

## Задача А. Освещение

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Близится зима. На вашей улице стоят  $M$  фонарей, а длина улицы —  $N$  метров. Улица разделена на  $N$  участков длиной по 1 метр (участки улицы пронумерованы от 1 до  $N$ ). Каждый фонарь освещает тот участок улицы, на котором стоит, а также по  $K$  участков влево и вправо от своего положения. Иначе говоря, если фонарь располагается на  $X$ -м участке, то он освещает с  $(X - K)$ -го по  $(X + K)$ -й участки включительно. Конечно, возможна ситуация, в которой один участок улицы освещается несколькими фонарями. Никакие два фонаря не могут располагаться на одном и том же участке.

Загвоздка в том, что не вся улица может быть освещена фонарями, поэтому вам требуется определить минимальное количество дополнительных фонарей, которые надо расположить вдоль улицы, чтобы осветить её всю (все участки с 1 по  $N$ ).

### Формат входных данных

В первой строке дано число  $N$  — длина улицы в метрах ( $1 \leq N \leq 1000$ ).

Во второй строке дано число  $M$  — количество фонарей на улице ( $1 \leq M \leq N$ ).

В третьей строке дано число  $K$  ( $0 \leq K \leq N$ ).

В каждой из следующих  $M$  строк дано по одному целому числу  $X_i$  ( $1 \leq X_i \leq N$ ). Эти числа даны в возрастающем порядке, они задают участки, на которых располагаются фонари.

### Формат выходных данных

Выведите единственное число — минимальное количество фонарей, которые нужно добавить, чтобы осветить всю улицу.

### Система оценки

В данной задаче кроме тестов из условия есть 10 тестов, каждый оценивается независимо в 10 баллов. Вам будет показан результат проверки на первых 7 тестах.

### Примеры

стандартный ввод	стандартный вывод
5 2 2 1 5	0
26 3 3 3 19 26	2
13 2 10 1 2	1

### Замечание

В первом примере нет необходимости добавлять фонари, так как улица уже освещена. В третьем примере необходимо добавить один фонарь, например, на 13-м участке.

## Задача В. Перестановки

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 64 мегабайта

Дана строка  $S$ . К ней приходят  $N$  запросов. В каждом запросе сообщаются 4 числа  $A, B, C, D$ . Пусть  $X$  - строка из букв  $S$ , стоящих на позициях между  $A$  и  $B$ , а  $Y$  - строка из букв  $S$ , стоящих на позициях между  $C$  и  $D$ . Требуется определить, можно ли переставить буквы в  $X$  и  $Y$  так, чтобы полученные строки стали одинаковыми.

### Формат входных данных

В первой строке вводится строка  $S$  ( $1 \leq |S| \leq 50\,000$ ).

Во второй строке дано число  $N$  ( $1 \leq N \leq 50\,000$ ) - число запросов.

В следующих  $N$  строк даны числа  $A, B, C, D$  ( $1 \leq A \leq B \leq |S|$ ,  $1 \leq C \leq D \leq |S|$ ) - параметры запроса.

### Формат выходных данных

На каждый из  $N$  запросов в отдельной строке выведите «DA» (без кавычек) если можно переставить буквы в  $X$  и  $Y$  так, чтобы  $X = Y$ , а в случае, когда так сделать нельзя, выведите «NE» (без кавычек).

### Система оценки

Решения, правильно работающие при  $1 \leq |S| \leq 1\,000$  и  $1 \leq N \leq 1\,000$  будут оцениваться в 61 балл. Вам будет сообщён результат проверки вашей программы только на этих тестах.

### Примеры

стандартный ввод	стандартный вывод
kileanimal 2 2 2 7 7 1 4 6 7	DA NE
abababba 2 3 5 1 3 1 2 7 8	DA DA
vodevovode 2 5 8 3 6 2 5 3 6	NE DA

## Задача С. Новый корабль

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Космический корабль Валериана сломался, и он решил построить новый. Планета, на которой Валериан решил построить свой космический корабль, представляет из себя клетчатое поле  $n \times m$ , часть клеток которого пригодна для строительства, а часть нет.

Корабль Валериана должен представлять из себя крест какого-то целого положительного размера  $k$ . Крест размера  $k$  — это такая клетчатая фигура, состоящая из 5 квадратов  $k \times k$  клеток, при этом есть один центральный квадрат, а остальные четыре являются его соседями по стороне.

Валериан хочет, чтобы его корабль был как можно больше, поэтому он хочет найти максимальное  $k$ , такое что он сможет построить на этой планете корабль такого размера. Поскольку планета очень большая, сам он не справится с этой задачей.

Помогите Валериану найти максимальный возможный размер корабля. Гарантируется, что он сможет построить корабль размера хотя бы 1.

### Формат входных данных

В первой строке задано два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 2000$ ) — длина и ширина планеты.

В каждой из последующих  $n$  строк задана строка, состоящая из  $m$  символов,  $j$ -й символ в  $i$ -й строке равен #, если клетка с координатами  $(i, j)$  пригодна для строительства и . иначе.

### Формат выходных данных

Выведите одно целое положительное число — максимальный возможный размер корабля.

### Система оценки

Эта задача состоит из четырех подзадач. Для подзадач выполняются дополнительные ограничения, указанные в таблице ниже. Для получения баллов за подзадачу необходимо пройти все тесты данной подзадачи, а также все тесты всех предыдущих подзадач и тесты из условия. Оффлайн групп нет.

Подзадача	Баллы	Ограничения
1	12	$1 \leq n, m \leq 20$
2	22	$1 \leq n, m \leq 300$
3	22	$1 \leq n, m \leq 700$
4	44	$1 \leq n, m \leq 2000$

## Примеры

стандартный ввод	стандартный вывод
9 12 ...##.###... ...##.###... .#####... .##### ..##### ..##### .....###... .....###... .....###...	3
6 6 .##... .##... ##### ##### .##... .##...	1

## Замечание

В первом тесте из примера можно выбрать крест размера 3. Этот крест выглядит следующим образом:

```
...##...  
...##...  
...##...  
#####  
#####  
#####  
...##...  
...##...  
...##...
```

## Задача D. Галактический апокалипсис

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Давным-давно в одной далекой-далекой галактике, было  $N$  планет. Также было  $N - 1$  межпланетных магистралей, соединявших между собой все планеты (не обязательно напрямую). Иными словами, сеть планет и магистралей образовывала дерево. Кроме того, каждая магистраль имеет свой показатель интересности, заданный неотрицательным целым числом. Пара планет  $(A, B)$  называется скучной, если выполняются следующие условия:

1.  $A$  и  $B$  — различные планеты.
2. действующей сети межпланетных магистралей существует путь между  $A$  и  $B$ .
3. Побитовый XOR показателей интересности всех магистралей в этом пути равен 0.

Ныне в галактике правит злой император, и он планирует использовать Силу, чтобы уничтожить все межпланетные магистрали в определенном порядке. Для того, чтобы спасти вселенную от гибели, вам необходимо определить количество пар скучных планет и после каждого разрушения вновь подсчитывать эту величину.

### Формат входных данных

Первая строка содержит одно целое число  $N$  ( $1 \leq N \leq 100\,000$ ).

Каждая из следующих  $N - 1$  строк содержит три целых числа  $A_i, B_i, Z_i$  ( $1 \leq A_i, B_i \leq 100\,000$ ,  $0 \leq Z_i \leq 1000\,000\,000$ ), которые означают, что планеты с номерами  $A_i$  и  $B_i$  соединены магистралью с показателем интересности  $Z_i$ .

Последняя строка содержит  $N - 1$  число: перестановку натуральных чисел от 1 до  $N - 1$ , отражающую порядок уничтожения магистралей (если  $i$ -е число в строке равно  $j$ , то император уничтожит дорогу между планетами  $A_j$  и  $B_j$  на  $i$ -м шаге).

### Формат выходных данных

Выведите  $N$  строк, в  $k$ -й строке выведите одно число — количество пар скучных планет после уничтожения  $k - 1$  дорог.

### Система оценки

Решения, работающие при  $N \leq 1000$ , будут оцениваться в 20 баллов. Решения, работающие в случае когда показатель интересности всех путей равен 0, будут оцениваться не менее чем в 30 баллов. Вам будет показан результат работы вашей программы только на 50% тестов (включая вышеуказанные).

### Примеры

стандартный ввод	стандартный вывод
2 1 2 0 1	1 0
3 1 2 4 2 3 4 1 2	1 0 0
4 1 2 0 2 3 0 2 4 0 3 1 2	6 3 1 0