Задача 1. Выбор зала

Имя входного файла: ha11. in

Имя выходного файла: ha11. 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 *А* до *В* 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 *С* до *D* метров, включительно.

Прежде чем сделать окончательный выбор, организаторы олимпиады решили просмотреть по одному залу каждого подходящего размера. Залы с размерами *У Z* и *Z У* 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 *А, В, С* и *D* определяет количество различных залов, площадь которых находится в пределах от *А* до *В, а* периметр — от *С* go *D,* включительно.

Формат вхоdлосо *файла*

Входной файл содержит четыре разделенных пробелами целых числа: *А, В, С* и *D*

(1 Л q< *В* q< 109, 4 *С q< D \_<* 10 ).

#### *Формат выходного файла*

Выходной файл должен содержать одно число — искомое количество залов.

*Пример входных и выходных Qайлов*

|  |  |
| --- | --- |
| **hall.in** | **hall.out** |
| 2 10 4 8 |  |

## Поясиеиия к примеру

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

### *Система оценки и описание подзадач*

*Подзадача 1 (50 баллов)*

##### 1 < Л < *В* < 1000, 4 < *С* < *D* < 1000.

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

*Подзадача 2 (50 баллов)*

1 < Л < *В* < 109, 4 < *С* < *D* < 10 .

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

Лолучелие ьлформацьь о *результатах окончательной проверки*

По запросу сообщается результат окончательной проверки на каждом тесте.

# Задача 2. Призы

Имя входного файла: pr i z еs . in

Имя выходного файла: pr i zеs . 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 п призов, пронумерованных от 1 до п.

Распределение призов происходит следующим образом. Организаторы телевикторины сообщают победителям целое положительное число /г (1 *k \_< n /* 3). Сначала Алиса выбирает себе любые k подряд идущих номеров призов. Потом Боб выбирает себе I 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 х Алиса сможет добиться того, чтобы Боб не смог выбрать призы с суммарной ценностью больше х.

Формат вхоdлосо *файла*

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

Вторая строка содержит в целых положительных чисел: 2. ,. Для К£tЖДОГО

приза указана его ценность для Боба (1 *<;* 10 ).

#### *Формат выходного файла*

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

*Пример входных и выходных файлов*

|  |  |
| --- | --- |
| **prizes.in** | **prizes.out** |
| 10 21 2 4 5 2 4 2 2 1 6 | 7 |

/7ояслелье х л£\*•меру

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 п 50, 1 п, 105

Подзадача *2 (30 баллов)*

3 < п < 5000, 1 < *а,<* 105

###### *Подзадача 3 (40 баллов)*

3 п 100 000, 1 *п;* 109

### *Получение информации о результатах окончательной проверки*

По запросу сообщается результат окончательной проверки на каждом тесте.

Задача 3. Река

ИмявходногофаЗла: river.in

Имя выходного файла: гiver . out Ограничение по времени: 1 секунда Ограничение по памяти: 256 мегабайт

Во Флатландии протекает богатая рыбои река Большои Флат. Много лет назад река была поделена между п рыболовными предприятиями, каждое из которых получило непрерывныи отрезок реки. При этом i-e предприятие, если рассматривать их по порядку, начиная от истока, изначально получило отрезок реки длинои п,.

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

При некоторых событиях отрезок реки, принадлежащии предприятию, с которым это событие происходит, делится на две части. Каждыи такои 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, принадлежавшии обанкротившемуся предприятию, переходит к его соседям. Если у обанкротившегося предприятия один сосед, то этому соседу целиком передается отрезок реки обанкротившегося предприятия. Если же соседеи двое, то отрезок реки делится на две части описанным выше способом, после чего каждыи из соседеи присоединяет к своему отрезку ближаишую к нему часть.

При разделении предприятия отрезок реки, принадлежавшии 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, пропорциональныи 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 событии, происходивших с предприятиями, определит, чему равна сумма квадратов длин отрезков реки, принадлежащих предприятиям, в начальныи момент времени и после каждого события.

Формат *входного файла*

Первая строка входного файла содержит два целых числа: u и *р* — исходное количество предприятий (2 < п < 100 000) и номер подзадачи (0 *< р <* 4).

Вторая строка входного файла содержит п целых чисел *а , az, . . .,* nn — длины исходных отрезков реки.

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

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



порядку, если считать с единицы от истока реки, обанкротилось, а значение *е, ——* 2 означает, что это предприятие разделилось на два.

Гарантируется, что значение v, 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 выхоdлоао *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 /г 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 12 1 | 10183 |
| 1 3 | 113 |
| 2 2 |  |
| 1 3 |  |

*Система* оцелхь ь *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 п указан номер подзадачи, для теста из примера указано число 0, в тестах первой подзадачи указано число 1, и т. д.

Пodзadaчa I *(30 баллов)*

2 u 100, 1 k 100, 1 *а, q<* 100, *р* —— 1

*Подзадача 2 (30 баллов)*

2 < п < 100 000, 1 < k < 100 000, 1 < *а; <* 104, *р* ——2

Для всех *i* от 1 до /г — 1 выполнено условие: v, — v, + Ј< 10

###### *Подзадача 3 (20 баллов)*

2 < п < 100 000, 1 < k < 100 000, 1 < *а; <* 104, *р* ——3

Все события имеют тип *ee ——* 1 (нет разделений предприятий, только банкротства).

*Подзадача 4 (20 баллов)*

2 < п < 100 000, 1 < k < 100 000, 1 < *а; <* 104, *р* —— 4

Лолучелие ьлформацьь о *результатах окончательной проверки*

По запросу сообщается результат окончательной проверки на каждом тесте.

Поясиеиие к примеру

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

75

Сумма квадра тов

*После банкротства предприятия 1*

Устье

##### 8 S 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| | 1 | | | 2 | | | 3 |

Бол ьшой Флат



*После разделения предприятия 1*

*После банкротства предприятия 3*

*После разделения предприятия 2*

Бол ьшой Флат



*После банкроства предприятия 3*



|

Устье

Устье

Устье

Устье

Устье

больш ой Фла т

105

Сумма квадра тов

73

Сумма квадра тов

101

Сумма квадра тов

83

Сумма квадра тов

113

Сумма квадра тов

**Задача 4. Чемпионат по поиску в сети Меганет**

Имя входного файла: sear ci . in

Имя выходного файла: sear ch . 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: «а»,

«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: «», «/ а», «/ aba», «/а/Ь /с/d/е».

*Адрес* формируется приписыванием имени раздела в конец имени сервера. Например, корректными адресами являются строки: «а», «aba/d/ I /q/h», «а . b», «aba . eaba /def /q»,

«с . d. е . f . q/а/Ь/с/d/е».

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 fi.

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

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 |
| \*.а | а | ах.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/\* | аa/b/c | х.аx.a/ddd/c | e.fg.ae.fg.a/b/c/g/haha/i |
| \*.a/b/c/\* | a/b/c a/b/c/xxx | x.a/b/c e.fg.a/b/c 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 вхоdлосо *файла*

Первая строка входного файла содержит два целых числа: п — количество фильтров (1 < п < 50 000) и *р* — номер подзадачи (0 *< р <* 3).

Последующие п 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 < *k <* 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 выхоdлоао *файла*

Выходной файл должен содержать /г 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 0a.bb/c bb/3/d 4a.bb bb/3/d a.bb/c/d bb/c | 000 |
| 4 0\*.bb/c\*.bb/c/\*bb/c/\* | 0430 |
| bb/c/\* | 2 |
| 6 |  |
| bb |  |
| Ьь/с |  |
| bb/c/d |  |
| a.bb |  |
| a.bb/c |  |
| a.bb/c/d |  |

/7ояслелья If *ПR<мера м*

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 «.. . / <paздeл> / \*» 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* оцелхь ь *описание подзадач*

В этой задаче три подзадачи. В первой строке каждого теста после числа в 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 < п < 1000, 1 < k < 1000, *р* —— 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 < п < 50 000, 1 < /г < 50 000, *р* —— 2

Фильтры не содержат символов «\*».

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

Пodзadaчa 3 *(48* баллов)

1 < п < 50 000, 1 < /г < 50 000, *р* —— 3

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

НІІЧИСЛЯЮТСЯ НeЗIIBИCИMO.

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

По запросу сообщаются баллы за каждую подзадачу.