

Ты решал(а) на школьном этапе вчера задачи по комплекту 9-11. Верно? Если нет – этот комплект заданий не для тебя – обратись сейчас к организатору.

Если все верно, то вот твоя ссылка на проверяющую систему:

[contest.yandex.ru/contest/56232](https://contest.yandex.ru/contest/56232)

## Задача 1. Геометрический этюд

*Имя входного файла:* `etude.in`  
*Имя выходного файла:* `etude.out`  
*Ограничение времени* `1 секунда на тест`  
*Ограничение по памяти* `256 Мб`

Айвар учится в восьмом классе. Он всерьез увлекается двумя школьными предметами – геометрией и технологией. Сегодня он решает геометрический этюд. Из проволоки Айвар вырезал три прямолинейных отрезка длиной  $a$ ,  $b$  и  $c$  сантиметров соответственно. А из этих трех кусочков проволоки Айвар формирует на столе треугольник. Далее он совершает "события". За одно событие Айвар укорачивает каждый кусочек проволоки (т.е. каждую из сторон треугольника) на 1 см и вновь пытается сложить из таких укороченных кусочков проволоки треугольник. Спустя какое минимальное количество событий из имеющихся кусочков проволоки уже нельзя будет сложить треугольник?

### **Формат входных данных:**

На вход программе подаются три натуральных числа  $a$ ,  $b$  и  $c$  ( $1 \leq a, b, c \leq 10^9$ ). Гарантируется, что из отрезков заданной длины треугольник составить можно.

### **Формат выходных данных:**

В качестве результата Ваша программа должна вывести одно целое число: минимальное количество событий, после которых, из имеющихся кусочков проволоки уже нельзя будет сложить треугольник.

### **Пример входных данных и верного результата:**

Входные данные	Ответ программы
10 18 12	4

### **Пояснения к примеру:**

После первого события длины прямолинейных кусочков проволоки стали такими: 9, 17, 11. После второго – 8, 16, 10. После третьего – 7, 15, 9. После четвертого – 6, 14, 8, а из таких кусочков треугольник не получится составить.

## Задача 2. Исполнитель "Корректор"

Имя входного файла: *editor.in*  
Имя выходного файла: *editor.out*  
Ограничение времени: *1 секунда на тест*  
Ограничение по памяти: *128 Мб*

Исполнитель «Корректор» обрабатывает только маленькие буквы латинского алфавита. Исполнителя "Корректор" умеет:

- подсчитывать количество символов в слове.
- вставлять буквы в слове в заданное место;
- заменять одну букву на другую.

Вам задан алгоритм.

- Вычисляется длина исходной цепочки символов. Если она чётна, то в середину цепочки добавляется буква 'a'. Если длина исходной цепочки нечётна, то в начало цепочки добавляется буква 'b'.

- В полученной цепочке символов каждая буква заменяется буквой, следующей за ней в английском алфавите ('a' – на 'b', 'b' – на 'c' и так далее, 'z' – на 'a').

Получившаяся таким образом цепочка является результатом работы алгоритма.

Например, если применить данный алгоритм к цепочке 'cat', то получится цепочка 'cdbu'. Если применить алгоритм к этому результату ещё раз, то получится цепочка 'debcv'.

Вам необходимо для заданной цепочки (назовём ее базовой) ответить на один из 2 вопросов:

1. Какая цепочка символов получится, если дважды применить этот алгоритм к базовой?
2. В результате двукратной обработки какой исходной цепочки, получилась базовая? (гарантируется, что такая исходная цепочка существует)

Замечание Алфавит английского языка: a b c d e f g h i j k l m n o p q r s t u v w x y z

### Формат входных данных:

На вход программе в первой строке поступает цепочка символов. Длина цепочки не превышает 100 символов. Во второй строке записан цифрой номер вопроса 1 или 2.

### Формат выходных данных:

В качестве результата Ваша программа должна вывести ответ на вопрос.

### Примеры входных данных и верных результатов:

Входные данные	Ответ программы
sc 1	cuce
cuce 2	sc

### Система оценки:

Если Ваша программа умеет отвечать только на один вопрос, Вы получите за решение 50 баллов.

### Задача 3. Лампы

Имя входного файла: *lamps.in*  
Имя выходного файла: *lamps.out*  
Ограничение времени: 3 секунды на тест  
Ограничение по памяти: 256 Мб

У Камиля на рабочем столе находятся  $n$  листов бумаги, пронумерованных от 1 до  $n$ . Чтобы было комфортно писать на  $i$ -м листе бумаги, надо чтобы на него попадало как минимум  $a_i$  света. Для того, чтобы освещать листы, у Камиля есть  $q$  ламп,  $j$ -я лампа освещает листы с номерами от  $l_j$  до  $r_j$ . Если на  $j$ -ю лампу подать  $x$  электричества, то освещённость каждого листа бумаги с номером  $i$ , где  $l_j \leq i \leq r_j$ , увеличится ровно на  $x$ .

Камиль не любит просто так тратить электричество, поэтому он просит вас найти такой минимальный  $x$ , что если подать на каждую лампу ровно  $x$  электричества, то ему будет комфортно писать на хотя бы  $k$  листах бумаги. Изначально освещённость каждого листа равна нулю.

#### Формат входных данных:

В первой строке входного файла «lamps.in» дано два целых числа  $n$  и  $k$  ( $1 \leq k \leq n \leq 10^5$ ) — количество листов бумаги на рабочем столе и количество листов бумаги, которые необходимо осветить, соответственно.

Во второй строке дано  $n$  целых чисел  $a_1, a_2, a_3, \dots, a_n$  ( $0 \leq a_i \leq 10^9$ ), где  $a_i$  — необходимый уровень освещённости для  $i$ -го листа бумаги.

В третьей строке дано целое число  $q$  ( $1 \leq q \leq 10^5$ ) — количество ламп.

В следующих  $q$  строках описываются сами лампы, в  $j$ -й строке дано два целых числа  $l_j$  и  $r_j$  ( $1 \leq l_j \leq r_j \leq n$ ) — границы отрезка листов бумаг, которые освещает  $j$ -я лампа.

#### Формат выходных данных:

В выходной файл «lamps.out» выведите одно минимальное целое неотрицательное число  $x$ , что если подать на каждую лампу по  $x$  электричества, то будет комфортно писать на хотя бы  $k$  листах бумаги. Если такого  $x$  не существует, выведите  $-1$ .

#### Система оценки:

Тесты к этой задаче состоят из нескольких групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов всех необходимых групп.

Группа	Баллы	Дополнительные ограничения	Необходимые группы	Комментарий
0	0	—	—	Тесты из условия
1	25	$n \leq 100, q \leq 100, a_i \leq 100$	—	В этой группе $a_i \leq 100$ , то есть необходимый уровень освещённости каждого листа не больше 100.

2	20	$n \leq 1000, q \leq 1000, k = n$	—	
3	30	$q=1$	—	
4	25	—	0, 1, 2, 3	Полные ограничения

**Примеры входных данных и верных результатов:**

Входные данные	Ответ программы
6 3 7 5 9 8 3 6 2 2 4 4 6	5
10 9 10 8 9 5 3 1 6 7 4 2 3 3 5 1 4 7 9	-1

**Замечание:**

В первом примере первая лампа отвечает за листы 2, 3, 4, а вторая — за 4, 5, 6. Если подать  $x = 5$  электричества на каждую лампу, то освещённости листов станут равны 0, 5, 5, 10, 5, 5 соответственно. В этом случае листы 1, 4, 5 получают необходимую освещённость. Можно показать, что при  $x < 5$  не получится осветить хотя бы 3 листа необходимым образом.

## Задача 4. Катет

Имя входного файла: *cathetus.in*  
Имя выходного файла: *cathetus.out*  
Ограничение времени: *2 секунды на тест*  
Ограничение по памяти: *256 Мб*

Вам дано одно целое число  $x$ . Найдите количество различных прямоугольных треугольников ненулевой площади с целыми сторонами, что один из катетов равен  $x$ . Два треугольника считаются различными, если их нельзя наложить друг на друга при помощи параллельных сдвигов, поворотов и отражений.

### Формат входных данных:

В первой строке входного файла «cathetus.in» дано одно целое число  $t$  ( $1 \leq t \leq 5$ ) — количество наборов входных данных. В каждой из последующих  $t$  строк дано одно целое число  $x$  ( $1 \leq x \leq 10^9$ ) — длина катета.

### Формат выходных данных:

В выходной файл «cathetus.out» выведите  $t$  строк — количество прямоугольных треугольников с катетом  $x$  для каждого набора входных данных.

### Система оценки:

Тесты к этой задаче состоят из нескольких групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов всех необходимых групп.

Группа	Баллы	Дополнительные ограничения	Необходимые группы	Комментарий
0	0	—	—	Тесты из условия
1	30	$x \leq 40$	0	Можно доказать, что при $x \leq 40$ , длины всех сторон всех подходящих треугольников не превышают 1000.
2	25	$x \leq 1000$	0, 1	
3	15	$x \leq 10^5$	0, 1, 2	
4	10	$x=2^k, 0 \leq k < 30$	—	
5	20	$x \leq 10^9$	0-4	

### Пример входных данных и верного результата:

Входные данные	Ответ программы
2	4
15	0
2	

### Замечание:

Для  $x = 15$  существует ровно 4 различных прямоугольных треугольника с целочисленными сторонами: (8, 15, 17), (15, 20, 25), (15, 36, 39) и (15, 112, 113).

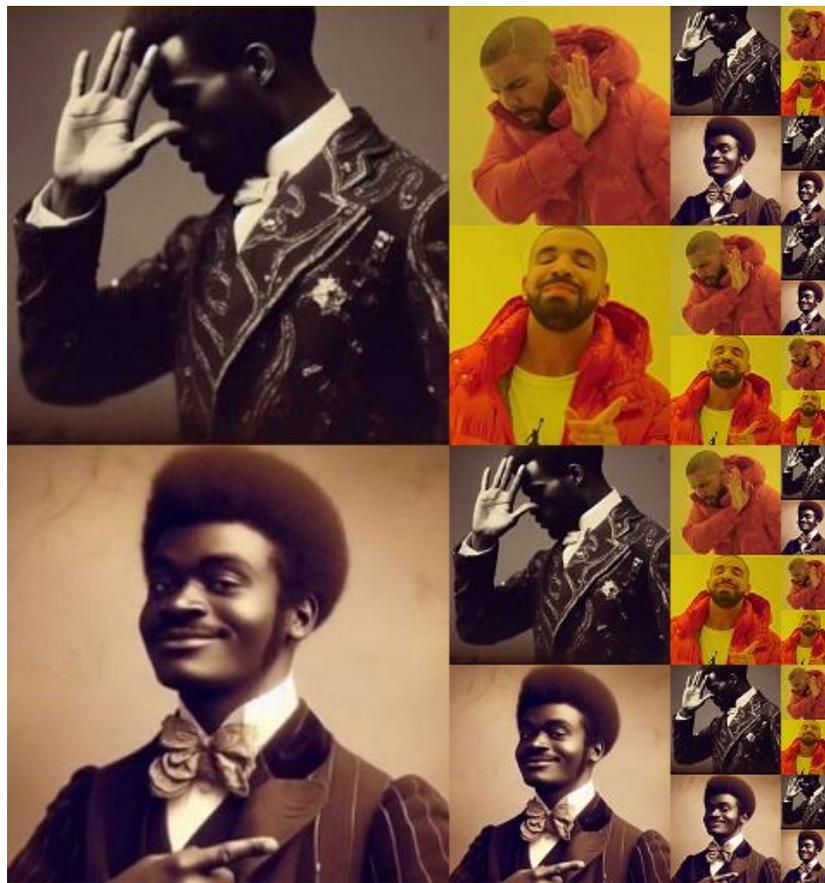
## Задача 5. Рекурсивный мем (100 баллов)

Имя входного файла: *mete.in*  
Имя выходного файла: *mete.out*  
Ограничение времени: 2 секунды на тест  
Ограничение по памяти: 256 Мб

Прямоугольное поле высоты  $2^N$  и ширины  $2^N-1$  разбито на квадратные единичные клеточки. Сначала выделяется прямоугольная область высоты  $2^N$  и ширины  $2^{N-1}$ , прилегающая к левому краю поля. Затем для каждой прямоугольной области слева приклеиваются две области вдвое меньшего размера. Процесс продолжается до тех пор, пока размеры областей не станут  $2 \times 1$ , и всё поле не будет заполнено.

Далее, клеточки поля красятся в чёрный и жёлтый цвет по следующему принципу. Самая большая область красится в чёрный цвет. Затем для каждой области нижняя из прилегающих слева областей красится в тот же цвет, что и текущая область, а верхняя – в противоположный.

Результат раскраски для  $N = 4$  иллюстрирует следующая картинка:



У Вас есть  $q$  запросов, каждый из которых задаёт произвольный прямоугольник внутри поля. Посчитайте для каждого запроса количество чёрных клеточек, которые покрывает соответствующий прямоугольник.

### Формат входных данных:

В первой строке входного файла `mete.in` дано два целых числа  $N$  и  $q$  ( $1 \leq N \leq 30$ ,  $1 \leq q \leq 10^4$ ).

Далее идут  $q$  строк, в  $i$ -й из них дано описание  $i$ -го запроса: четыре целых числа  $r_i$ ,  $c_i$ ,  $h_i$  и  $w_i$ . Здесь  $r_i$  и  $c_i$  – номер строки и номер столбца клеточки, в которой находится левый верхний угол прямоугольника, а  $h_i$  и  $w_i$  – высота и ширина прямоугольника соответственно ( $1 \leq r_i, h_i, r_i+h_i-1 \leq 2^N$ ,  $1 \leq c_i, w_i, c_i+w_i-1 \leq 2^N-1$ ,  $1 \leq i \leq q$ ).

### Формат выходных данных:

В выходной файл `memo.out` выведите  $q$  строк, в  $i$ -й из них должно быть одно целое число – ответ на  $i$ -й запрос.

Отметим, что ответы могут получиться очень большими и не поместиться в 32-битную переменную. Пожалуйста, используйте 64-битный тип данных.

### Система оценки:

Тесты к этой задаче состоят из нескольких групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов всех необходимых групп.

Группа	Баллы	Ограничения	Необходимые группы	Комментарий
0	0	–	–	Пример из условия
1	30	$N \leq 5, q \leq 1000$	0	
2	20	$N \leq 10, q \leq 10^4$	0, 1	
3	20	$N \leq 15, q \leq 100$	0, 1	
4	20	$N \leq 20, q \leq 10^4$	0, 1, 2, 3	
5	10	$N \leq 30, q \leq 10^4$	0, 1, 2, 3, 4	Полные ограничения

### Примеры файлов с входными данными и файлов с результатом:

<code>memo.in</code>	<code>memo.out</code>
3 4	44
1 1 8 7	14
2 4 6 4	10
4 1 2 6	3
6 3 1 3	

### Замечание:

Прямоугольники, для которых нужно посчитать количество чёрных клеточек, показаны на следующем рисунке:

