

## Задача А. ИИ

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

Работа в Яндексe шла своим чередом. Закончив работу над проектом, Стёпа и Катя решили продолжить создание нового искусственного интеллекта. Спустя множество проб и ошибок, они были близки к завершению. Всё, что осталось — калибровка. Процесс делится на три шага:

1. ИИ выдает два числа  $A$  и  $B$ .
2. На основе полученных чисел начинаются расчеты.
3. Проверяется контрольное значение.

Ребята решили не терять времени даром и запустили процесс. Но вот незадача, во время второго шага настало время обеда. Прием пищи — важная составляющая, а значит, его пропускать нельзя. Тогда Стёпа решил обратиться к Вам за помощью. Всё, что требуется — посчитать контрольное значение.

По словам Кати, алгоритм получения значения полностью зависит от числа  $B$ .

- В случае, если  $B$  неотрицательно, то необходимо к нулю  $B$  раз прибавить  $A$ ;
- В противном случае, необходимо из нуля  $|B|$  раз вычесть  $A$ .

### Формат входных данных

В первой строке дано целое число  $A$  ( $-10^9 \leq A \leq 10^9$ ).

Во второй строке дано целое число  $B$  ( $-10^9 \leq B \leq 10^9$ ).

### Формат выходных данных

Выведите единственное целое число — контрольное значение, необходимое для завершения процесса.

### Система оценки

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	50	$0 \leq A \leq 10^4; B = 0$	
2	20	$0 \leq A, B \leq 10^4$	1
3	20	$-10^4 \leq A, B \leq 10^4$	1, 2
4	10	$-10^9 \leq A, B \leq 10^9$	1, 2, 3

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и предыдущих подзадач успешно пройдены, а также решение **корректно работает на примерах из условия**.

### Пример

	стандартный ввод	стандартный вывод
3		12
4		

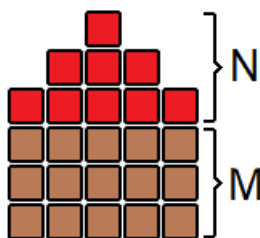
## Задача В. Новый учебный корпус

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.5 секунд  
Ограничение по памяти: 256 мегабайт

Программистов всем не хватает. Поэтому УрФУ увеличивает количество бюджетных мест на айтишных специальностях и открывает новые образовательные программы: <https://clck.ru/TU9iA>. Для комфортного обучения университет строит современный учебный корпус.

Архитекторы предложили проект будущего здания, поэтому пора оценить бюджет на строительство.

Здание можно условно разделить на две части: основание и крышу. Обе части состоят из квадратных панелей одинакового размера. Основание представляет собой прямоугольник высоты  $M$ , крыша представляет собой прямоугольный треугольник высоты  $N$ . Точную форму дома можно узнать по рисунку из проектной документации:



Ваша задача — посчитать точное количество панелей, которое необходимо для строительства.

### Формат входных данных

В первой строке дано целое число  $M$  ( $0 \leq M \leq 10^9$ ).

Во второй строке дано целое число  $N$  ( $1 \leq N \leq 10^9$ ).

### Формат выходных данных

Выведите единственное целое число — количество панелей, из которых будет состоять корпус.

### Система оценки

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	25	$M \leq 2, N \leq 2$	
2	25	$M = 0, N \leq 100$	
3	25	$M \leq 100, N \leq 100$	1, 2
4	25	$M \leq 10^9, N \leq 10^9$	1, 2, 3

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены, а также решение **корректно работает на примерах из условия**.

### Пример

стандартный ввод	стандартный вывод
2 2	10

## Задача С. Простое сложение

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

Компания NAUMEN заботится не только о своих сотрудниках, но и о растущем поколении. Поскольку программист должен прекрасно работать с числами, в компании разрабатывается новая методика обучения детей, цель которой — научить детей работать с большими числами.

Прежде чем обучить сложению, предполагается обучить так называемому «простому сложению» (патент находится на рассмотрении). Простое сложение — это сложение двух чисел без переносов разряда. Например, если сложить числа 135 и 77 с помощью простого сложения, получится число 102. Кроме того, предлагается упростить процесс обучения ещё больше следующим образом: сначала научить складывать числа, состоящих из цифр от 0 до  $K$ . Так, если  $K = 1$ , будут складываться только числа, состоящие из нулей и единиц.

Конечно же, обучение будет автоматизировано с помощью проверяющей системы. Вам поручено задание написать программу, реализующую простое сложение чисел  $A$  и  $B$ .

### Формат входных данных

В первой строке дано целое число  $K$  ( $0 \leq K \leq 9$ ).

Во второй строке дано целое число  $A$  ( $0 \leq A \leq 10^{1000}$ ).

Во третьей строке дано целое число  $B$  ( $0 \leq B \leq 10^{1000}$ ).

Гарантируется, что числа  $A$  и  $B$  состоят только из цифр от 0 до  $K$ .

### Формат выходных данных

Выведите единственное целое число — результат простого сложения чисел  $A$  и  $B$  без ведущих нулей (если сумма равна 0, выведите один ноль).

### Система оценки

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	20	$K \leq 4; A, B \leq 10^9$	
2	20	$A, B \leq 10$	
3	20	$A, B \leq 10^9$	1, 2
4	20	$A, B \leq 10^{18}$	1, 2, 3
5	20	$A, B \leq 10^{1000}$	1, 2, 3, 4

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены, а также решение **корректно работает на примерах из условия**.

### Пример

стандартный ввод	стандартный вывод
4 1 3	4

## Задача D. Привет, мир!

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

Первая программа, которую многие пишут, осваивая новый язык программирования — программа Hello, world! Однако, у Вас что-то пошло не так. Вместо того, чтобы вывести строку из 12 символов «Hello,world!», программа стала писать эту строку без остановки, даже не разделяя ни пробелами, ни переводами строк. Только когда было напечатано  $10^9$  символов, Вы нажали Ctrl+C и остановили программу.

Чтобы хоть как-то разобраться в этом хаосе, восстановите, что было выведено программой с  $L$ -го по  $R$ -й символ. Выведенные символы нумеруются, начиная с 1.

### Формат входных данных

В первой строке дано целое число  $L$ .

Во второй строке дано целое число  $R$  ( $1 \leq L \leq R \leq 10^9$ ,  $R - L \leq 10^5$ ).

### Формат выходных данных

В единственной строке выведите то, что написала программа с  $L$ -го по  $R$ -й символ. Вывод должен состоять только из символов, входящих в строку «Hello,world!» (строка не содержит пробела).

### Система оценки

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	25	$R \leq 12$	
2	25	$R \leq 24$	1
3	25	$R \leq 10^5$	1, 2
4	25	$R \leq 10^9$	1, 2, 3

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены, а также решение **корректно работает на примерах из условия**.

### Пример

стандартный ввод	стандартный вывод
1 12	Hello,world!

## Задача Е. Игнат и игрушки

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

Игнат снова поссорился со своими родителями. Он решил собрать свои любимые игрушки в портфель и уйти жить к своему другу Роме.

К сожалению Игнат может носить портфель весом не более  $W$ , а это значит, что какие-то игрушки должны остаться у родителей. Чтобы меньше грустить, он для каждой игрушки определил её вес и радость, которую она ему приносит.

Помогите Игнату заполнить портфель игрушками так, чтобы максимизировать суммарную радость.

### Формат входных данных

В первой строке заданы числа  $n, W$  ( $1 \leq n \leq 10^5, 1 \leq W \leq 2 * 10^5$ ) — количество игрушек и максимальный вес, который может переносить Игнат.

Во второй строке заданы  $n$  чисел  $w_i$  ( $1 \leq w_i \leq 2$ ) — вес  $i$ -й игрушки.

В третьей строке заданы  $n$  чисел  $c_i$  ( $1 \leq c_i \leq 10^9$ ) — радость от  $i$ -й игрушки.

### Формат выходных данных

Выведите единственное число — максимально возможную радость.

### Система оценки

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	30	$n, W \leq 1000, w_i = 1, c_i \leq 1000$	
2	30	$n \leq 1000, W \leq 2000, c_i \leq 1000$	1
3	30	$n \leq 10^5, W \leq 2 * 10^5, c_i \leq 10^4$	1, 2
4	10	$n \leq 10^5, W \leq 2 * 10^5, c_i \leq 10^9$	1, 2, 3

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены, а также решение **корректно работает на примерах из условия**.

### Пример

стандартный ввод	стандартный вывод
3 3 2 1 2 1 3 2	5

## Задача F. Транспортировка рамок

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

На день математика и механика студенты ФИИТ решили украсить стены своего факультета фотографиями любимых преподавателей. Для этого нужно напечатать фотографии и изготовить рамки для них.

Рамки толщиной в 5 миллиметров с произвольным размером делают в багетной мастерской, от которой придётся доставить до матмеха целых  $n$  рамок! Хорошо, что студенты уже прошли курс «Основы проектной деятельности» и знают, что такое управление рисками! Поэтому о том, как они будут тащить эти рамки на матмех, они подумали заранее.

Их план заключается в том, чтобы заказать рамки таких размеров, чтобы их все можно было компактно вложить друг в друга для удобства перевозки. Или более формально — для любой пары рамок  $i$  и  $j$ ,  $i$  входит в  $j$  или наоборот. Естественно, рамки при этом можно поворачивать на угол, кратный  $90^\circ$ .

Зная размеры рамок в сантиметрах, определите, получится ли вложить друг в друга все рамки описанным выше способом или нет.

### Формат входных данных

В первой строке дано число  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) — количество рамок.

В следующих  $n$  строках даны целые числа  $x_i, y_i$  ( $1 \leq x_i, y_i \leq 10^9$ ) — размеры рамок в сантиметрах.

### Формат выходных данных

Выведите «YES», если вложить рамки описанным выше способом возможно, и «NO» в противном случае.

### Система оценки

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	25	$n \leq 1000, x_i = y_i$	
2	25	$n \leq 1000$	1
3	50	$n \leq 2 \cdot 10^5$	1, 2

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены, а также решение **корректно работает на примерах из условия**.

### Примеры

стандартный ввод	стандартный вывод
3 1 1 2 2 3 3	YES
2 1 1 1 2	NO

## Задача G. Сортировка

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

В языке программирования «C++» есть переменные, условные операторы и даже массивы, но нет циклов. Чтобы обработать элементы массива, нужно обращаться к каждому по отдельности. А отсортировать массив — вообще кошмарная задача.

Допустим, вам нужно отсортировать массив `ar` длины  $n$ . Элементы этого массива пронумерованы от 0 до  $n - 1$ , обратиться к  $i$ -му элементу массива можно так: `ar[i]`. Чтобы отсортировать массив, достаточно кода, состоящего из комбинаций следующих двух строчек:

- `<if(ar[i]<ar[j])>`, где  $i$  и  $j$  — индексы элементов массива. Эта строчка описывает условие. Следующая команда выполнится только в том случае, если  $i$ -й элемент массива меньше  $j$ -го.
- `<swap(ar[i],ar[j]);>`, где  $i$  и  $j$  — индексы элементов массива. Эта строчка меняет местами элементы массива на  $i$ -й и  $j$ -й позициях.

Вы давно уже просили разработчиков языка «C++» включить в стандартную библиотеку алгоритм сортировки. Пришло время доказать, что это возможно! Напишите программу, которая для произвольного  $n$  выведет код на языке «C++», состоящий только из двух вышеуказанных команд и сортирующий массив длины  $n$ .

### Формат входных данных

В единственной строке дано число  $n$  — длина массива `ar`, который нужно отсортировать ( $2 \leq n \leq 400$ ).

### Формат выходных данных

Выведите программу на языке «C++», сортирующий массив `ar` длины  $n$ . Каждая строчка должна иметь не более одной команды, которая в точности соответствует одной из двух вышеуказанных команд, в которые вместо `<i>` и `<j>` вставлены числа. Нельзя менять регистр букв, изменять знак сравнения, обращаться к несуществующим элементам массива. Но можно вставлять пустые строки и пробелы, не разделяя одну команду на две строки и не деля слова и числа на части.

Получившаяся программа на языке «C++» должна иметь не более  $n^2$  строк и занимать не более 4 мегабайт (4 194 304 символов).

Программа будет считаться корректной, если она корректно отсортирует 1000 случайно сгенерированных массивов длины  $n$ . Гарантируется, что для каждого  $n$  набор тестовых массивов одинаков для каждой посылки.

### Система оценки

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	20	$n = 2$	
2	20	$n \leq 3$	1
3	20	$n \leq 4$	1, 2
4	20	$n \leq 40$	1, 2, 3
5	20	$n \leq 400$	1, 2, 3, 4

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены, а также решение **корректно работает на примерах из условия**.

## Пример

стандартный ввод	стандартный вывод
2	<pre>if (ar[0] &lt; ar[1])     swap(ar[0], ar[1]);</pre>

## Замечание

Обратите внимание, что ответ в примере является неверным, однако он является корректной программой на языке «C++» и приведён только для демонстрации формата вывода.



## Задача Н. Сложность решётки

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

Всем студентам ФИИТ известно, что лучшие конспекты ведёт Паша Вилкин. Такой красоты Вы ещё не видели! Ходят даже слухи, что конспекты дают своему обладателю суперсилу — сдавать все экзамены на отлично! Естественно, каждый двоечник готов на всё, чтобы заполучить себе оригиналы.

Паша, как отличник, живёт в лучшем общежитии университета. Там есть всё, что нужно для комфортной жизни, даже комната для хранения драгоценных конспектов, с крепкой решётчатой дверью. Дверь состоит из металлических прутьев, представляющих из себя бесконечно длинные и бесконечно тонкие прямые линии.

Но наш герой не уверен в том, что однажды решётку не смогут распилить. Чтобы оценить вероятность взлома, Паша ввёл понятие «сложности решётки». Под ним он подразумевает количество различных пар прутьев, которые имеют хотя бы одну общую точку.

Найдите сложность решётки!

### Формат входных данных

В первой строке дано целое число  $n$  — количество прутьев ( $1 \leq n \leq 10^5$ ).

В каждой из следующих  $n$  строк заданы целые числа  $x_{i,1}, y_{i,1}, x_{i,2}, y_{i,2}$  через пробел — координаты двух различных точек  $i$ -го прута ( $-10^9 \leq x_{i,1}, y_{i,1}, x_{i,2}, y_{i,2} \leq 10^9$ ).

### Формат выходных данных

В единственной строке выведите сложность данной решётки — количество пар  $(i, j)$  таких, что  $i < j$  и  $i$ -й прут имеет хотя бы одну общую точку с  $j$ -м.

### Система оценки

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	10	$n \leq 10$ , для любых 2 прямых угол наклона различный	
2	25	$n \leq 10^3$ , для любых 2 прямых точек пересечения меньше 2	1
3	25	$n \leq 10^3$	1, 2
4	40	$n \leq 10^5$	1, 2, 3

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены, а также решение **корректно работает на примерах из условия**.

### Пример

стандартный ввод	стандартный вывод
3 1 1 2 1 1 1 2 2 1 1 1 2	3