

## Задача 5. Кольцевая линия

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

В городе, в котором живут друзья Андрей и Борис, метро состоит из единственной кольцевой линии, вдоль которой на равном расстоянии друг от друга расположены  $n$  станций, пронумерованных от 1 до  $n$ . Участок линии метро между двумя соседними станциями называется перегонном.

Поезда по кольцевой линии двигаются как по часовой стрелке, так и против часовой стрелки, поэтому чтобы добраться от одной станции до другой, пассажир может выбрать то направление, в котором требуется проехать меньше перегонов. Минимальное число перегонов, которое необходимо проехать, чтобы добраться от одной станции до другой, назовем расстоянием между станциями.

Друзья заметили, что выполняется следующее условие: если загадать некоторую станцию  $X$  и выписать для нее два числа:  $D_a$  — расстояние от станции, на которой живет Андрей, до станции  $X$  и  $D_b$  — расстояние от станции, на которой живет Борис, до станции  $X$ , то полученная пара чисел  $[D_a, D_b]$  будет однозначно задавать станцию  $X$ .

Например, если  $n = 4$ , Андрей живет на станции 1, а Борис живет на станции 2, то станция 1 задается парой  $[0, 1]$ , станция 2 — парой  $[1, 0]$ , станция 3 — парой  $[2, 1]$  и станция 4 — парой  $[1, 2]$ .

Их одноклассник Сергей живет в соседнем городе и не знает, на каких станциях живут Андрей и Борис. Чтобы найти друзей, он заинтересовался, сколько существует вариантов пар станций  $A, B$ , таких что если Андрей живет на станции  $A$ , а Борис — на станции  $B$ , то выполняется описанное выше условие.

Требуется написать программу, которая по числу станций  $n$  на кольцевой линии определяет искомое количество вариантов.

### Формат входного файла

Первая строка входного файла содержит одно целое число  $n$  ( $3 \leq n \leq 40\,000$ ).

### Формат выходного файла

Выходной файл должен содержать одно число — искомое количество вариантов.

### Примеры входных и выходных файлов

circle.in	circle.out
4	8
5	20

### Пояснения к примерам

В первом примере подходят следующие варианты:

- Андрей живет на станции 1, а Борис на станции 2;
- Андрей живет на станции 1, а Борис на станции 4;
- Андрей живет на станции 2, а Борис на станции 1;
- Андрей живет на станции 2, а Борис на станции 3;
- Андрей живет на станции 3, а Борис на станции 2;
- Андрей живет на станции 3, а Борис на станции 4;
- Андрей живет на станции 4, а Борис на станции 1;
- Андрей живет на станции 4, а Борис на станции 3.

### **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 (25 баллов)**

$$3 \leq n \leq 50.$$

#### **Подзадача 2 (25 баллов)**

$$3 \leq n \leq 500.$$

#### **Подзадача 3 (50 баллов)**

$$3 \leq n \leq 40000.$$

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

По запросу сообщается результат окончательной проверки на каждом тесте.

## Задача 6. Вырубка леса

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

Фермер Николай нанял двух лесорубов: Дмитрия и Федора, чтобы вырубить лес, на месте которого должно быть кукурузное поле. В лесу растут  $X$  деревьев.

Дмитрий срубает по  $A$  деревьев в день, но каждый  $K$ -й день он отдыхает и не срубает ни одного дерева. Таким образом, Дмитрий отдыхает в  $K$ -й,  $2K$ -й,  $3K$ -й день, и т.д.

Федор срубает по  $B$  деревьев в день, но каждый  $M$ -й день он отдыхает и не срубает ни одного дерева. Таким образом, Федор отдыхает в  $M$ -й,  $2M$ -й,  $3M$ -й день, и т.д.

Лесорубы работают параллельно и, таким образом, в дни, когда никто из них не отдыхает, они срубают  $A + B$  деревьев, в дни, когда отдыхает только Федор —  $A$  деревьев, а в дни, когда отдыхает только Дмитрий —  $B$  деревьев. В дни, когда оба лесоруба отдыхают, ни одно дерево не срубается.

Фермер Николай хочет понять, за сколько дней лесорубы срубят все деревья, и он сможет засеять кукурузное поле.

Требуется написать программу, которая по заданным целым числам  $A$ ,  $K$ ,  $B$ ,  $M$  и  $X$  определяет, за сколько дней все деревья в лесу будут вырублены.

### Формат входного файла

Входной файл содержит пять целых чисел, разделенных пробелами:  $A$ ,  $K$ ,  $B$ ,  $M$  и  $X$  ( $1 \leq A, B \leq 10^9$ ,  $2 \leq K, M \leq 10^{18}$ ,  $1 \leq X \leq 10^{18}$ ).

### Формат выходного файла

Выходной файл должен содержать одно целое число — искомое количество дней.

### Пример входных и выходных файлов

forest.in	forest.out
2 4 3 3 25	7

### Пояснение к примеру

В приведенном примере лесорубы вырубят 25 деревьев за 7 дней следующим образом:

- 1-й день: Дмитрий срубает 2 дерева, Федор срубает 3 дерева, итого 5 деревьев;
- 2-й день: Дмитрий срубает 2 дерева, Федор срубает 3 дерева, итого 10 деревьев;
- 3-й день: Дмитрий срубает 2 дерева, Федор отдыхает, итого 12 деревьев;
- 4-й день: Дмитрий отдыхает, Федор срубает 3 дерева, итого 15 деревьев;
- 5-й день: Дмитрий срубает 2 дерева, Федор срубает 3 дерева, итого 20 деревьев;
- 6-й день: Дмитрий срубает 2 дерева, Федор отдыхает, итого 22 дерева;
- 7-й день: Дмитрий срубает 2 дерева, Федор срубает оставшееся 1 дерево, итого все 25 деревьев срублены.

**Внимание!** Тест из примера не подходит под ограничения для подзадач 2 и 3, но решение принимается на проверку только в том случае, если оно выводит правильный ответ на тесте из примера. Решение должно выводить правильный ответ на тест даже, если оно рассчитано на решение только каких-либо из подзадач 2 и 3.

## **Система оценки и описание подзадач**

### **Подзадача 1 (32 балла)**

$$1 \leq X \leq 1000, 1 \leq A, B \leq 1000, 2 \leq K, M \leq 1000$$

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

### **Подзадача 2 (10 баллов)**

$$1 \leq X \leq 10^{18}$$

$$X < K$$

$$X < M$$

При решении этой подзадачи можно считать, что лесорубы не отдыхают.

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

### **Подзадача 3 (10 баллов)**

$$1 \leq X \leq 10^{18}$$

Дополнительно к приведенным ограничениям выполняется условие  $K = M$ .

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

### **Подзадача 4 (48 баллов)**

$$1 \leq X \leq 10^{18}, 1 \leq A, B \leq 10^9, 2 \leq K, M \leq 10^{18}$$

В этой подзадаче 16 тестов, каждый тест оценивается в 3 балла. Баллы за каждый тест начисляются независимо.

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

По запросу сообщается результат окончательной проверки на каждом тесте.

## Задача 7. Укладка плитки

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

В процессе ремонта в Лаборатории Информационных Технологий строителям необходимо заменить поврежденные напольные плитки в коридоре лаборатории, который имеет размер  $2 \times n$  метров. В распоряжении строителей есть неограниченный запас плиток двух размеров:  $1 \times 2$  метра и  $1 \times 1$  метр. При этом плитки размером  $1 \times 2$  метра перед укладкой разрешается поворачивать на 90 градусов и размещать как вдоль, так и поперек коридора.

Строители уже начали ремонт и уложили в некоторых местах пола коридора  $k$  плиток размером  $1 \times 1$ . Для завершения ремонта прорабу необходимо подготовить план дальнейших работ. Для этого ему надо решить, каким образом уложить плитки на места, где они еще не уложены. Это можно сделать различными способами и прораб хочет перебрать все варианты и выбрать самый удачный. Перед тем как это сделать, прораб хочет знать, какое количество вариантов ему придется рассмотреть. Это число требуется найти по модулю  $10^9 + 7$ .

Требуется написать программу, которая по заданной длине коридора  $n$  и расположению плиток, которые уже уложены, определяет количество способов укладки плиток на оставшиеся места. Ответ необходимо вывести по модулю  $10^9 + 7$ .

### Формат входного файла

Первая строка входного файла содержит два целых числа:  $n$  — длину коридора и  $k$  — количество уже уложенных единичных плиток ( $1 \leq n \leq 100\,000$ ,  $0 \leq k < 2n$ ).

Последующие  $k$  строк содержат по два целых числа  $x_i$  и  $y_i$ , которые задают позиции уже уложенных единичных плиток,  $i$ -я плитка уложена на  $x_i$ -м метре коридора в  $y_i$ -м ряду ( $1 \leq x_i \leq n$ ,  $1 \leq y_i \leq 2$ ).

### Формат выходного файла

Выходной файл должен содержать одно целое число — количество способов укладки плиток в коридоре, взятое по модулю  $10^9 + 7$ .

### Примеры входных и выходных файлов

<code>tiling.in</code>	<code>tiling.out</code>
2 0	7
3 0	22
3 1 2 1	8

### Пояснение к примерам

**Внимание!** Третий тест не подходит под ограничения для первых трех подзадач, но решение принимается на проверку только в том случае, если оно выводит правильный ответ на все тесты из примера. Решение должно выводить правильный ответ на третий тест даже, если оно рассчитано на решение только каких-либо подзадач из первых трех.



Рисунок 1. Все способы укладки плиток в первом примере

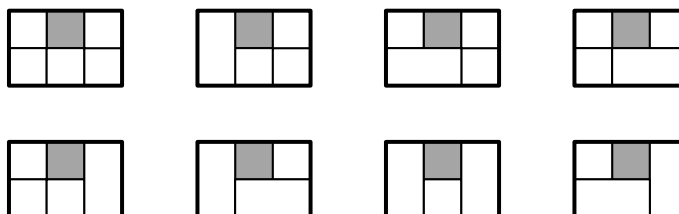


Рисунок 2. Все способы укладки плиток в третьем примере.  
Уже уложенная плитка отмечена серым цветом.

## **Система оценки и описание подзадач**

### **Подзадача 1 (20 баллов)**

$$1 \leq n \leq 8, k = 0$$

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

### **Подзадача 2 (20 баллов)**

$$1 \leq n \leq 1000, k = 0$$

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

### **Подзадача 3 (20 баллов)**

$$1 \leq n \leq 100\,000, k = 0$$

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

### **Подзадача 4 (40 баллов)**

$$1 \leq n \leq 100\,000, 1 \leq k \leq 2n$$

В этой подзадаче 20 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 8. Магические порталы

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

В некотором королевстве есть  $n$  городов, соединенных магическими порталами. Каждая пара различных городов соединена ровно одним магическим порталом, позволяющим мгновенно перемещаться из одного города в другой.

Из-за свойств магии, определяющей работу порталов, каждый портал можно использовать только в одну сторону. Для каждой пары городов  $A$  и  $B$  известно, можно ли воспользоваться порталом для перемещения напрямую из  $A$  в  $B$  или из  $B$  в  $A$ .

Из-за особенностей магических порталов иногда при перемещении жителей королевства из одного города в другой приходится использовать несколько порталов. Также могут существовать такие пары городов, что из одного города в другой нельзя добраться, используя только магические порталы.

Жители королевства называют город *совершенным*, если из него можно добраться до любого другого города в королевстве, используя только магические порталы. Пусть изначально количество совершенных городов в королевстве равно  $k$ .

Недавно король принял решение выбрать пару городов и изменить разрешенное направление перемещения по portalу, соединяющему их, на противоположное. Для выбора лучшего варианта король хочет понять, какое количество совершенных городов может оказаться в королевстве после перенастройки ровно одного портала.

Для получения этой информации король планирует запросить в министерстве транспорта соответствующий отчет. Король может запросить либо *частичный*, либо *полный* отчет. Содержимое отчета зависит от параметра  $L$ , для частичного отчета  $L = k + 1$ , для полного отчета  $L = 1$ .

Отчет содержит для каждого целого числа  $m$ , такого что  $m \geq L$ , число таких пар городов  $A$  и  $B$ , для которых выполняются следующие условия:

- исходно магический портал позволяет перемещаться напрямую из города  $A$  в город  $B$ ;
- если изменить направление перемещения этого магического портала на противоположное, чтобы он позволял напрямую перемещаться из города  $B$  в город  $A$ , то количество совершенных городов в королевстве станет равным  $m$ .

Таким образом, частичный отчет содержит информацию только о тех способах перенастройки одного портала, которые строго увеличивают количество совершенных городов в королевстве. Полный отчет содержит информацию обо всех способах перенастройки одного портала.

Требуется написать программу, которая по информации о разрешенных направлениях перемещения с использованием магических порталов, и информации о том, требуется предоставить частичный или полный отчет, формирует соответствующий отчет и выводит его в описанном ниже формате.

### Формат входного файла

Первая строка входного файла содержит два целых числа:  $n$  — количество городов в королевстве ( $2 \leq n \leq 2000$ ) и  $p$ , равное либо 0, если требуется вывести частичный отчет, либо 1, если требуется вывести полный отчет.

Последующие  $n$  строк содержат по  $n$  символов, каждый из которых может быть «+», «-» или «.», и  $i$ -я из этих строк описывает магические порталы, соединяющие  $i$ -й город с другими городами.

В  $i$ -й строке  $j$ -й символ равен «+», если магический портал позволяет напрямую перемещаться из  $i$ -го города в  $j$ -й, равен «-», если магический портал позволяет напрямую перемещаться из  $j$ -го города в  $i$ -й, и равен «.», если  $i = j$ .

### Формат выходного файла

Первая строка выходного файла должна содержать одно целое число  $k$  — количество совершенных городов в королевстве.

Если требуется частичный отчет ( $p = 0$ ), то вторая строка выходного файла должна содержать  $(n - k)$  целых неотрицательных чисел, разделенных пробелами, где  $i$ -е из этих чисел должно быть равно количеству пар городов, изменение направления портала между которыми на противоположное приводит к тому, что количество совершенных городов в королевстве станет равным  $(k + i)$ . Если при этом  $k = n$ , то вторая строка может отсутствовать, либо быть пустой.

Если требуется полный отчет ( $p = 1$ ), то вторая строка должна содержать  $n$  целых неотрицательных чисел, разделенных пробелами, где  $i$ -е из этих чисел должно быть равно количеству пар городов, изменение направления портала между которыми на противоположное приводит к тому, что количество совершенных городов в королевстве станет равным  $i$ .

### Примеры входных и выходных файлов

transform.in	transform.out
5 0 .-+++ + .+++ -- .+- --- .+ ---+-.	1 0 0 0 3
5 1 .-+++ + .+++ -- .+- --- .+ ---+-.	1 7 0 0 0 3

### Пояснение к примерам

В приведенных примерах изначально совершенным является только город 2.

Изменив направление порталов, соединяющих пары городов (2, 3), (2, 4) или (2, 5), можно сделать все города совершенными. Изменение направления любого другого портала делает совершенным один город.

**Внимание!** Второй тест не подходит под ограничения для первых трех подзадач, но решение принимается на проверку только в том случае, если оно выводит правильный ответ на все тесты из примера. Решение должно выводить правильный ответ на второй тест даже, если оно рассчитано на решение только каких-либо подзадач из первых трех.

В других тестах  $n \neq 5$ .



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

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

#### **Подзадача 1 (20 баллов)**

$$2 \leq n \leq 50, p = 0$$

#### **Подзадача 2 (30 баллов)**

$$2 \leq n \leq 300, p = 0$$

#### **Подзадача 3 (20 баллов)**

$$2 \leq n \leq 2000, p = 0$$

#### **Подзадача 4 (30 баллов)**

$$2 \leq n \leq 2000, p = 1$$

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

По запросу сообщается баллы за каждую подзадачу.