

## Задача 1. Выбор зала

Имя входного файла: hall.in  
Имя выходного файла: hall.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Для проведения церемонии открытия олимпиады по информатике организаторы осуществляют поиск подходящего зала. Зал должен иметь форму прямоугольника, длина каждой из сторон которого является целым положительным числом.

Чтобы все участники церемонии поместились в зале, и при этом он не выглядел слишком пустым, площадь зала должна находиться в пределах от  $A$  до  $B$  квадратных метров, включительно.

Чтобы разместить на стенах зала плакаты, рассказывающие об успехах школьников на олимпиадах, но при этом не создать ощущения, что успехов слишком мало, периметр зала должен находиться в пределах от  $C$  до  $D$  метров, включительно.

Прежде чем сделать окончательный выбор, организаторы олимпиады решили просмотреть по одному залу каждого подходящего размера. Залы с размерами  $Y \times Z$  и  $Z \times Y$  считаются одинаковыми. Чтобы понять необходимый объем работ по просмотру залов организаторы задались вопросом, сколько различных залов удовлетворяют приведенным выше ограничениям.

Требуется написать программу, которая по заданным  $A$ ,  $B$ ,  $C$  и  $D$  определяет количество различных залов, площадь которых находится в пределах от  $A$  до  $B$ , а периметр — от  $C$  до  $D$ , включительно.

### Формат входного файла

Входной файл содержит четыре разделенных пробелами целых числа:  $A$ ,  $B$ ,  $C$  и  $D$  ( $1 \leq A \leq B \leq 10^9$ ,  $4 \leq C \leq D \leq 10^9$ ).

### Формат выходного файла

Выходной файл должен содержать одно число — искомое количество залов.

### Пример входных и выходных файлов

hall.in	hall.out
2 10 4 8	3

### Пояснения к примеру

В примере ограничениям удовлетворяют залы следующих размеров:  $1 \times 2$ ,  $1 \times 3$ ,  $2 \times 2$ .

### Система оценки и описание подзадач

#### Подзадача 1 (50 баллов)

$1 \leq A \leq B \leq 1000$ ,  $4 \leq C \leq D \leq 1000$ .

Баллы за подзадачу начисляются только в случае, если все тесты успешно пройдены.

#### Подзадача 2 (50 баллов)

$1 \leq A \leq B \leq 10^9$ ,  $4 \leq C \leq D \leq 10^9$ .

В этой подзадаче 25 тестов, каждый тест оценивается в 2 балла. Баллы за каждый тест начисляются независимо.

### Получение информации о результатах окончательной проверки

По запросу сообщается результат окончательной проверки на каждом тесте.

## Задача 2. Призы

Имя входного файла: prizes.in  
Имя выходного файла: prizes.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Алиса и Боб стали победителями телевикторины, и теперь им предстоит выбрать себе призы. На выбор предлагается  $n$  призов, пронумерованных от 1 до  $n$ .

Распределение призов происходит следующим образом. Организаторы телевикторины сообщают победителям целое положительное число  $k$  ( $1 \leq k \leq n/3$ ). Сначала Алиса выбирает себе любые  $k$  подряд идущих номеров призов. Потом Боб выбирает себе  $k$  подряд идущих номеров призов, при этом он не может выбирать номера, которые уже выбрала Алиса. После этого победители забирают выбранные ими призы.

Алиса хорошо знает Боба, и для каждого приза выяснила его ценность для Боба, которая является целым положительным числом. Алиса обижена на Боба и хочет выбрать свои призы так, чтобы суммарная ценность призов, которые достанутся Бобу, была как можно меньше. При этом Алису не волнует, какие призы достанутся ей.

Требуется написать программу, которая по информации о ценности призов и значению  $k$  определит, для какого минимального значения  $x$  Алиса сможет добиться того, чтобы Боб не смог выбрать призы с суммарной ценностью больше  $x$ .

### Формат входного файла

Первая строка входного файла содержит два целых числа:  $n$  — общее количество призов и  $k$  — количество подряд идущих номеров призов, которое должен выбрать каждый из победителей ( $3 \leq n \leq 100\,000$ ,  $1 \leq k \leq n/3$ ).

Вторая строка содержит  $n$  целых положительных чисел:  $a_1, a_2, \dots, a_n$ . Для каждого приза указана его ценность для Боба ( $1 \leq a_i \leq 10^9$ ).

### Формат выходного файла

Выходной файл должен содержать одно число — минимальное значение  $x$ , для которого Алиса сможет добиться того, чтобы Боб не смог выбрать призы с суммарной ценностью больше  $x$ .

### Пример входных и выходных файлов

prizes.in	prizes.out
10 2 1 2 4 5 2 4 2 2 1 6	7

### Пояснение к примеру

В приведенном примере Алиса может, например, выбрать 4-й и 5-й призы. После этого для Боба оптимально выбрать 9-й и 10-й призы с суммарной ценностью 7.

### **Система оценки и описание подзадач**

В этой задаче три подзадачи. Баллы за подзадачу начисляются только в случае, если все тесты для данной подзадачи успешно пройдены.

#### **Подзадача 1 (30 баллов)**

$$3 \leq n \leq 50, 1 \leq a_i \leq 10^5$$

#### **Подзадача 2 (30 баллов)**

$$3 \leq n \leq 5000, 1 \leq a_i \leq 10^5$$

#### **Подзадача 3 (40 баллов)**

$$3 \leq n \leq 100\,000, 1 \leq a_i \leq 10^9$$

### **Получение информации о результатах окончательной проверки**

По запросу сообщается результат окончательной проверки на каждом тесте.

### Задача 3. Река

Имя входного файла: river.in  
Имя выходного файла: river.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Во Флатландии протекает богатая рыбой река Большой Флат. Много лет назад река была поделена между  $n$  рыболовными предприятиями, каждое из которых получило непрерывный отрезок реки. При этом  $i$ -е предприятие, если рассматривать их по порядку, начиная от истока, изначально получило отрезок реки длиной  $a_i$ .

С тех пор с рыболовными предприятиями во Флатландии  $k$  раз происходили различные события. Каждое из событий было одного из двух типов: банкротство некоторого предприятия или разделение некоторого предприятия на два.

При некоторых событиях отрезок реки, принадлежащий предприятию, с которым это событие происходит, делится на две части. Каждый такой отрезок имеет длину большую или равную 2. Деление происходит по следующему правилу. Если отрезок имеет четную длину, то он делится на две равные части. Иначе он делится на две части, длины которых различаются ровно на единицу, при этом часть, которая ближе к истоку реки, имеет меньшую длину.

При банкротстве предприятия происходит следующее. Отрезок реки, принадлежавший обанкротившемуся предприятию, переходит к его соседям. Если у обанкротившегося предприятия один сосед, то этому соседу целиком передается отрезок реки обанкротившегося предприятия. Если же соседей двое, то отрезок реки делится на две части описанным выше способом, после чего каждый из соседей присоединяет к своему отрезку ближайшую к нему часть.

При разделении предприятия отрезок реки, принадлежавший разделяемому предприятию, всегда делится на две части описанным выше способом. Разделившееся предприятие ликвидируется, и образуются два новых предприятия.

Таким образом, после каждого события каждое предприятие владеет некоторым отрезком реки.

Министерство финансов Флатландии предлагает ввести налог на рыболовные предприятия, пропорциональный квадрату длины отрезка реки, принадлежащего соответствующему предприятию. Чтобы проанализировать, как будет работать этот налог, министр хочет по имеющимся данным узнать, как изменялась величина, равная сумме квадратов длин отрезков реки, принадлежащих предприятиям, после каждого произошедшего события.

Требуется написать программу, которая по заданному начальному разделению реки между предприятиями и списку событий, происходивших с предприятиями, определит, чему равна сумма квадратов длин отрезков реки, принадлежащих предприятиям, в начальный момент времени и после каждого события.

#### Формат входного файла

Первая строка входного файла содержит два целых числа:  $n$  и  $p$  — исходное количество предприятий ( $2 \leq n \leq 100\,000$ ) и номер подзадачи ( $0 \leq p \leq 4$ ).

Вторая строка входного файла содержит  $n$  целых чисел  $a_1, a_2, \dots, a_n$  — длины исходных отрезков реки.

Третья строка входного файла содержит целое число  $k$  — количество событий, происходивших с предприятиями ( $1 \leq k \leq 100\,000$ ).

Последующие  $k$  строк содержат описания событий,  $i$ -я строка содержит два целых числа:  $e_i$  и  $v_i$  — тип события и номер предприятия, с которым оно произошло. Значение  $e_i = 1$  означает, что предприятие, которое после всех предыдущих событий является  $v_i$ -м по

порядку, если считать с единицы от истока реки, обанкротилось, а значение  $e_i = 2$  означает, что это предприятие разделилось на два.

Гарантируется, что значение  $v_i$  не превышает текущее количество предприятий. Гарантируется, что если отрезок предприятия при банкротстве или разделении требуется поделить на две части, то он имеет длину большую или равную 2. Гарантируется, что если на реке осталось единственное предприятие, оно не банкротится.

### Формат выходного файла

Выходной файл должен содержать  $(k + 1)$  целых чисел, по одному в строке. Первая строка должна содержать исходную сумму квадратов длин отрезков реки, а каждая из последующих  $k$  строк — сумму квадратов длин отрезков реки после очередного события.

### Пример входных и выходных файлов

river.in	river.out
4 0	75
3 5 5 4	105
5	73
1 1	101
2 1	83
1 3	113
2 2	
1 3	

### Система оценки и описание подзадач

В этой задаче четыре подзадачи. Баллы за каждую подзадачу начисляются только в случае, если все тесты для данной подзадачи успешно пройдены.

**Внимание!** Тест из примера не подходит под ограничение подзадачи 3. Тем не менее, решение задачи принимается на проверку только в том случае, если оно выводит правильный ответ на тесте из примера, даже если участник претендует на правильное решение только этой подзадачи.

В первой строке каждого теста после числа  $n$  указан номер подзадачи, для теста из примера указано число 0, в тестах первой подзадачи указано число 1, и т. д.

#### Подзадача 1 (30 баллов)

$$2 \leq n \leq 100, 1 \leq k \leq 100, 1 \leq a_i \leq 100, p = 1$$

#### Подзадача 2 (30 баллов)

$$2 \leq n \leq 100\,000, 1 \leq k \leq 100\,000, 1 \leq a_i \leq 10^4, p = 2$$

Для всех  $i$  от 1 до  $k - 1$  выполнено условие:  $|v_i - v_{i+1}| \leq 10$

#### Подзадача 3 (20 баллов)

$$2 \leq n \leq 100\,000, 1 \leq k \leq 100\,000, 1 \leq a_i \leq 10^4, p = 3$$

Все события имеют тип  $e_i = 1$  (нет разделений предприятий, только банкротства).

#### Подзадача 4 (20 баллов)

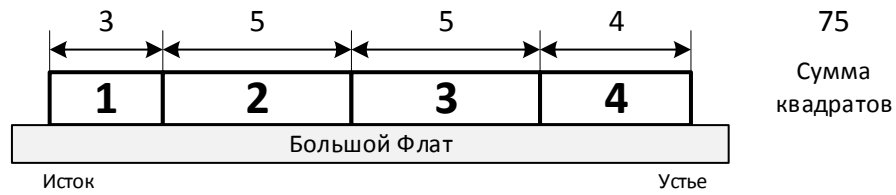
$$2 \leq n \leq 100\,000, 1 \leq k \leq 100\,000, 1 \leq a_i \leq 10^4, p = 4$$

### Получение информации о результатах окончательной проверки

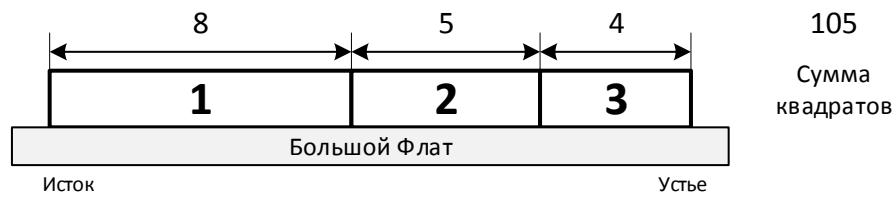
По запросу сообщается результат окончательной проверки на каждом тесте.

### Пояснение к примеру

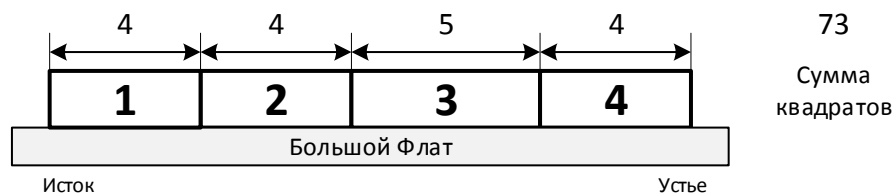
Распределение отрезков реки между предприятиями после каждого события, описанного в примере, приведено на рисунке ниже.



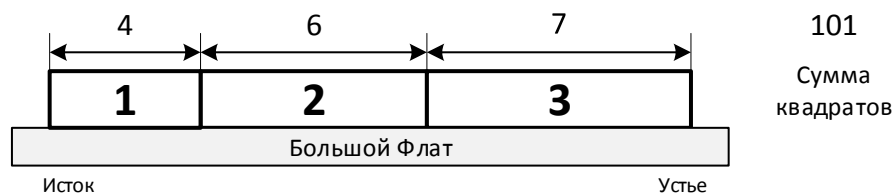
*После банкротства предприятия 1*



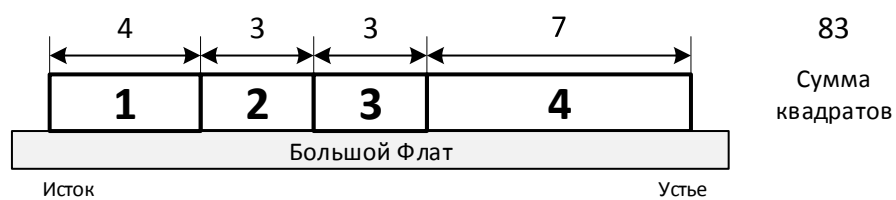
*После разделения предприятия 1*



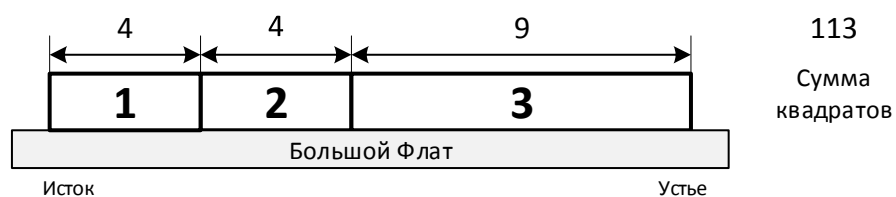
*После банкротства предприятия 3*



*После разделения предприятия 2*



*После банкротства предприятия 3*



## Задача 4. Чемпионат по поиску в сети Меганет

Имя входного файла: search.in  
Имя выходного файла: search.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Для проведения чемпионата мира по поиску в сети Меганет организаторам необходимо ограничить доступ к некоторым адресам. Адрес в сети Меганет представляет собой строку, состоящую из имени сервера и имени раздела.

*Имя сервера* представляет собой строку, содержащую от одной до пяти частей включительно. Каждая часть представляет собой непустую строку, состоящую из строчных букв латинского алфавита. Части разделены точкой. Примеры корректных имен сервера: «a», «ab.cd», «abacaba», «a.b.c.d.e».

*Имя раздела* представляет собой строку, которая может быть либо пустой, либо содержать от одной до пяти частей включительно. Каждая часть начинается с символа «/», после которого следует одна или несколько строчных латинских букв. Примеры корректных имен разделов: «», «/a», «/aba», «/a/b/c/d/e».

*Адрес* формируется приписыванием имени раздела в конец имени сервера. Например, корректными адресами являются строки: «a», «aba/d/f/g/h», «a.b», «aba.caba/def/g», «c.d.e.f.g/a/b/c/d/e».

Для ограничения доступа к некоторым адресам сети Меганет организаторы чемпионата подготовили несколько фильтров. Фильтр, как и адрес, состоит из двух частей: фильтра сервера и фильтра раздела.

*Фильтр сервера* состоит из имени сервера, перед которым может также идти строка «\*». Если фильтр сервера представляет собой только имя сервера, то этому фильтру соответствует только сервер, имеющий точно такое же имя. Если фильтр сервера представляет собой строку «\*.S», где *S* — имя сервера, то ему соответствуют сервера, удалением нуля или более начальных частей от имени которых можно получить строку *S*.

Аналогично, *фильтр раздела* представляет собой имя раздела, после которого может идти строка «/\*». Фильтру раздела, который представляет собой просто имя раздела *R*, соответствуют только разделы, в точности совпадающие с *R*. Если фильтр раздела представляет собой строку «R/\*», то ему соответствуют все разделы, удалением от имен которых нуля или более конечных частей можно получить строку *R*.

Адрес *соответствует* фильтру, если его имя сервера соответствует фильтру сервера, а его имя раздела соответствует фильтру раздела.

Примеры фильтров и соответствующих им адресов приведены в таблице ниже.

Фильтр	Примеры подходящих адресов		
ab.c/d/e	ab.c/d/e		
*.a	a	ax.a	efg.a
*.a/b/c	a/b/c	x.a/b/c	e.fg.a/b/c
x.yz/a/*	x.yz/a	x.yz/a/b/c	x.yz/a/xyz
*.a/*	a	x.a	e.fg.a
	a/b/c	x.a/ddd/c	e.fg.a/b/c/g/haha/i
*.a/b/c/*	a/b/c	x.a/b/c	e.fg.a/b/c
	a/b/c/xxx	e.fg.a/b/c/d/e/f	

Требуется написать программу, которая по заданному набору фильтров и списку адресов определяет для каждого адреса, какому числу фильтров соответствует этот адрес.

### Формат входного файла

Первая строка входного файла содержит два целых числа:  $n$  — количество фильтров ( $1 \leq n \leq 50\,000$ ) и  $p$  — номер подзадачи ( $0 \leq p \leq 3$ ).

Последующие  $n$  строк содержат фильтры, по одному на строке. Каждый фильтр удовлетворяет ограничениям, описанным выше.

Следующая строка содержит одно целое число  $k$  — количество адресов, которые необходимо проверить на соответствие фильтрам ( $1 \leq k \leq 50\,000$ ).

Последующие  $k$  строк содержат адреса, по одному на строке. Каждый адрес удовлетворяет ограничениям, описанным выше.

Длина каждой строки входного файла не превышает 50 символов.

Общий размер входного файла не превышает 4 мегабайт.

### Формат выходного файла

Выходной файл должен содержать  $k$  целых чисел, по одному на строке — для каждого адреса необходимо вывести количество фильтров, которым соответствует этот адрес.

### Примеры входных и выходных файлов

search.in	search.out
2 0	0
a.bb/c	1
bb/c/d	0
4	0
a.bb	
bb/c/d	
a.bb/c/d	
bb/c	
4 0	0
*.bb/c	4
*.bb/c/*	3
bb/c/*	0
bb/c/*	2
6	1
bb	
bb/c	
bb/c/d	
a.bb	
a.bb/c	
a.bb/c/d	

### Пояснения к примерам

В первом примере в фильтрах не встречается символ «\*», поэтому адрес соответствует фильтру только в случае полного совпадения.

Во втором примере следует обратить внимание на то, что фильтры могут повторяться, а также, что фильтрам вида «\*.<сервер>/...» соответствуют также адреса, в которых часть имени сервера полностью совпадает с соответствующей частью фильтра. Аналогично фильтрам «.../<раздел>/\*» соответствуют также адреса, в которых часть имени раздела полностью совпадает с соответствующей частью фильтра.

**Внимание!** Первый тест из примера не подходит под ограничение подзадачи 1, а второй тест из примера не подходит под ограничения подзадач 1 и 2. Тем не менее, решение принимается на проверку только в том случае, если оно выводит правильный ответ на оба теста из примера, даже если участник претендует на правильное решение только одной из указанных подзадач.



### **Система оценки и описание подзадач**

В этой задаче три подзадачи. В первой строке каждого теста после числа  $n$  указан номер подзадачи, для теста из примера указано число 0, в тестах первой подзадачи указано число 1, и т. д.

#### **Подзадача 1 (27 баллов)**

$$1 \leq n \leq 1000, 1 \leq k \leq 1000, p = 1$$

Каждый фильтр начинается с «\* .» и заканчивается на «/\*».

В этой подзадаче 9 тестов, каждый тест оценивается в 3 балла. Баллы за каждый тест начисляются независимо.

#### **Подзадача 2 (25 баллов)**

$$1 \leq n \leq 50\,000, 1 \leq k \leq 50\,000, p = 2$$

Фильтры не содержат символов «\*».

В этой подзадаче 5 тестов, каждый тест оценивается в 5 баллов. Баллы за каждый тест начисляются независимо.

#### **Подзадача 3 (48 баллов)**

$$1 \leq n \leq 50\,000, 1 \leq k \leq 50\,000, p = 3$$

В этой подзадаче 12 тестов, каждый тест оценивается в 4 балла. Баллы за каждый тест начисляются независимо.

### **Получение информации о результатах окончательной проверки**

По запросу сообщаются баллы за каждую подзадачу.