

## Общая информация по задачам олимпиады

### **Ограничение на размер исходного кода программы-решения**

Во всех задачах размер файла с исходным кодом решения не должен превышать 256 КБ.

### **Процесс тестирования**

Обратите внимание, что у каждой подзадачи есть список необходимых подзадач. Подзадача будет тестироваться, если все тесты во всех ее необходимых подзадачах пройдены.

### **Система оценки**

На нашем туре есть два типа оценки подзадач: «тест» и «подзадача». «Тест» означает, что в подзадаче баллы за каждый тест выставляются независимо, и сумма баллов за все тесты равна числу баллов за всю подзадачу. «Подзадача» означает, что баллы начисляются только за полностью пройденную подзадачу.

### **Доступ к результатам проверки решений задач во время тура**

Во всех подзадачах с оценкой типа «подзадача» сообщаются вердикты проверки по каждому тесту.

Во всех подзадачах с оценкой типа «тест» сообщается суммарный балл по всем пройденным тестам этой подзадачи.

## Задача А. Ётта

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Нияз с Ниязом в ожидании результатов олимпиады решили поиграть в Ётту. В этой игре участники выкладывают карточки в линии по определенным правилам. Карточки бывают четырех разных форм, цветов и достоинств. За выложенную линию игрок получает определенное количество очков.

Ниязу обычные правила показались слишком сложными, и он предложил убрать у карточек цвет и форму, оставив лишь достоинство. А Ниязу такие правила показались слишком простыми, и он предложил увеличить количество различных достоинств карточек до 100.

В модифицированных правилах линия считается корректной, если либо она состоит из карточек одного достоинства, либо все карточки в ней имеют различные достоинства. Если игрок выложил некорректную линию, то он получает  $-1$  очков за невнимательность. За корректную линию начисляется количество очков равное сумме достоинств всех карточек в ней. При этом, если достоинство каждой следующей карточки на один больше предыдущей, то очки за такую линию умножаются на два.

Но при этих правилах Ниязу сложно быстро посчитать очки за выложенную линию. Поэтому он просит Вас ему помочь и написать программу, которая по заданной линии из  $n$  карточек определяет количество очков за нее.

### Формат входных данных

В первой строке задано целое число  $n$  — количество карточек на столе ( $2 \leq n \leq 100$ ). В следующей строке задано  $n$  целых чисел  $a_i$  — достоинства карточек на столе ( $1 \leq a_i \leq 100$ ). Числа  $a_i$  заданы в порядке неубывания ( $a_i \leq a_{i+1}$ ).

### Формат выходных данных

Выведите единственное число — сколько очков получит игрок, выложивший данную линию.

### Система оценки

Номер подзадачи	Баллы	Ограничения	Оценка	Необх. подзадачи
1	17	$2 \leq n \leq 3$	подзадача	—
2	34	Все $a_i$ различны	подзадача	—
3	49	—	подзадача	1 и 2

### Примеры

стандартный ввод	стандартный вывод
3 1 3 5	9
3 2 2 2	6
3 1 2 3	12
3 1 1 2	-1

## Задача В. Антиэлементы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

В 2345 году космическая экспедиция с кодовым названием «New universes» совершила прорыв в истории человечества — нашла портал в антивселенную. Как бы это ни было странно, в антивселенной практически все как у нас — есть свои законы физики, химии, математики. Только вот время идет вспять. И массы у всего отрицательные. И на ноль можно делить. В общем, интересная вселенная, по просторам которой уже бороздит наша земная экспедиция.

И вот когда экспедиция добралась до Антиземли, молодой алхимик Марат сразу начал изучать все антиэлементы из которых состоят антивещества. Конечно, к концу экспедиции все элементы ему найти не удалось, но часть из них он все же успел изучить и даже сделать какие-то выводы.

Марат заметил, что в антивселенной всего ровно  $n$  антиэлементов с различными антизарядами — натуральными числами от 1 до  $n$ . Также у каждого антиэлемента есть своя антимасса — целое число от 1 до  $m$ .

За то время, пока экспедиция находилась на Антиземле, Марат нашел информацию только о  $k$  антиэлементах — их антизаряд и антимассу. Теперь Марат хочет узнать, может ли быть так, что для каждой пары антиэлементов верно, что у антиэлемента с бóльшим антизарядом больше антимасса.

### Формат входных данных

В первой строке заданы три целых числа  $n$ ,  $m$  и  $k$  ( $1 \leq n, m \leq 10^9$ ,  $0 \leq k \leq n$ ) — число различных антиэлементов, максимальная антимасса антиэлемента и число найденных антиэлементов.

Каждая из следующих  $k$  строк содержит по два целых числа  $i$  ( $1 \leq i \leq n$ ) и  $a_i$  ( $1 \leq a_i \leq m$ ) — антизаряд антиэлемента и его антимасса. Антиэлементы заданы в порядке возрастания их антизарядов.

### Формат выходных данных

В выходной файл выведите “YES”, если существует такой набор из  $n$  антиэлементов, который не противоречит текущей информации и для которого верно, что антиэлемент с бóльшим антизарядом имеет бóльшую антимассу, и “NO”, иначе.

### Система оценки

Номер подзадачи	Баллы	Ограничения		Оценка	Необх. подзадачи
		$n$ и $m$	$k$		
1	29	$n, m \leq 10^3$	$k \leq 10^3$	подзадача	—
2	32	$n, m \leq 10^5$	$k \leq 10^5$	подзадача	1
3	39	$n, m \leq 10^9$	$k \leq 10^5$	подзадача	1 и 2

### Примеры

стандартный ввод	стандартный вывод
5 6 3 1 1 3 4 5 6	YES
10 15 4 2 2 3 4 6 10 7 10	NO

### **Пояснения к примерам**

В первой примере подходящие антиэлементы могут иметь антимассы: 1, 2, 4, 5, 6.

Во втором примере антиэлементы 6 и 7 оба имеют антимассу 10, а значит, антимассы не могут строго возрастать с возрастанием антизаряда.

## Задача С. Кусудамы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Эдвард очень любит бумагу и все, что с ней связано. Недавно он научился делать кусудамы — модульные шары из бумаги в технике оригами. Эдварду так понравились кусудамы, что он решил подарить всем своим  $n$  друзьям по одному такому шару. Мальчик хочет, чтобы все кусудамы были разных цветов, и  $i$ -я кусудамы была сделана из  $a_i$  одинаковых по цвету и размеру квадратных модулей ( $1 \leq i \leq n$ ). У Эдварда после прошлых поделок остались куски бумаги  $m$  цветов,  $c_j$  квадратов  $j$ -го цвета ( $1 \leq j \leq m$ ).  $k$ -й квадрат  $j$ -го цвета имеет длину стороны  $b_{jk}$ . Из каждого квадрата мальчик умеет вырезать один квадрат меньшей стороны с целой длиной. Эдвард не любит работать с маленькими бумажками, поэтому хочет, чтобы размер самой маленькой кусудамы, то есть размер модулей, из которых она сделана, получился как можно больше. Ваша задача — найти наибольший размер модулей наименьшей кусудамы, который может получиться у Эдварда. Гарантируется, что из данного набора бумаги можно сделать все  $n$  кусудам.

### Формат входных данных

В первой строке заданы два целых числа  $n$  и  $m$  ( $1 \leq n \leq m \leq 2 \cdot 10^5$ ) — количество кусудам и количество цветов бумаги, соответственно. Во второй строке задано  $n$  целых чисел  $a_i$  — число модулей, необходимое для  $i$ -й кусудамы ( $1 \leq a_i \leq 10^3$ ). Далее в  $2m$  строках описания цветов. В строке  $2j + 1$  — одно целое число  $c_j$  количество модулей  $j$ -го цвета ( $1 \leq c_j \leq 10^3$ ,  $\sum c_j \leq 2 \cdot 10^5$ ). В следующей строке  $c_j$  чисел  $b_{jk}$  — длина стороны каждого модуля  $j$ -го цвета ( $1 \leq b_{jk} \leq 10^6$ ).

### Формат выходных данных

В единственной строке выведите одно целое число — максимальный возможный размер модулей наименьшей кусудамы.

### Система оценки

Номер подзадачи	Баллы	Ограничения				Оценка	Необх. подзадачи
		$n$	$m$	$c_j$	$b_{jk}$		
1	9	$n = 7$	$m = 7$	$c_j \leq 100$	$b_{jk} \leq 10^3$	подзадача	—
2	33	$n \leq 500$	$m \leq 5\,000$	—	—	подзадача	1
3	21	$n \leq 500$	$m \leq 10^5$	—	—	подзадача	1, 2
4	37	$n \leq 2 \cdot 10^5$	$m \leq 2 \cdot 10^5$	—	—	подзадача	1, 2, 3

## Примеры

стандартный ввод	стандартный вывод
2 2 2 3 3 10 1 5 3 10 15 10	5
3 5 5 2 3 2 3 3 1 10 3 3 2 4 6 3 4 2 4 4 1 3 1 1 1	2

## Пояснение к примеру

В первом примере первая кусудама может быть сделана из двух квадратов первого цвета размерами 5 и 10 (из квадрата размером 10 будет вырезан модуль размером 5), вторая — из трех квадратов второго цвета размерами 10, 15 и 10 (из квадрата 15 будет вырезан квадрат с длиной стороны 10). Тогда первая кусудама будет наименьшей и иметь размер 5.

## Задача D. Оля и загадочное озеро

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Оля живет на берегу красивого озера. Как-то утром она заметила, что дальний берег стал чуть-точку ближе. Так повторялось в течение нескольких дней, и она поняла, что берега идут навстречу друг к другу! Более того, с одинаковой скоростью. Оля задалась вопросом, сколько новых мелких озер образуется, когда берега сойдутся.

Представим дальний и ближний берега озера как две непересекающихся ломаных на плоскости, причем одна находится строго над другой, верхняя ломаная строго выше оси абсцисс (ось  $Ox$ ), а нижняя — строго ниже. Пусть, дальний берег обозначит верхнюю ломаную, а ближний — нижнюю. В каждой ломаной  $x$ -координаты точек строго возрастают. У начал и концов ломаных  $x$ -координаты совпадают. Новое озеро — область между двумя ломаными, ограниченная со всех сторон ее отрезками или концами ломаной.

Помогите Оле узнать, сколько новых озер образуется, когда дальний берег соприкоснется с ближним (верхняя ломаная вертикально опустится на нижнюю).

### Формат входных данных

В первой строке входных данных заданы два целых числа  $n$  и  $m$  — количество точек в верхней и нижней ломаных, соответственно ( $2 \leq n, m \leq 10^5$ ).

В следующих  $n$  строках заданы пары целых чисел  $x_{up,i}$  и  $y_{up,i}$  — координаты точек верхней ломаной по одной на строку ( $|x_{up,i}| \leq 10^6$ ,  $1 \leq y_{up,i} \leq 10^6$ ),  $x$ -координаты строго возрастают ( $x_{up,i} < x_{up,i+1}$ ).

В следующих  $m$  строках заданы пары целых чисел  $x_{low,i}$  и  $y_{low,i}$  — координаты точек нижней ломаной по одной на строку ( $|x_{low,i}| \leq 10^6$ ,  $-10^6 \leq y_{low,i} \leq -1$ ),  $x$ -координаты строго возрастают ( $x_{low,i} < x_{low,i+1}$ ).

Ломаные не касаются и не пересекаются. У начал и концов ломаных  $x$ -координаты совпадают ( $x_{up,1} = x_{low,1}$  и  $x_{up,n} = x_{low,m}$ ).

### Формат выходных данных

В выходной файл требуется вывести единственное число — количество новых озер, которое образуется, когда два берега сойдутся.

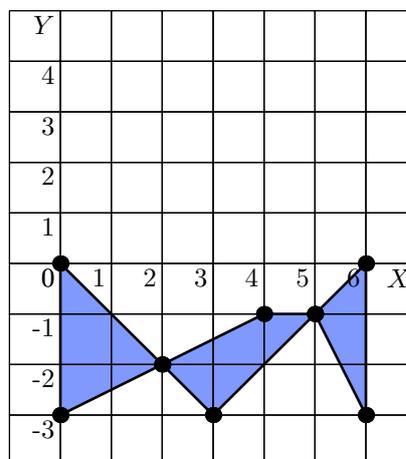
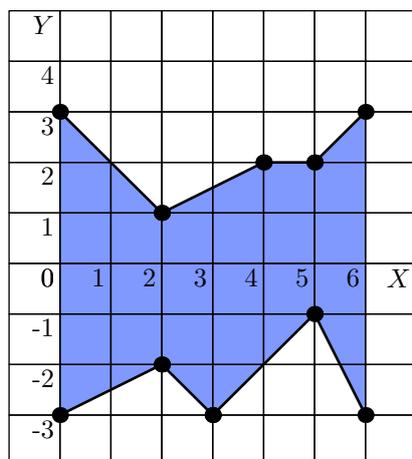
### Система оценки

Номер подзадачи	Баллы	Ограничения		Оценка	Необх. подзадачи
		$n$ и $m$	Дополнительные		
1	22	$n, m \leq 10^5$	$n = m, x_{up,i} = x_{low,i}$	подзадача	—
2	34	$n, m \leq 100$	нет параллельных отрезков	подзадача	—
3	44	$n, m \leq 10^5$	—	подзадача	1 и 2

## Примеры

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

## Пояснение к примеру



На первой картинке изображено озеро до загадочных событий. На второй — как в результате загадочных событий образовалось три новых озера. Картинки соответствуют первому примеру.

## Задача Е. Рома и битовые операции

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2.5 секунд
Ограничение по памяти:	512 мегабайт

Рома обожает компьютеры и все, что с ними связано, поэтому в последнее время он тренируется выполнять битовые операции, которые, как известно, являются основными операциями, которые может совершать компьютер. Рома очень талантливый и упорный, поэтому он решил выполнять операции не с одним числом, а сразу с целым подотрезком массива. Но помимо талантливости и упорства Рома обладает еще одним качеством — торопливостью, что зачастую мешает ему во время тренировок. Бывает, что Рома выполняет операцию, которую выполнять не хотел и позже решает ее отменить. Рома обратился к вам за помощью. Он просит написать программу, которая бы проверяла все его вычисления.

Более формально, вам дан массив целых чисел  $a_i$  длины  $n$ , и вам необходимо выполнить  $m$  запросов. Запросы бывают трех типов:

1. По заданным  $l$ ,  $r$  и  $x$ , для всех  $i$  от  $l$  до  $r$ , включительно, выполнить операцию  $a_i = a_i \wedge x$ , где  $\wedge$  — побитовое И.
2. По заданным  $l$  и  $r$  посчитать  $a_l \vee a_{l+1} \vee \dots \vee a_r$ , где  $\vee$  — побитовое ИЛИ.
3. Отменить  $k$ -й запрос. Гарантируется, что отменяемый запрос является запросом первого типа и имеет меньший номер, чем номер текущего запроса. Также гарантируется, что никакой запрос не отменяется дважды.

### Формат входных данных

В первой строке заданы числа  $n$  и  $m$  — длина массива и количество запросов, соответственно ( $1 \leq n, m \leq 2 \cdot 10^5$ ).

Вторая строка содержит  $n$  чисел  $a_i$ , разделенных пробелом ( $0 \leq a_i < 2^{30}$ ).

В последующих  $m$  строках содержатся описания запросов. Первое число  $t_i$  в каждой строке обозначает тип запроса ( $1 \leq t_i \leq 3$ ). Далее следуют:

1. ( $t_i = 1$ ) Три числа  $l$ ,  $r$ ,  $x$  — левая и правая границы подотрезка и число с которым необходимо выполнить побитовое И ( $1 \leq l \leq r \leq n$ ;  $0 \leq x < 2^{30}$ ).
2. ( $t_i = 2$ ) Два числа  $l$ ,  $r$  — левая и правая границы подотрезка на котором необходимо посчитать побитовое ИЛИ ( $1 \leq l \leq r \leq n$ ).
3. ( $t_i = 3$ ) Одно число  $k$  — номер запроса, который необходимо отменить ( $1 \leq k \leq m$ ).

### Формат выходных данных

Для каждого запроса второго типа вывести в отдельной строке побитовое ИЛИ всех чисел на подотрезке, указанном в запросе.

### Система оценки

Номер подзадачи	Баллы	Ограничения			Оценка	Необх. подзадачи
		$n$	$m$	$t_i$		
1	8	$n \leq 2000$	$m \leq 2000$	$t_i \neq 3$	подзадача	—
2	17	$n \leq 2 \cdot 10^5$	$m \leq 2 \cdot 10^5$	$t_i \neq 3$	тест	1
3	14	$n \leq 600$	$m \leq 600$	—	подзадача	—
4	15	$n \leq 5 \cdot 10^4$	$m \leq 2000$	—	подзадача	1 и 3
5	19	$n \leq 5 \cdot 10^4$	$m \leq 5 \cdot 10^4$	—	подзадача	1, 3 и 4
6	27	$n \leq 2 \cdot 10^5$	$m \leq 2 \cdot 10^5$	—	тест	1, 3, 4 и 5

## Примеры

стандартный ввод	стандартный вывод
4 4 1 1 1 1 2 1 2 1 1 3 2 2 2 3 2 1 4	1 0 1
5 5 1 2 3 4 5 2 2 4 1 2 3 1 2 2 4 3 2 2 2 4	7 5 7

## Пояснение к примеру

Пояснения для второго теста из условия:

1.  $[1, 2, 3, 4, 5]$  и  $2 \vee 3 \vee 4 = 7$  — ответ на первый запрос
2.  $[1, 2, 3, 4, 5] \rightarrow [1, 0, 1, 4, 5]$  — после второго запроса.
3.  $[1, 0, 1, 4, 5]$  и  $0 \vee 1 \vee 4 = 5$  — ответ на третий запрос
4.  $[1, 0, 1, 4, 5] \rightarrow [1, 2, 3, 4, 5]$  — после четвертого запроса.
5.  $[1, 2, 3, 4, 5]$  и  $2 \vee 3 \vee 4 = 7$  — ответ на пятый запрос