Задача A. Архимед

Имя входного файла:	*стандартный ввод* или input.txt
Имя выходного файла:	*стандартный вывод* или output.txt
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Недавно Виталька прошёл в школе закон Архимеда, в котором говорится, что на тело, погружённое в жидкость, действует выталкивающая или подъёмная сила, равная весу объёма жидкости, вытесненного частью тела, погружённой в жидкость. Так как Виталька любит придумывать различные задачки, то он сразу составил задачу (интересную на его взгляд), в которой по заданным размерам бассейна и начальному объему воды V в нём, нужно определить какой объем воды выльется через края, если в него поместить N предметов, каждый из которых имеет объём V_i.

Можно считать, что предметы укладываются максимально плотно и никакой предмет не выходит за края бассейна, если в бассейне еще есть свободное место (место, занятое либо водой, либо воздухом). Плотность предметов выше плотности воды (предметы не всплывают). Никакие предметы не погружаются в воду вместе с воздухом, то есть нет полых предметов (например, таких как кружка).

Формат входных данных

В первой строке заданы целые числа W, H, L, V и N, где W, H, L — ширина, глубина и длина бассейна, V — изначальный объём воды в бассейне и N — количество помещаемых предметов ($1 \leqslant W, H, L \leqslant 1000$; $0 \leqslant V \leqslant W \times H \times L$; $0 \leqslant N \leqslant 100$).

Во второй строке содержатся N целых чисел V_i ($1 \leqslant V_i \leqslant 1000$).

Формат выходных данных

В единственной строке требуется вывести ответ на придуманную Виталькой задачу .

Пример

стандартный ввод или input.txt	стандартный вывод или output.txt
10 10 10 900 5 100 10 100 100 250	460

Задача B. Верхняя граница

Имя входного файла:	*стандартный ввод* или input.txt
Имя выходного файла:	*стандартный вывод* или output.txt
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

У Ромы и Саши есть набор карточек с цифрами: для каждой цифры k от 0 до 9 имеется c_k карточек с цифрой k.

Саша хочет составить целое число, которое строго больше X. Рома дополнительно хочет, чтобы это число было как можно меньше.

Число составляется из карточек по его десятичной записи.

Определите, какое число им нужно составить или сообщите, что это невозможно.

Формат входных данных

В первой строке содержится целое неотрицательное число X, записанное в десятичной системе счисления без ведущих нулей. Длина числа не превосходит 10^5.

Во второй строке содержатся числа c_0, c_1, ..., c_9 ($0 \leqslant c_i \leqslant 10^6$).

Формат выходных данных

Если невозможно составить требуемое число, выведите «-1».

Иначе, выведите число без ведущих нулей.

Примеры

стандартный ввод или input.txt	стандартный вывод или output.txt
2019 1 2 0 3 0 0 0 0 0 0	3011
165 1 1 1 1 1 1 1 1 1 1	167
165 0 0 0 0 0 0 0 0 0 0	-1

Задача C. Волшебные цветы

Имя входного файла:	*стандартный ввод* или input.txt
Имя выходного файла:	*стандартный вывод* или output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Цветочник Стёпик посадил n волшебных цветков в своём саду. Высадил он их по кругу, и пронумеровал от 1 до n так, что первый и n-й соседи, а для $2 \leqslant i \leqslant n$ соседями являются цветы с номерами i и $i-1$. Каждый i-й цветок имеет свою высоту h_i.

Почему цветы Стёпика волшебные? Потому что к концу дня они меняют свою высоту, а именно высота i-го цветка становится равна наименьшей из высот его соседей. Изменение высот происходит для всех цветков одновременно.

Стёпик собирается уехать на k дней и ему интересно, какими будут высоты цветков к моменту его приезда?

Формат входных данных

Первая строка содержит два целых числа n и k — количество цветков и количество дней соответственно ($3 \leqslant n \leqslant 5 \times 10^5$; $0 \leqslant k \leqslant 10^9$).

Во второй строке содержатся n целых чисел h_i ($1 \leqslant h_i \leqslant 10^9$).

Формат выходных данных

Выведите в одной строке n целых чисел — высоты цветков по истечении k дней.

Примеры

стандартный ввод или input.txt	стандартный вывод или output.txt
5 1 3 5 4 1 2	2 3 1 2 1
10 2 9 2 6 7 1 4 5 8 3 10	3 2 1 2 1 4 1 4 3 2
3 3 3 3 3	3 3 3

Задача D. Двоичное упражнение

Имя входного файла:	*стандартный ввод* или input.txt
Имя выходного файла:	*стандартный вывод* или output.txt
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Маленький Бикарп очень любит степени двойки. Недавно он узнал о двоичной системе счисления и решил поупражняться на задачках по этой теме.

Ему попалась задача о сложении двух чисел, где требовалось определить количество единиц в двоичной записи этой суммы. В этой задаче требовалось сложить числа $2^x - 1$ и $2^y - 1$.

Пока Бикарп решает задачу на бумаге, напишите программу, которая быстро проверит его ответ.

Формат входных данных

Первая строка содержит число x, вторая — y.

Оба числа целые. $0 \leqslant x, y \leqslant 2^{30}$.

Формат выходных данных

Выведите количество единиц в двоичной записи требуемой суммы.

Пример

стандартный ввод или input.txt	стандартный вывод или output.txt
2 2	2

Задача E. Stack Unwinding

Имя входного файла:	*стандартный ввод* или input.txt
Имя выходного файла:	*стандартный вывод* или output.txt
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Давид увлекается изучением своего любимого языка программирования. Для этого он иногда просматривает видео с различных курсов. На одном из них он познакомился с исключениями и теперь везде их ~~пихает~~использует. К сожалению, он не совсем понимает в каком порядке удаляются объекты при срабатывании исключения, поэтому просит вас по написанному им коду, вывести в каком порядке будут создаваться и удаляться объекты в программе. Напомним, что в языке Давида, при завершении функции объекты удаляются в обратном порядке, а при срабатывании исключения, программа завершается (т.к. Давид забыл написать соответствующий обработчик), предварительно освобождая созданные ей объекты. Выполнение программы начинается с функции *main*.

Формат входных данных

В нескольких строках содержится псевдокод программы. Он представляет собой описание нескольких функций (не больше 10), каждая из которых задается следующим образом. Вначале записывается сигнатура функции. Для упрощения во всех программах сигнатура функции будет выглядеть как «`void funcName() {`» (без кавычек), где *funcName* — имя объявляемой функции. Затем до «`}`» описывается тело функции, в котором могут быть только следующие команды:

- «`Type name`» — создать объект *name* типа *Type*.
- «`throw`» — место, где сработает исключение. Гарантируется, что исключение встречается ровно в одном месте.
- «`funcName()`» — вызов функции. Здесь *funcName* - имя функции, которая описана в программе.

Каждая команда описана в отдельной строке. Все имена состоят только из латинских букв и цифр. Код программы не превышает 10 000 символов. Гарантируется, что определение функции *main* является последним, а так же то, что программа совершает конечное число шагов.

Формат выходных данных

Требуется вывести в каком порядке будут создаваться и удаляться объекты.

При создании объекта типа *Type* нужно вывести «`Type()`», при удалении — «`~Type()`», возникновение исключения — «`throw`».

Гарантируется, что потребуется вывести не более 20 000 строк.

Пример

стандартный ввод или input.txt	стандартный вывод или output.txt

void foo() {
B b
C c
bar()
throw
bar()
}
void bar() {
T t
}
void main() {
A a
foo()
}
	A()
	B()
	C()
	T()
	~T()
	throw
	~C()
	~B()
	~A()

Задача F. Пробежка

Имя входного файла:	*стандартный ввод* или input.txt
Имя выходного файла:	*стандартный вывод* или output.txt
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Хомячок Хома наконец-то получил в подарок долгожданный тренажёр — колесо для бега. Колесо представляет из себя два круга, соединённых n спицами. Спицы пронумерованы числами от 1 до n по часовой стрелке. Соседние спицы расположены на равном расстоянии друг от друга.

Хома начинает пробежку так, что его передние лапки стоят на первой спице. Далее он делает k шагов таким образом, что на переход к следующей спице он тратит одну секунду. Сделав k шагов, он смотрит на какой спице он остановился: если это первая спица, то он завершает пробежку и слазит с колеса, а иначе отдыхает r секунд и повторяет процедуру снова.

Хозяйка Хомы интересуется, сколько времени будет занимать одна такая пробежка?

Формат входных данных

Ввод содержит три целых числа n, k и r по одному в отдельной строке.
$1 \leqslant n, k, r \leqslant 2 \cdot 10^9$.

Формат выходных данных

Выведите единственное число — сколько секунд потратит Хома на пробежку.

Пример

стандартный ввод или input.txt	стандартный вывод или output.txt
6 4 1	14

Задача G. Три монеты

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

Это интерактивная задача.

Перед вами стопка из n монет. В стопке есть все монеты номиналов от 1 до n, причём они упорядочены по возрастанию сверху-вниз. То есть, верхняя монета номиналом 1, а нижняя — n.

Вес каждой монеты равен её номиналу.

Некто заменил ровно три монеты в стопке на фальшивые — вес таких монет равен нулю.

Вы можете спрашивать суммарный вес первых k монет сверху для любого $0 \leqslant k \leqslant n$.

Определите номиналы фальшивых монет **не более чем за 32 вопроса**, и тогда Некто подарит вам всю стопку монет.

Протокол взаимодействия

Вначале на ввод подаётся число n — количество монет ($3 \leqslant n \leqslant 10^9$).

Далее ваша программа должна выводить запросы вида «`? k`», где $0 \leqslant k \leqslant n$.

В ответ вы получите сумму весов первых k монет вверху стопки.

Как только ваша программа будет готова сообщить ответ, выведите «`! a b c`», где $1 \leqslant a, b, c \leqslant n$ — различные номиналы фальшивых монет.

После этого программа должна завершить работу.

Пример

стандартный ввод	стандартный вывод
6 12 6 6 2 2 0 0	? 6 ? 5 ? 4 ? 3 ? 2 ? 1 ? 0 ! 1 5 3

Замечание

Для корректной работы программы после каждой операции вывода данных вам необходимо выводить **перенос строки**, а также очищать буфер вывода, то есть делать следующие операции:

- В языке Pascal: **flush(output)**;
- В C/C++: **fflush(stdout)** или **cout.flush()**;
- В Java: **System.out.flush()**;
- В Python: **sys.stdout.flush()** из библиотеки **sys**;
- В C#: **Console.Out.Flush()**;

Задача H. Кинозал

Имя входного файла:	*стандартный ввод* или `input.txt`
Имя выходного файла:	*стандартный вывод* или `output.txt`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Однажды Вася пришёл в известный кинотеатр «Прямая», чтобы посмотреть новый боевик. Когда он проходил к своему месту в зале, то ему пришлось потревожить некоторых людей, которые заняли свои места раньше него. Он был этим так расстроен, что решил в следующий раз заранее определить минимальное количество людей, которым он должен помешать.

Известно, что кинозал имеет форму прямоугольника шириной W (количество мест в ряду) и высотой H (количество рядов). Каждая клетка внутри прямоугольника задаётся координатами (a, b) так, что $1 \leqslant a \leqslant H$ и $1 \leqslant b \leqslant W$. Клетка может быть либо местом, на котором можно сидеть зрителю, либо быть часть прохода. Всего существует два типа проходов: вертикальный и горизонтальный, оба проходят от края до края. По проходам можно перемещаться свободно, не мешая зрителям. Дополнительно, по периметру зала тоже можно перемещаться свободно.

Вам дано описание зала с указанием проходов и уже занятых мест. Определите минимальное количество человек, которое необходимо потревожить Васе, чтобы добраться до своего места.

Помимо прохождения по проходам, Вася может двигаться вдоль любого ряда. Считается, что Вася тревожит зрителя, если он проходит через его место на пути к своему. Аналогичное движение в направлении, перпендикулярном рядам, невозможно. Однако, если перед местом Васи располагается горизонтальный проход, то Вася его может занять, не потревожив других зрителей. Также, Вася довольно культурный и не перепрыгивает через спинки кресел.

Формат входных данных

В первой строке заданы четыре целых числа W, H, C и P ($1 \leqslant W, H \leqslant 10^9$, $0 \leqslant C, P \leqslant 10^5$), где W и H — размеры кинозала, C — количество уже занятых мест, P — количество проходов.

В следующих P строках описаны проходы. Горизонтальный проход описывается как «`hor` h_i» ($1 \leqslant h_i \leqslant H$), а вертикальный — «`vert` v_i» ($1 \leqslant v_i \leqslant W$).

Затем в C строках записаны координаты (a_i, b_i) занятых мест ($1 \leqslant a_i \leqslant H$, $1 \leqslant b_i \leqslant W$).

Последняя строка содержит координаты места Васи в аналогичном формате. Гарантируется, что ни одно место не указано дважды, а так же то, что места не находятся в проходах.

Формат выходных данных

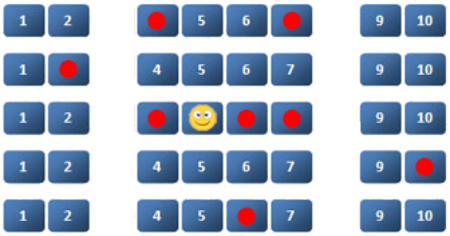
Выведите целое число — минимальное количество людей, которое нужно потревожить Васе.

Примеры

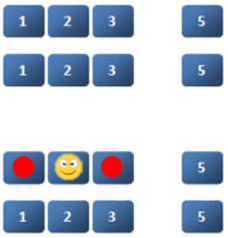
стандартный ввод или input.txt	стандартный вывод или output.txt
10 5 8 2 vert 3 vert 8 2 2 5 6 1 7 3 4 3 6 3 7 4 10 1 4 3 5	1
5 5 2 2 hor 3 vert 4 4 1 4 3 4 2	0

Замечание

Иллюстрации примеров. Считается, что экран кинотеатра находится сверху.



Пример 1.



Пример 2.

Задача I. Индикатор загрузки

Имя входного файла:	*стандартный ввод* или input.txt
Имя выходного файла:	*стандартный вывод* или output.txt
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Для отображения процесса загрузки файла часто используют индикатор в виде полоски, заполняющейся слева направо.

Мы имеем дело с индикатором длиной в n пикселей. Если индикатор отображает k пикселей, то это значит, что хотя бы $\frac{k}{n}$-я часть файла загружена.

При этом для уменьшения неопределённости заполняется максимальное возможное количество пикселей.

Вам нужно вывести индикатор длиной в n пикселей в момент, когда загружено ровно $p\%$ файла.

Формат входных данных

Единственная строка содержит два целых числа n и p ($1 \leqslant n \leqslant 1\,000$; $0 \leqslant p \leqslant 100$).

Формат выходных данных

Выведите индикатор загрузки, окружив его квадратными скобками.

Заполненную часть индикатора следует обозначать символом «`*`» («звёздочка»), а пустую — «`.`» (точка).

Примеры

стандартный ввод или input.txt	стандартный вывод или output.txt
10 40	[****......]
10 99	[*********.]

Задача J. Заверните две!

Имя входного файла:	*стандартный ввод* или input.txt
Имя выходного файла:	*стандартный вывод* или output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Петя как обычно шёл утром на работу по обычной дороге, как вдруг ему на глаза попалась новая шаурмичная. Он решил зайти в неё и запастись пищей на весь следующий день. Петя хочет купить ровно две шаурмы — на обед и ужин (не беспокойтесь, микроволновка на работе у Пети есть).

Всего в шаурмичной существует n ингредиентов, пронумерованных от 1 до n, и m рецептов шаурмы. Каждый рецепт для удобства представляет из себя отрезок $[l_i, r_i]$, и означает, что в i-й рецепт входят ингредиенты с номерами от l_i до r_i включительно.

Петя быстрым взглядом ознакомился со всеми ингредиентами и оценил каждый i-й ингредиент целым числом a_i — насколько ему нравится этот продукт. Естественно, Петя хочет получить максимально возможное суммарное удовольствие от приёма пищи. Удовольствие от одной шаурмы равно сумме всех a_i входящих в неё ингредиентов.

Так же, Петя хочет разнообразия — выбранные рецепты не должны иметь общего ингредиента.

Помогите Пете с нелёгким выбором, определите, какие две шаурмы нужно заказать, чтобы условия были выполнены, а он в качестве благодарности угостит вас как-нибудь шаурмой.

Формат входных данных

Первая строка содержит целое число n — количество ингредиентов ($1 \leqslant n \leqslant 2 \times 10^5$).

Во второй строке содержатся n целых чисел a_i ($|a_i| \leqslant 10^9$).

Третья строка содержит целое число m — количество рецептов ($2 \leqslant m \leqslant 2 \times 10^5$).

Следующие m строк содержат пары чисел l_i и r_i — границы i-го отрезка ($1 \leqslant l_i \leqslant r_i \leqslant n$).

Формат выходных данных

Выведите номера рецептов, которые стоит выбрать Пете.

Если же выбрать невозможно, выведите «-1 -1».

Примеры

стандартный ввод или input.txt	стандартный вывод или output.txt
5 5 1 2 -3 1 3 1 4 2 3 4 5	2 3
3 1 2 3 2 1 2 2 3	-1 -1

Задача K. Сладкая вата

Имя входного файла:	*стандартный ввод* или input.txt
Имя выходного файла:	*стандартный вывод* или output.txt
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Как все вы знаете, решение олимпиадных задач на алгоритмы и структуры данных — дело весьма энергозатратное. Именно поэтому все участники Открытого личного первенства ИКИТ СФУ по программированию по окончании соревнований получают сладкую вату калорийностью C калорий. На олимпиаде участникам предлагается решить N задач, для решения каждой требуется потратить V калорий.

Программист Федя будет участвовать в олимпиаде и собирается решить все задачи! Он хочет посчитать, хватит ли калорий, полученных из сладкой ваты, чтобы компенсировать калории, потраченные на решение всех задач. Помогите ему!

Формат входных данных

В одной строке три целых числа C, N, V ($1 \leqslant C, V \leqslant 10^4$, $1 \leqslant N \leqslant 15$).

Формат выходных данных

Одно число — разница между калорийностью ваты и калориями, необходимыми для решения всех задач олимпиады.

Примеры

стандартный ввод или input.txt	стандартный вывод или output.txt
100 10 5	50
15 3 6	-3