей реки. У археологов есть предположение, что по обе стороны реки были крепости. План крепости – прямоугольник наибольшей площади, в вершинах которого стояли башни.

Река задана уравнением прямой $ax+by+c=0$. Стены крепости должны быть параллельны осям координат.

<i>Предмет</i>	<i>Класс</i>	<i>Дата</i>	<i>Время начала</i>	<i>Время окончания</i>
<i>Информатика и ИКТ</i>	<i>9-11</i>	<i>9.12.2022</i>	<i>11-00</i>	<i>14-55</i>

Напишите программу, которая среди координат руин находит координаты башен крепостей по обе стороны реки.

Входные данные

В первой строке записано три целых числа a, b, c ($0 \leq a, b, c \leq 100$) — параметры уравнения прямой.

Во второй строке записано одно целое число N ($0 < N \leq 2000$), N — количество обнаруженных руин.

В каждой из следующих N строк содержится по два целых числа, абсциссы и ординаты очередной руины. Все координаты по модулю не превосходят 10^3 . В одной точке может находиться не более одной руины.

Выходные данные

Сначала необходимо вывести в четырех строках координаты башен одной крепости, по два целых числа в каждой строке, затем в таком же формате в последующих четырех строках выведите координаты башен второй крепости. Крепости должны быть по разные стороны реки. Если есть несколько ответов, то выведите любой. Гарантируется, что с обеих сторон реки можно найти крепости.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
1 -1 -4	1 1
1 1	1 4
6 0	4 1
1 1	4 4
2 3	6 0
4 4	7 0
1 4	7 1
6 1	6 1
2 1	
4 1	
7 1	
7 0	
4 3	

Оценивание

Решения, корректно работающие для случаев, когда река параллельна одной из осей координат, будут набирать не менее 50 баллов.

Решения, корректно работающие для случаев, когда $N \leq 500$, будут набирать не менее 30 баллов.

Задача 5. Сбор шишек

Входные данные:

стандартный ввод

Выходные данные:

стандартный вывод

Ограничение по памяти:

256 МБ

Ограничение по времени:

2 секунды на тест

Максимальная оценка за задачу:

100 баллов

<i>Предмет</i>	<i>Класс</i>	<i>Дата</i>	<i>Время начала</i>	<i>Время окончания</i>
<i>Информатика и ИКТ</i>	<i>9-11</i>	<i>9.12.2022</i>	<i>11-00</i>	<i>14-55</i>

В лесу растет N кедров, с каждого кедров можно набрать некоторое количество шишек. Медведь пошел в лес собрать шишки для Маши. У некоторых кедров Медведь шишки может не брать. Но если начинает брать, то забирает все шишки с кедров. При переходе от одного кедров к следующему усталость Медведя увеличивается на значение, равное количеству шишек в его мешке. И ему хочется набрать побольше шишек, пока усталость не достигнет критического значения K . Начальное значение усталости равно нулю.

Напишите программу, которая подсчитывает, какое максимальное количество шишек Медведь сможет набрать.

Входные данные

В первой строке записано два целых числа N и K ($0 < N \leq 10^3$, $0 < K \leq 10^8$), N — количество кедров, K — наибольшее значение усталости.

В каждой из следующих N строк содержится по одному целому числу — количество шишек у очередного кедров. Суммарное количество шишек у всех кедров не превышает 10^4 . Кедров перечислены в порядке их обхода Медведем.

Выходные данные

Необходимо вывести одно целое число — наибольшее количество шишек, которое Медведь может собрать.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
4 20 10 6 20 5	30

Комментарий

Медведь забирает 10 шишек под первым кедром. Проходит мимо второго кедров к третьему. Забирает еще 20 шишек у третьего кедров. При подходе к третьему кедров усталость становится равна 20, поэтому Медведь прекращает дальнейший сбор шишек.

Оценивание

Решения, корректно работающие при равном количестве шишек под каждым кедром, будут набирать не менее 30 баллов.

Решения, корректно работающие при $N \leq 30$, будут набирать не менее 40 баллов.