

## Задача А. Найдите отличия

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

Даны два массива целых чисел  $a$  и  $b$  — оба длины  $n$ . Известно, что в массиве  $a$  выбрали число  $u$  и заменили его на число  $v \neq u$ , затем массив  $a$  перемешали, и так получился массив  $b$ .

По заданным массивам  $a$  и  $b$  определите и выведите два числа  $u$  и  $v$ . Обратите внимание, что нужно вывести значения чисел  $u$  и  $v$ , а не их индексы в массивах.

### Формат входных данных

В первой строке входных данных содержится целое число  $n$  ( $1 \leq n \leq 10^5$ ) — длина массивов  $a$  и  $b$ . Во второй строке содержится  $n$  целых чисел — массив  $a$  ( $-10^9 \leq a_i \leq 10^9$ ). В третьей строке содержится  $n$  целых чисел — массив  $b$  ( $-10^9 \leq b_i \leq 10^9$ ).

### Формат выходных данных

Выведите через пробел два целых числа — значения чисел  $u$  и  $v$ .

### Примеры

стандартный ввод	стандартный вывод
5 1 2 3 4 5 4 3 6 1 5	2 6
4 1 1 2 2 2 1 2 2	1 2
6 -5 2 10 11 12 20 20 -5 12 12 2 10	11 12

## Задача В. Мне повезет

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

Вы играете в игру на отчисление. Или в игру на «зачёт»: у вас имеется  $n$  баллов, для получения «зачёта» необходимо набрать  $m$  баллов.

Вы начинаете игру со ставки в 1 балл, затем с равной вероятностью  $1/2$  вы или теряете балл, или получаете, затем ставка увеличивается на 1 балл, и так далее до тех пор, пока игра не закончится (каждый раунд ставка повышается на 1 балл, при этом ставка может быть больше ваших баллов).

Игра заканчивается при двух условиях:

1. Если в какой-то момент вы набираете не менее  $m$  баллов — тогда вы получаете «зачёт»;
2. Если в какой-то момент у вас оказывается меньше нуля баллов — тогда вас отчисляют.

Определите вероятность того, что вы получите «зачёт», если сыграете в эту игру.

### Формат входных данных

В единственной строке входных данных содержатся два целых числа  $n, m$  ( $0 \leq n < m \leq 55$ ) — начальное количество баллов у вас и количество баллов, необходимое для «зачёта».

### Формат выходных данных

Выведите единственное число — вероятность того, что вы получите «зачёт» с точностью не менее  $10^{-9}$ .

### Примеры

стандартный ввод	стандартный вывод
1 3	0.500000000000000000
0 55	0.026584374625156282
9 10	0.893554687500000000

## Задача С. Выпуклость лепестка

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

Для визуализации некоторого множества значений параметров, которых можно привести к одной шкале, могут использоваться лепестковые диаграммы. В этой задаче количество параметров постоянно и равно 6, поэтому формально лепестковая диаграмма описывается так:

из некоторой точки  $D$  на двумерной плоскости исходят 6 лучей так, что угол между парой соседних при обходе в любом направлении равен  $60^\circ$ . На каждом из 6 лучей, начиная с некоторого и следуя далее по направлению часовой стрелки, отмечают точку: на  $i$ -ом луче точку отметили на расстоянии  $a_i$  от точки  $D$ . Далее, так же следуя по направлению часовой стрелки, соединили все точки на соседних лучах и получили ограниченную отрезками область — лепестковую диаграмму.

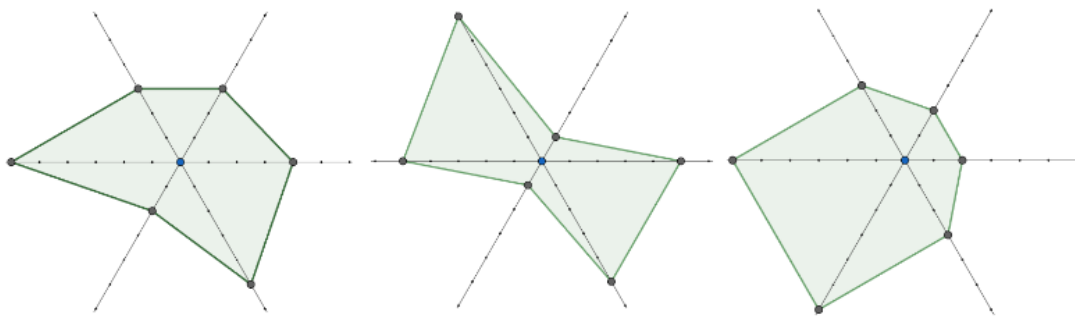


Иллюстрация к трем тестовым примерам

Определите по заданным числам  $a_i$ , отмеченным на лучах, является ли построенный многоугольник лепестковой диаграммы выпуклым.

### Формат входных данных

В единственной строке содержатся 6 целых чисел  $a$  ( $1 \leq a_i \leq 10^9$ ) — расстояния от точки  $D$ , на котором отметили точки, начиная с некоторого луча и следуя далее по направлению часовой стрелки.

### Формат выходных данных

Если построенный многоугольник является выпуклым, выведите «Convex»; иначе выведите «Non-convex».

### Примеры

стандартный ввод	стандартный вывод
3 3 4 5 2 6	Non-convex
6 1 5 5 1 5	Non-convex
3 2 2 3 6 6	Convex

## Задача D. Три фитиля

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

У вас есть 3 фитиля с целочисленными длинами  $a$ ,  $b$ ,  $c$ . Фитили можно поджигать и тушить. Каждый фитиль горит равномерно, причем фитиль длины  $x$  при поджигании его с одного конца сгорает полностью за  $x$  секунд, а при поджигании с двух концов — за  $x/2$  секунд.

Вы хотите с помощью зажигалки и имеющихся фитилей отмерить ровно  $t$  секунд. Определите, возможно ли это.

Процесс замера  $t$  секунд формально описывается следующими условиями:

1. Процессы поджигания фитилей зажигалкой и их тушения происходят моментально;
2. Отсчет времени начинается с момента первого поджигания одного из фитилей (вы можете поджечь сразу несколько) — после первого поджигания фитиля вы должны через ровно  $t$  секунд определить, что прошло  $t$  секунд;
3. После начала отсчета времени вы не можете поджигать или тушить фитили в произвольный момент времени: отныне поджигать и тушить фитили можно только в моменты времени когда какой-либо фитиль сгорел полностью;
4. Любой фитиль можно поджигать как с одной стороны, так и с двух. То же самое касается процесса тушения.

### Формат входных данных

В первой строке входных данных содержатся три целых числа  $a$ ,  $b$ ,  $c$  ( $1 \leq a, b, c \leq 1000$ ) — длины фитилей. Во второй строке входных данных содержится целое число  $t$  ( $1 \leq t \leq 3000$ ) — время, которое вам нужно отмерить.

### Формат выходных данных

Выведите «Yes», если с помощью фитилей можно отмерить  $t$  секунд, иначе выведите «No».

### Примеры

стандартный ввод	стандартный вывод
1 3 5 2	Yes
8 3 15 6	No
6 4 12 13	Yes

### Замечание

В первом примере одновременно поджигаем фитиль длины 1 и длины 3, через 1 секунду фитиль с длиной 1 сгорает, а от фитиля первоначальной длины 3 остается фитиль длины 2, далее поджигаем фитиль длины 2 с двух сторон и он сгорает через еще одну секунду. Итого мы отмерили 2 секунды. Фитиль длины 5 остался нетронутым.

Во втором примере нельзя отмерить 6 секунд.

В третьем примере одновременно поджигаем все три фитиля с одного конца, через 4 секунды фитиль с длиной 4 сгорает — в этот момент мы тушим фитиль с первоначальной длиной 12 (от него осталась длина 8) и поджигаем с двух сторон фитиль с первоначальной длиной 6 (в этот момент его длина равна 2), через 1 секунду (и 5 секунд с отсчета времени) фитиль с длиной 2 сгорает, в этот момент поджигаем фитиль с длиной 8 с одного конца, через 8 секунд он сгорает и мы отсчитали ровно 13 секунд.

## Задача Е. Операции с массивом

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

Задан массив целых положительных чисел  $a$ , состоящий из  $n$  элементов. С этим массивом можно сколько угодно раз выполнять операцию  $V$ , которая описывается следующим образом:

$$V(a) : a_i = a_i - 1 \text{ for } i = \min(j : a_j = \max(a))$$

По простому: выбирается такой минимальный индекс массива  $i$ , что элемент  $a_i$  равен максимуму в массиве  $a$ , затем элемент  $a_i$  уменьшается на единицу. Определите, какое минимальное количество раз нужно применить операцию  $V$ , чтобы в массиве  $a$  стало ровно  $k$  нулей.

### Формат входных данных

В первой строке входных данных содержатся два целых числа  $n, k$  ( $1 \leq k \leq n \leq 2 \times 10^5$ ) — количество элементов в массиве  $a$  и количество нулей, которое нужно получить. Во второй строке содержатся  $n$  целых положительных чисел  $a$  ( $1 \leq a_i \leq 10^9$ ).

### Формат выходных данных

Выведите единственное целое число — минимальное количество применений операции  $V$ , чтобы в массиве  $a$  стало ровно  $k$  нулей. Гарантируется, что ответ всегда существует.

### Примеры

стандартный ввод	стандартный вывод
3 1 1 1 1	1
4 2 4 1 5 2	10

## Задача F. Покрытие вершин дерева

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Дано корневое дерево на  $n$  вершинах с корнем в вершине 1. У каждой вершины дерева есть вес — вес  $i$ -ой вершины равен  $w_i$ . Обратите внимание, что вес может быть неположительным.

Заданы  $m$  особых различных вершин в дереве, которые требуется *покрыть*. Вы можете любое число раз выполнять следующую операцию:

- Выбрать некоторую вершину дерева  $u$  и *покрыть* все вершины в поддереве<sup>†</sup> вершины  $u$ . Каждая вершина может быть покрыта лишь единожды.

Определите, какую минимальную сумму весов всех покрытых вершин можно получить, используя описанную операцию любое число раз, чтобы все  $m$  особых вершин были покрыты.

### Формат входных данных

В первой строке входных данных содержится целое число  $n$  ( $1 \leq n \leq 10^5$ ) — количество вершин в дереве. Во второй строке содержатся  $n$  целых чисел  $p$ , где  $p_1 = 1$  и  $p_i$  — родительская вершина для вершины  $i$  при  $i > 1$ . В третьей строке содержатся  $n$  целых чисел  $w$  ( $-10^9 \leq w_i \leq 10^9$ ) — веса для каждой вершины дерева.

В четвертой строке содержится целое число  $m$  ( $1 \leq m \leq n$ ) — количество особых вершин в дереве, которые необходимо покрыть. В пятой строке содержатся  $m$  различных целых чисел  $d$  ( $1 \leq d_i \leq n$ ) — особые вершины дерева.

### Формат выходных данных

Выведите единственное целое число — минимальную возможную сумму весов всех покрытых вершин дерева, при условии, что все  $m$  особых вершин так же покрыты.

### Примеры

стандартный ввод	стандартный вывод
8 1 1 1 1 2 3 3 4 1 -2 1 5 4 0 2 -3 2 3 5	2
8 1 1 1 1 2 3 3 4 1 -2 1 5 4 0 2 -3 4 2 6 7 4	6
3 1 1 1 -2 4 4 2 2 3	6

### Замечание

<sup>†</sup> Вершинами в поддереве некоторой вершины  $u$  является сама вершина  $u$ , а также все вершины, достижимые из  $u$  при переходе по ребрам вниз по дереву (в направлении увеличения расстояния до корневой вершины).

## Задача G. КНБ

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

В игре «камень-ножницы-бумага» камень побеждает ножницы, ножницы побеждают бумагу, а бумага побеждает камень. У вас есть список из  $n$  элементов, каждый из которых или камень, или ножницы, или бумага.

За один ход вы можете выбрать два **различных по типу** элемента и провести между ними игру, по итогу игры проигравший элемент заменяется на победивший. Определите минимальное число ходов, чтобы получить список из  $n$  одинаковых элементов.

### Формат входных данных

В первой строке входных данных содержится целое число  $n$  ( $1 \leq n \leq 10^5$ ). Во второй строке через пробел содержится список  $s$  из  $n$  элементов ( $s_i \in R, S, P$ ), где «R» — камень, «S» — ножницы, «P» — бумага.

### Формат выходных данных

Выведите единственное целое число — минимальное число ходов, чтобы в списке  $s$  все элементы стали одинаковыми.

### Примеры

стандартный ввод	стандартный вывод
5 R P S S P	4
7 P S R R P R R	5
3 S S S	0

## Задача Н. Таблица умножения

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

Представьте себе бесконечную по размеру таблицу умножения: на пересечении  $i$ -го столбца и  $j$ -й строки в ней содержится число  $i \cdot j$ . Определите, сколько чисел в этой таблице строго меньше заданного числа  $n$ .

### Формат входных данных

В единственной строке входных данных содержится целое число  $n$  ( $1 \leq n \leq 10^{12}$ ).

### Формат выходных данных

Выведите единственное целое число — количество чисел в бесконечной таблице умножения строго меньших  $n$ .

### Примеры

стандартный ввод	стандартный вывод
1	0
5	8
10	23



## Задача I. Вставки в очереди

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

В очереди за мороженым стоит один человек — Aaron, познакомьтесь! Затем поочередно по одному приходят новые люди и встают в очередь перед кем-то или за кем-то, кто уже стоит в очереди.

Вам нужно обработать  $q$  запросов, запросы бывают 3 типов:

1.  $\langle \text{name1} \rangle$  *before*  $\langle \text{name2} \rangle$  — приходит человек с именем  $\langle \text{name1} \rangle$  и встает в очереди **перед** человеком с именем  $\langle \text{name2} \rangle$ ;
2.  $\langle \text{name1} \rangle$  *after*  $\langle \text{name2} \rangle$  — приходит человек с именем  $\langle \text{name1} \rangle$  и встает в очереди **после** человека с именем  $\langle \text{name2} \rangle$ ;
3. *leave*  $\langle \text{name} \rangle$  — человек с именем  $\langle \text{name1} \rangle$  покидает очередь, а все люди после него сдвигаются на одну позицию ближе к началу очереди.

Вам нужно обработать все  $q$  запросов и определить, как будет выглядеть очередь после выполнения всех запросов.

Гарантируется, что:

1. Для запросов первых двух типов человек с соответствующим именем  $\langle \text{name2} \rangle$  есть в очереди, а с именем  $\langle \text{name1} \rangle$  нет в очереди;
2. Для запроса третьего типа человек с соответствующим именем  $\langle \text{name} \rangle$  есть в очереди;
3. Не бывает двух различных людей с одним именем, но человек с одним именем может уходить из очереди, а затем вставать в нее, и делать так произвольное число раз.

### Формат входных данных

В первой строке входных данных содержится целое число  $q$  ( $1 \leq q \leq 2 \times 10^5$ ) — количество запросов. Далее следуют  $q$  строк, каждая строка является командой одного из трех типов. Все используемые в задаче имена состоят только из букв латинского алфавита и имеют длину от 1 до 10 символов.

### Формат выходных данных

Выведите имена всех людей в очереди после выполнения всех запросов в том порядке, в котором они находятся в очереди (первое имя — имя первого человека в очереди, второе — второго в очереди и т.д.). Выводите каждое имя в отдельной строке. Если очередь пустая, ничего не выводите.

## Примеры

стандартный ввод	стандартный вывод
4 Din before Aaron Jacob after Aaron Boris after Din leave Din	Boris Aaron Jacob
6 Vova after Aaron DJgold before Vova leave Aaron Fedor after DJgold Rick before Vova Aaron after Rick	DJgold Fedor Rick Aaron Vova
9 Roma after Aaron Dima after Roma Masha before Aaron Anton after Roma voenkomat after Anton leave Anton leave Aaron leave Dima leave Roma	Masha voenkomat

## Задача J. Запросы на кувшинах

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Есть  $n$  кувшинов, вместимость  $i$ -го кувшина равна  $v_i$ .

Поступает  $q$  запросов, каждый  $i$ -ый запрос описывается единственным числом  $d_i$ . Требуется для каждого  $i$ -го запроса сказать, какое минимальное количество кувшинов нужно взять, чтобы их суммарная вместимость была не менее  $d_i$ .

### Формат входных данных

В первой строке входных данных содержится целое число  $n$  ( $1 \leq n \leq 2 \times 10^5$ ). Во второй строке содержится  $n$  целых чисел  $v$  ( $1 \leq v_i \leq 10^9$ ) — вместимости кувшинов.

В третьей строке содержится целое число  $q$  ( $1 \leq q \leq 2 \times 10^5$ ) — количество запросов. Затем следуют  $q$  строк, в  $i$ -й содержится целое число  $d_i$  ( $1 \leq d_i \leq \text{sum}(v)$ ) — необходимая суммарная вместимость кувшинов в  $i$ -ом запросе.

### Формат выходных данных

Для каждого  $i$ -го запроса в отдельной строке в том же порядке выведите одно целое число — какое минимальное количество кувшинов нужно взять, чтобы их суммарная вместимость была не менее  $d_i$ .

### Примеры

стандартный ввод	стандартный вывод
5 1 1 3 2 4 3 7 4 10	2 1 4
8 10 2 8 9 14 12 9 5 4 50 18 31 40	5 2 3 4

## Задача К. Они чередуются!

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

Вам дана бинарная строка  $s$  длины  $n$ . Требуется определить длину ее самой большой чередующейся<sup>†</sup> подстроки.

<sup>†</sup> Чередующейся подстрокой в бинарной строке  $s$  назовем любую последовательность подряд идущих символов с  $l$  по  $r$ , что  $s_l \neq s_{l+1} \neq s_{l+2} \neq \dots \neq s_r$ .

### Формат входных данных

В первой строке входных данных содержится целое число  $n$  ( $1 \leq n \leq 5 \times 10^5$ ) — длина строки  $s$ . Во второй строке входных данных содержится сама строка  $s$  длины  $n$ . Каждый символ строки или «0», или «1».

### Формат выходных данных

Выведите единственное целое число — максимальную длину чередующейся подстроки в строке  $s$ .

### Примеры

стандартный ввод	стандартный вывод
5 10010	3
4 0000	1
9 011010100	6

## Задача L. Заготовка камня

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

Фараон приказал вам заготовить  $n$  блоков камня для последующей постройки пирамиды. На задачу у вас есть  $d$  дней, если вы не успеваете за  $d$  дней заготовить  $n$  блоков камня, то вас отправят в ссылку в Сибирь.

Сами вы не заготавливаете камень, а лишь управляете процессом, в частности: нанимаете рабочих. На местной бирже труда есть  $m$  рабочих, каждый  $i$ -ый рабочий описывается двумя числами  $a_i$  и  $b_i$ , означающими, что при наеме рабочего  $i$  вы отдадите  $a_i$  золотых рублей, затем через  $b_i$  дней рабочий заготовит один блок камня и покинет место работы. Вы можете нанимать одного рабочего повторно сколько угодно раз, каждый раз отдавая соответствующую плату.

Вам нужно определить такое **минимальное** число золотых рублей, при котором вы можете нанять рабочих для изготовления  $n$  блоков камня не более чем за  $d$  дней.

### Формат входных данных

В первой строке входных данных содержатся два целых числа  $n, d$  ( $1 \leq n, d \leq 10^9$ ) — количество блоков камня для заготовки и количество дней, за которое нужно успеть заготовить блоки. Во второй строке содержится целое число  $m$  ( $1 \leq m \leq 10^5$ ) — количество рабочих на бирже. Затем следуют  $m$  строк, каждая  $i$ -ая строка содержит два целых числа  $a_i, b_i$  ( $1 \leq a_i, b_i \leq 10^9$ ) — стоимость в золотых рублях наема рабочего  $i$  и срок заготовки одного блока камня рабочим  $i$  соответственно.

### Формат выходных данных

Выведите единственное целое число — **минимальное** число золотых рублей, при котором вы можете нанять рабочих для изготовления  $n$  блоков камня не более чем за  $d$  дней. Если же ни при каких условиях невозможно за  $d$  дней заготовить  $n$  блоков камня — выведите «-1» и собирайте чемоданы.

### Примеры

стандартный ввод	стандартный вывод
4 10 3 4 5 2 8 1 7	11
10 5 4 1 2 4 1 2 2 8 4	34
12 4 2 4 1 2 2	-1

### Замечание

Обратите внимание, что «готовый» блок камня  $i$ -й рабочий получает через  $b_i$  дней, недоделанные блоки не считаются, т.е. вы должны сделать  $n$  «готовых» блоков не более чем за  $d$  дней. Так же над одним блоком камня не могут работать несколько разных рабочих.

## Задача М. Чипи-чипи Чапа-чапа

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

Два друга улитки Чипи-чипи и Чапа-чапа договорились встретиться в баре у Старого Пня. Они договорились встретиться ровно в полдень, но часы каждой улитки могут ошибаться, из-за чего каждая улитка может как приползти раньше, так и позже (или же ровно в полдень, если часы не ошибаются).

Каждая улитка не знает, что ее часы могут ошибаться, но ориентируется на них и приползает в бар ровно в 12:00 по своим часам. Вам известно спешат или опаздывают часы каждой улитки и на сколько минут. Определите, какая улитка приползет к бару раньше и сколько минут она будет ждать своего друга.

### Формат входных данных

В первой строке входных данных содержится описание часов Чипи-чипа в таком формате:

- Если часы показывают точное время, в строке содержится единственное слово «accurately»;
- Если часы спешат на  $n > 0$  минут, то в строке содержится слово «hurry», а затем целое число  $n$ ;
- Если часы опаздывают на  $n > 0$  минут, то в строке содержится слово «late», а затем целое число  $n$ .

Во второй строке по таким же правилам задано описание часов Чапа-чапы. Гарантируется, что часы каждой улитки ошибаются не более чем на 10 минут.

### Формат выходных данных

Если улитки приползут в бар одновременно, выведите «together». Иначе выведите имя улитки, которая приползет в бар раньше: если это будет Чипи-чипи — выведите «Chipy-chipy», если же Чапа-чапа — выведите «Chapa-chapa»; после имени улитки через пробел нужно вывести одно целое положительное число — сколько минут улитка будет ждать прибытие своего друга.

### Примеры

стандартный ввод	стандартный вывод
accurately hurry 4	Chapa-chapa 4
late 10 late 10	together
hurry 8 late 5	Chipy-chipy 13