

## Общая информация по задачам второго тура

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

В течение тура можно не более 10 раз по каждой задаче запросить информацию о результатах оценивания решения на тестах жюри.

### Ограничение на размер исходного кода программы-решения

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

### Процесс тестирования

Во всех задачах, очередная подзадача будет тестироваться, только если пройдены все тесты всех предыдущих подзадач.

### Ограничения

Задача	Ограничение по времени	Ограничение по памяти	Получение результатов во время тура
5. Поможем дикой природе	2 секунды	512 МБ	Для каждой подзадачи сообщаются баллы за эту подзадачу и результат проверки программы на каждом тесте.
6. Подводная лодка	2 секунды	512 МБ	Для каждой подзадачи сообщаются баллы за эту подзадачу и результат проверки программы на каждом тесте.
7. Фонари	2 секунды	512 МБ	Для подзадач 1 и 2 сообщаются баллы за эту подзадачу и результат проверки программы на каждом тесте. Для подзадач 3, 4 и 5 сообщаются только баллы за эту подзадачу.
8. Сигнализация	4 секунды	512 МБ	Для подзадач 1 и 2 сообщаются баллы за эту подзадачу и результат проверки программы на каждом тесте. Для подзадач 3, 4 и 5 сообщаются только баллы за эту подзадачу.

## Задача 5. Поможем дикой природе

Имя входного файла:	grants.in
Имя выходного файла:	grants.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 МБ

Фонд изучения дикой природы в течение  $t$  лет ежегодно выделяет денежные гранты в поддержку исследований северной фауны. На гранты претендуют три организации, одна из которых занимается изучением тюленей, вторая — оленей, третья — белых медведей.

Для упрощения бухгалтерского учёта фонд принял следующие решения:

- размер любого гранта в денежных единицах должен быть степенью числа 2, то есть равен  $2^k$  для некоторого целого  $k \geq 0$ ;
- все гранты, получаемые одной организацией в одном году, должны иметь различные размеры.

В  $i$ -м году фонд планирует полностью распределить  $n_i$  денежных единиц, выделенных на гранты. Сравнивать результативность использования средств возможно только для грантов одинакового размера, выделенных каждой из трёх организаций. Такие гранты называются *целевыми*. Распределение денежных единиц на гранты между тремя организациями считается оптимальным, если как можно большая часть общей суммы выделена на целевые гранты.

Например, если в текущем году на все гранты выделено 47 денежных единиц, то оптимальным вариантом распределения будет: выделить каждой из организаций целевые гранты размерами по 2 и 8 денежных единиц, что составит в сумме 30 единиц. Остальные 17 единиц можно распределить, например, выделив первой организации 16 денежных единиц, а третьей — 1 денежную единицу. Выделить более 30 денежных единиц на целевые гранты, распределяя 47 денежных единиц, нельзя.

Требуется написать программу, которая по заданной в  $i$ -м году общей сумме грантов  $n_i$  определяет, сколько денежных единиц следует выделить каждой из трёх организаций при оптимальном распределении грантов.

### Формат входных данных

В первой строке входных данных записано целое число  $t$  — количество лет ( $1 \leq t \leq 100$ ). В каждой из последующих  $t$  строк записано целое число  $n_i$  — общая сумма грантов, которую необходимо полностью распределить в  $i$ -м году.

### Формат выходных данных

Выходные данные должны содержать  $t$  строк по три целых числа в каждой — суммы грантов, которые следует выделить каждой из трёх организаций в соответствующий год. Если оптимальных вариантов распределения несколько, необходимо вывести любой из них.

### Таблица системы оценивания

Номер подзадачи	Баллы	Ограничения
		$n_i$
1	16	$1 \leq n_i < 64$
2	33	$1 \leq n_i < 512$
3	17	$1 \leq n_i < 2^{17}$
4	34	$1 \leq n_i < 2^{60}$

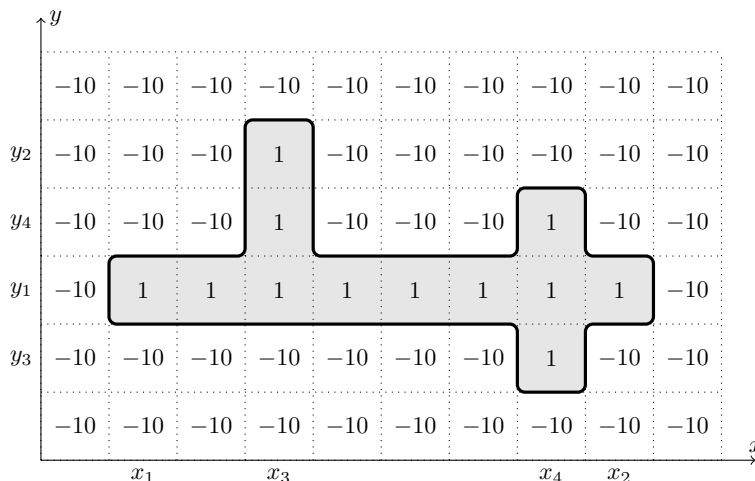
### Примеры

	grants.in	grants.out
3		0 0 4
4		7 7 7
21		26 10 11
47		

## Задача 6. Подводная лодка

Имя входного файла: `submarine.in`  
Имя выходного файла: `submarine.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 МБ

Подводная лодка легла на грунт на мелководье. Для её обнаружения используются данные спутника, который с высокой точностью измеряет отклонение высоты поверхности воды от среднего уровня моря. Снимок, получаемый со спутника, представляет собой массив из  $h$  строк по  $w$  элементов в каждой строке.



Введём на снимке систему координат с осью абсцисс, направленной вдоль строк снимка слева направо, и осью ординат, направленной вдоль столбцов снимка снизу вверх. Потенциальное изображение подводной лодки представляет собой любое множество элементов массива, состоящее из следующих частей:

- «корпус» — полоса из элементов с координатами от  $(x_1, y_1)$  до  $(x_2, y_1)$ , где  $x_1 < x_2$ ;
- «рубка» — полоса из элементов с координатами от  $(x_3, y_1)$  до  $(x_3, y_2)$ , где  $x_1 \leq x_3 < x_2$ ;  $y_1 \leq y_2$ ;
- «хвост» — полоса из элементов с координатами от  $(x_4, y_3)$  до  $(x_4, y_4)$ , где  $x_3 < x_4 \leq x_2$ ;  $y_3 \leq y_1 \leq y_4$ .

Поскольку подводная лодка находится вблизи поверхности в районе с сильным течением, уровень воды над ней немного повышается. Поэтому изображением подводной лодки на снимке будем считать потенциальное изображение с максимально возможной суммой входящих в него элементов массива.

Требуется написать программу, которая находит на снимке изображение подводной лодки и выводит сумму его элементов.

### Формат входных данных

Для сжатия передаваемых со спутника данных каждый элемент снимка кодируется строчной буквой английского алфавита. Первая строка входных данных содержит число  $k$  — количество использованных для кодирования букв ( $k \leq 26$ ). Вторая строка входных данных содержит  $k$  целых чисел  $c_i$  — значения отклонений соответствующих каждому кодовому символу по порядку букв в английском алфавите от 1 до  $k$ -й.

Третья строка входных данных содержит числа  $h$  и  $w$  — размеры снимка. Последующие  $h$  строк содержат по  $w$  символов — кодовые значения элементов снимка.

### Формат выходных данных

Выходные данные должны содержать единственное целое число — сумму элементов массива, соответствующих изображению подводной лодки.

### Таблица системы оценивания

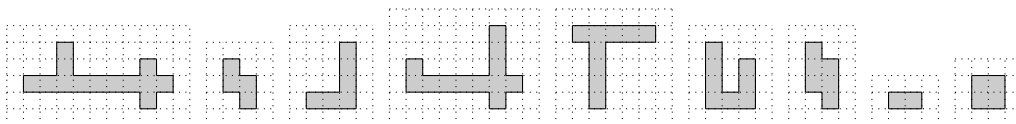
Номер подзадачи	Баллы	Ограничения		Комментарии
		$h, w$	$ c_i $	
1	32	$5 \leq h, w \leq 10$	$ c_i  \leq 10$	
2	22	$5 \leq h, w \leq 100$	$ c_i  \leq 100$	
3	23	$5 \leq h, w \leq 500$	$ c_i  \leq 500$	
4	до 23	$5 \leq h, w \leq 2000$	$ c_i  \leq 2000$	Тесты этой подзадачи оцениваются независимо

## Примеры

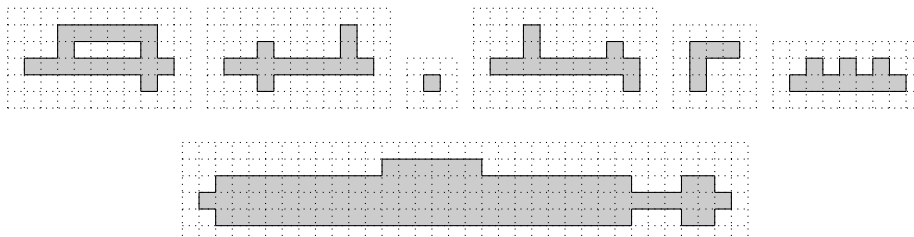
submarine.in	submarine.out	Изображение
<pre>2 -10 1 6 11 aaaaaaaaaaa aaabaaaaaaaa aaabaaaabaa abbbbbbbba aaaaaaaaabaa aaaaaaaaaaa</pre>	13	<pre>..... ...b..... ...b...b.. .bbbbbbbb. .....b.. .....</pre>
<pre>3 -4 -3 4 5 5 bbabc ссаас accba baccb baaaa</pre>	16	<pre>..... .c... .cc.. ..c.. .....</pre>
<pre>3 -2 4 0 5 5 abccb ссас cbcba cccbb accba</pre>	24	<pre>.b... .c... .b.b. cccbb ...b.</pre>
<pre>4 -1 -5 -3 0 5 5 bbabc ссаас acdba baccb baaaa</pre>	-2	<pre>..... ..aa. ..... ..... .....</pre>

## Пояснение

Для примера ниже приведены несколько потенциальных изображений подводной лодки.



Ниже приведены несколько множеств элементов снимка, которые не являются потенциальными изображениями подводной лодки:



## Задача 7. Фонари

Имя входного файла: `lamps.in`  
Имя выходного файла: `lamps.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 МБ

Улицу Подводный канал освещают  $n$  фонарей, пронумерованных вдоль улицы от 1 до  $n$ . Один или несколько подряд стоящих фонарей назовём сегментом. Таким образом, общее количество сегментов равно  $\frac{n(n+1)}{2}$ . Сегмент считается исправным, если лампочки во всех фонарях этого сегмента исправны.

С фонарями регулярно происходят события одного из двух типов:

- в каком-то сегменте из-за скачков напряжения все лампочки одновременно перегорают;
- Архиэнерго выбирает некоторый сегмент и посылает ремонтников, чтобы заменить на нем все перегоревшие лампочки на исправные.

После каждого события мэрия города требует от Архиэнерго предоставить отчёт о количестве исправных сегментов. Для улучшения показателей работы ремонтники включают в отчёт все сегменты, которые исправны сейчас или были исправны когда-либо ранее.

Требуется написать программу, определяющую количество сегментов после каждого события, которые исправны в этот момент или были исправны когда-либо до этого события.

### Формат входных данных

В первой строке входных данных содержатся два числа  $n$  и  $q$  — количество фонарей и количество произошедших событий. Следующая строка входных данных состоит из  $n$  символов 0 и 1, описывающих начальное состояние фонарей, где 1 обозначает фонарь с исправной лампочкой, а 0 — с перегоревшей.

В каждой из последующих  $q$  строк содержатся описания событий в виде трёх чисел  $l_i, r_i$  и  $c_i$ , которые означают, что после этого события все лампочки в фонарях с номерами  $l_i, l_i + 1, \dots, r_i$ :

- перегорают при  $c_i = 0$ ,
- становятся исправными при  $c_i = 1$ .

В описаниях всех событий  $1 \leq l_i \leq r_i \leq n$ , а  $c_i$  принимает значение 0 или 1.

### Формат выходных данных

В первой строке выходных данных выведите единственное число — количество исправных сегментов в начальном состоянии. Затем по одному в строке выведите  $q$  чисел: для каждого из произошедших событий выведите количество сегментов, указываемых в отчёте после этого события.

### Таблица системы оценивания

Номер подзадачи	Баллы	Ограничения		Комментарии
		$n$	$q$	
1	17	$1 \leq n \leq 50$	$1 \leq q \leq 150$	
2	19	$1 \leq n \leq 500$	$1 \leq q \leq 250$	
3	до 12	$1 \leq n \leq 5000$	$1 \leq q \leq 1000$	Тесты этой подзадачи оцениваются независимо
4	до 20	$1 \leq n \leq 50\,000$	$1 \leq q \leq 1000$	Тесты этой подзадачи оцениваются независимо
5	до 32	$1 \leq n \leq 300\,000$	$1 \leq q \leq 300\,000$	Тесты этой подзадачи оцениваются независимо

### Примеры

<code>lamps.in</code>	<code>lamps.out</code>
7 4	5
1100101	13
4 6 1	13
3 6 0	19
3 4 1	28
5 7 1	

## Задача 8. Сигнализация

Имя входного файла:	alarm.in
Имя выходного файла:	alarm.out
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 МБ

Подземный бункер состоит из  $n$  комнат, соединённых  $n - 1$  коридорами. Каждый коридор соединяет две различные комнаты и имеет определённую длину. Бункер устроен таким образом, что из любой комнаты  $i$  можно пойти в любую другую комнату  $j$ . Заметим, что существует единственный такой путь, не проходящий по одному и тому же коридору дважды. Сумма длин коридоров, составляющих этот путь, называется расстоянием между комнатами  $i$  и  $j$  и обозначается  $\rho(i, j)$ .

Каждая комната бункера оборудована звуковой сигнализацией, состоящей из сирены и датчика звука, который её включает. Сирена, включённая в комнате  $i$ , активирует датчик звука в каждой комнате, расстояние до которой не превосходит расстояние  $d_i$ , определяемое мощностью этой сирены. Другими словами, включение сирены в комнате  $i$  автоматически включает сирену во всех комнатах  $j$ , таких что  $\rho(i, j) \leq d_i$ . Эта сирена, в свою очередь, может вызвать автоматическое включение других сирен и так далее.

В случае возникновения чрезвычайной ситуации некоторые сирены необходимо включить вручную, после чего звук от них автоматически включит сирены в других комнатах. Правила безопасности предписывают выбор такого набора сирен для ручного включения, который в конце концов приведёт к автоматическому включению сирен во всех комнатах.

Требуется написать программу, которая определяет минимальное количество сирен в наборе, удовлетворяющем правилам безопасности.

### Формат входных данных

Первая строка входных данных содержит единственное число  $n$  — количество комнат.

Вторая строка содержит последовательность из  $n$  целых чисел  $d_i$ ,  $i$ -е из них равно максимальному расстоянию, на котором расположенная в комнате  $i$  сирена активирует датчики ( $0 \leq d_i \leq 10^9$ ).

Последующие  $n - 1$  строк описывают коридоры бункера. В  $i$ -й из них находятся три целых числа:  $u_i, v_i, l_i$ , где  $u_i, v_i$  — номера различных комнат, соединённых коридором  $i$ , а  $l_i$  — длина этого коридора ( $1 \leq u_i, v_i \leq n$ ;  $1 \leq l_i \leq 10^9$ ).

### Формат выходных данных

Выходные данные должны состоять из единственного числа — минимального количества сирен, которые необходимо включить вручную.

### Таблица системы оценивания

Номер подзадачи	Баллы	Ограничения
		$n$
1	16	$1 \leq n \leq 15$
2	23	$1 \leq n \leq 100$
3	17	$1 \leq n \leq 3000$
4	24	$1 \leq n \leq 100\,000$
5	20	$1 \leq n \leq 300\,000$

## Примеры

alarm.in	alarm.out
10	3
1 2 2 2 6 3 4 5 4 3	
1 2 5	
2 3 1	
2 4 5	
4 5 2	
4 6 4	
4 7 3	
1 8 1	
8 9 5	
8 10 4	

## Замечания

В тесте из примера сирена в комнате 4 включает сирену в комнате 5, которая, в свою очередь, включает сирены в комнатах 6 и 7. Сирена в комнате 2 включает сирену в комнате 3. Сирена в комнате 8 включает сирены в комнатах 1, 9 и 10.

