

Задача А. Киоск с напитками

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Вася открыл киоск прохладительных напитков, в котором решил продавать различные лимонады и смузи. Отличительной особенностью его киоска является то, что он полностью автоматизирован — напитки готовит и продаёт робот. В состав напитков могут входить различные ингредиенты: лёд, газировка, сиропы, фрукты, овощи и т.п.

Вася хочет написать программу, которая должна по имеющемуся в наличии на текущий момент набору ингредиентов сообщать, какое количество напитков робот может приготовить. Известно, сколько у робота было ингредиентов на начало дня, известны рецепты напитков и заказы посетителей киоска в порядке очереди. Необходимо помочь Васе написать такую программу.

Формат входных данных

В первой строке входного файла записано число M — количество различных ингредиентов, которые робот получает на начало дня ($1 \leq M \leq 500$).

В следующих M строках дается описание всех ингредиентов, по одному на каждой строке.

Описание i -го ингредиента содержит его название и число x_i — количество ингредиента на начало дня в определенных единицах измерения ($0 \leq x_i \leq 10^9$, $1 \leq i \leq M$). Название ингредиента — это последовательность строчных букв латинского алфавита, длиной не более 100 символов. Гарантируется, что все названия ингредиентов различны.

Следующая строка содержит число N — количество рецептов напитков, известных роботу ($1 \leq N \leq 100$).

Далее N блоков описывают рецепты напитков.

Каждый блок начинается строкой с названием напитка и количеством ингредиентов r_j , входящих в рецепт напитка ($1 \leq r_j \leq M$, $1 \leq j \leq N$). Название напитка — это последовательность строчных букв латинского алфавита, длиной не более 100 символов.

Затем каждая из r_j строк содержит название ингредиента и его необходимое количество v_k для изготовления напитка ($1 \leq v_k \leq 10^9$, $1 \leq k \leq r_j$). Гарантируется, что в рецептах содержатся только те названия ингредиентов, которые известны роботу, и ингредиенты в одном рецепте не повторяются.

В следующей строке задано число K — количество заказов, поступивших роботу в течение дня ($1 \leq K \leq 100$).

Следующие K строк содержат по одному названию напитка, которые посетители заказали в киоске в течение дня в порядке их прихода. Гарантируется, что названия совпадают с названиями напитков, известных роботу, а также никакой посетитель не мог заказать напиток, который нельзя было приготовить на момент заказа.

Формат выходных данных

Выходной файл должен содержать ровно $K + 1$ строку. В каждую строку необходимо записать по одному числу — количеству различных видов напитков, которые мог бы приготовить робот перед тем, как пришёл очередной заказ, в порядке перечисления заказов, указанных во входном файле. Последняя ($K + 1$)-я строка должна содержать количество видов напитков, которые мог бы приготовить робот в конце дня, после последнего заказа.

Пример

input.txt	output.txt
5	2
lemon 7	2
water 51	1
syrup 6	0
sugar 28	
mint 17	
2	
lemonade 3	
lemon 3	
water 3	
sugar 1	
lemonmint 4	
mint 1	
lemon 2	
water 3	
syrup 2	
3	
lemonade	
lemonmint	
lemonmint	

Задача В. Магический дождь

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Маг Иннокентий путешествует в поисках редких ингредиентов для своих зелий. В настоящее время он находится в гористой местности, через которую проложен тракт. С этого тракта невозможно свернуть. Вдоль тракта расположено несколько деревень, каждая из которых находится на некоторой высоте над уровнем моря. Начало и конец тракта находятся на высоте 0.

Местные жители выращивают полезные растения, необходимые Иннокентию. Поля простираются вдоль тракта между соседними селениями. К несчастью, из-за засухи весь урожай может погибнуть. Иннокентий может призвать дождевую тучу несколько раз, поливая все участки тракта между деревнями $x - r$ и $x + r$, где x — номер поселения или конец тракта, в котором маг призвал тучу, а r — размер тучи.

Проблема в том, что для полного орошения участка, его нужно полить не менее T раз, а Иннокентий не хочет вызывать тучу в каждой деревне более одного раза. Помогите ему оросить наибольшую суммарную длину участков.

Формат входных данных

В первой строке входного файла даны четыре числа N , K , r и T — количество деревень, количество туч, которые может призвать Иннокентий, размер тучи и количество раз, которые необходимо полить участок для орошения ($1 \leq K \leq N \leq 200$, $1 \leq r \leq 5$, $1 \leq T \leq 2 \cdot r$).

Во второй строке даны N целых чисел y_i — высоты, на которых находятся деревни ($0 \leq y_i \leq 10^5$, $1 \leq i \leq N$).

Формат выходных данных

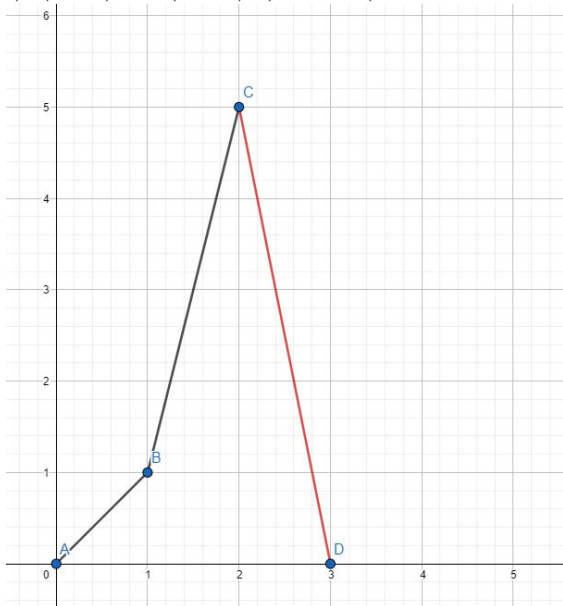
В выходной файл необходимо вывести одно число — наибольшую суммарную длину орошённых участков. Абсолютная или относительная погрешность должны быть не более 10^{-6} .

Пример

input.txt	output.txt
2 2 1 2 1 5	5.09901951359278449161

Замечание

Весь путь Иннокентия можно представить в виде последовательности $(0, 0), (1, y_1), \dots, (n, y_n), (n + 1, 0)$, где y — высоты над уровнем моря.



В первом примере наибольшую длину можно получить, применив заклинания в деревне 2 и в конце тракта.

Задача С. Поросячий бизнес

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Поросёнок Ниф-Ниф решил продолжить карьеру строителя домов. Он учел ошибки прошлого, поэтому теперь он будет строить дома из сена не себе, а другим зверям.

На строительство одного дома уходит W стогов сена. Так как сено на дороге не валяется, было решено закупать его у фермеров. i -й фермер продает неограниченное количество стогов за a_i монет каждый. Покупателей поросенок также нашел, j -й покупатель готов купить дом за b_j монет.

Каждый покупатель может купить только один дом. Помогите поросёнку максимизировать прибыль.

Формат входных данных

В первой строке входного файла записаны три числа: N , M и W — количества продавцов, количество покупателей и количество стогов, необходимых для постройки одного дома ($1 \leq N \leq 2 \cdot 10^5$, $1 \leq M \leq 2 \cdot 10^5$, $5 \leq W \leq 10^6$).

Во второй строке даны N чисел a_i — цены стогов сена у фермеров ($1 \leq a_i \leq 10^6$).

В третьей строке даны M чисел b_j — цены, за которые покупатели готовы купить дом ($1 \leq b_j \leq 10^6$).

Формат выходных данных

В выходной файл необходимо вывести одно число — максимально возможную прибыль Ниф-Нифа.

Пример

<code>input.txt</code>	<code>output.txt</code>
6 5 10 5 3 2 8 4 15 30 15 20 10 25	15

Задача D. Филворды

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Петя очень любит играть в филворды. Филворды — разновидность кроссвордов, где на поле $N \times M$ в каждую клетку вписаны буквы, и нужно соединять их линией, чтобы получать осмысленные слова. Но соединить можно только те буквы, что имеют общую грань, и никакую клетку нельзя включить более одного раза в одно и то же слово. Но любая клетка может входить в несколько слов одновременно.

Петя наткнулся на интересный филворд, решил его, но не уверен, что нашёл все слова. Он хочет проверить себя и зная, что Вы программист, просит написать ему программу, которая найдёт все возможные слова. При этом он просит искать только те слова, которые можно найти в словаре.

Формат входных данных

В первой строке входного файла содержатся два натуральных числа N и M — количество букв по вертикали и горизонтали в филворде соответственно ($1 \leq N, M \leq 500$).

В следующих N строках записано по M строчных латинских букв, задающих филворд.

Далее идет строка, в которой дано число D — количество слов в словаре ($1 \leq D \leq 10^4$).

В следующих D строках записаны слова, входящие в словарь. Каждое слово состоит из строчных латинских букв. Длина слова не превышает 10 символов.

Гарантируется, что все слова различны.

Формат выходных данных

В выходной файл требуется вывести целое число W — количество слов из словаря, которые можно найти в филворде.

В следующие W строк необходимо вывести найденные слова. Все выведенные слова должны быть различны. Порядок слов может быть любой.

Пример

<code>input.txt</code>	<code>output.txt</code>
5 5	8
fishy	dish
olioh	disk
rkdlw	fish
ksiet	folk
eeoto	fork
15	howl
fish	see
folk	seek
nose	
tomb	
howl	
dish	
fork	
see	
moto	
disk	
note	
host	
seek	
neck	
tie	

Задача Е. Про Дашу

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Даша играет в игру «Королевства и принцессы». У её персонажа есть по две характеристики в трех категориях: сила и защита; мудрость и интеллект; ловкость и выносливость. Во время приключений она столкнулась со злобным монстром, с которым ей предстоит сразиться.

Сражение происходит следующим образом: для каждой категории сравниваются характеристики и победителем объявляется тот, у кого либо обе характеристики больше, чем у оппонента, либо одна больше, а другая — равна. Выигрывает сражение тот, кто победил в строго большем количестве категорий.

Чтобы победить, Даша может добавить некоторое количество очков к характеристикам своего персонажа, но суммарно не более K . К сожалению, она не знает характеристик своего противника, но может провести несколько сражений, из которых ей достаточно выиграть хотя бы один раз.

Протокол взаимодействия

В первых шести строках заданы характеристики c_i Даши ($0 \leq c_i \leq 20$). В седьмой строке — количество очков K , которые она может распределить ($0 < K \leq 20$).

Характеристики монстра отличаются от характеристик Даши не более чем на 20 по модулю.

В данной задаче вам предстоит работать не с файловым вводом-выводом, а со специальной программой — интерактором. Взаимодействие с ней осуществляется через стандартные потоки ввода-вывода.

Каждый запрос вашей программы должен состоять из одной строки следующего вида: $! x_1 x_2 x_3 x_4 x_5 x_6$, где $x_1, x_2, x_3, x_4, x_5, x_6$ — целые числа, обозначающие характеристика персонажа Даши в очередном раунде. Все характеристики должны задаваться в порядке, перечисленном в условии. Ответом на запрос будет строка $c_1 c_2 c_3 c_4 c_5 c_6$ — результаты сравнений.

Результатом сравнения каждой характеристики является один из символов: $<$, если Даша проиграла по данной характеристике, $=$, если значение характеристики Даши совпало с значением характеристики монстра и $>$, если значение характеристики Даши больше значения характеристики монстра.

Если Даша выиграла раунд, ваше решение должно завершить работу, иначе после следующего запроса вы получите вердикт **Wrong Answer**. Гарантируется, что всегда можно победить монстра.

Если Даша проиграет 20 раз, то её персонаж погибнет, и ей придется создавать нового. А вы в этом случае получите вердикт **Wrong Answer**.

Убедитесь, что вы выводите символ перевода строки и очищаете буфер потока вывода (команда `flush` языка) после каждой выведенной команды. Иначе решение может получить вердикт **Timeout**.

Пример

стандартный ввод	стандартный вывод
2	
2	
2	
2	
2	
2	
15	
	! 17 2 2 2 2 2
> < < < < <	
	! 8 4 4 3 4 4
> = = < = =	
	! 7 4 4 4 4 4
> = = = = =	