

## Задача А. Олег и двоичные последовательности

Имя входного файла: `binary.in`  
Имя выходного файла: `binary.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Олег очень любит двоичные последовательности — последовательности из нулей и единиц. Совсем недавно он написал в тетради очередную двоичную последовательность из  $n$  элементов. Для выписанной последовательности Олег посчитал *Z-функцию*.

Z-функцией последовательности  $s_1, \dots, s_n$  называется массив  $z[1..n]$ , в котором:

- $z[1] = 0$ ;
- Если  $i > 1$ , то  $z[i]$  равно длине наибольшего общего префикса последовательности  $s$  и суффикса последовательности  $s$ , начинающегося с  $i$ -й позиции. Иначе говоря,  $z[i]$  равно максимальному  $k$ , такому что  $s_1 = s_i, s_2 = s_{i+1}, \dots, s_k = s_{i+k-1}$ .

Например, для последовательности  $s = \langle 0, 0, 1, 1, 0, 0, 1 \rangle$  Z-функция следующая:  $z = \langle 0, 1, 0, 0, 3, 1, 0 \rangle$ .

Записав в тетради последовательность и ее Z-функцию, Олег лег спать. Пока он спал, его младший брат Егор прокрался в комнату и закрасил фломастером последовательность и некоторые значения Z-функции. Проснувшись, Олег заинтересовался, сколько различных двоичных последовательностей он мог вечером написать в тетради, чтобы незакрашенные значения Z-функции были правильными.

Найдите число искомых последовательностей и выведите его по модулю  $10^9 + 7$ . Заметьте, что Олег мог и ошибиться при вычислении Z-функции, в этом случае ни одна последовательность не подходит и ответ равен 0.

### Формат входных данных

В первой строке входного файла находится целое число  $n$  — длина исходной двоичной последовательности ( $1 \leq n \leq 1000$ ). Во второй строке входного файла находятся  $n$  целых чисел  $z[1], \dots, z[n]$ , где  $z[i]$  — значение Z-функции в позиции  $i$ , или  $-1$ , если значение в  $i$ -й позиции было закрашено ( $-1 \leq z[i] \leq n$ ).

### Формат выходных данных

В выходной файл выведите единственное число — остаток от деления числа подходящих двоичных последовательностей на число  $10^9 + 7$ .

### Примеры

<code>binary.in</code>	<code>binary.out</code>
3 0 0 1	2
4 0 0 1 0	0
3 0 3 -1	0
3 -1 -1 -1	8

### Пояснение

В первом примере подходят последовательности  $\langle 0, 1, 0 \rangle$  и  $\langle 1, 0, 1 \rangle$ .

Во втором примере не существует ни одной двоичной последовательности длины 4 с заданной Z-функцией.

В третьем примере  $z[2] = 3$ , что противоречит определению Z-функции, поэтому ответ 0.

В четвертом примере подходит любая двоичная последовательность длины 3.

## Задача В. Город Че

Имя входного файла: `che.in`  
Имя выходного файла: `che.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

В центре города Че есть пешеходная улица — одно из самых популярных мест для прогулок жителей города. По этой улице очень приятно гулять, ведь вдоль улицы расположено  $n$  забавных памятников.

Девочке Маше из города Че нравятся два мальчика из ее школы, и она никак не может сделать выбор между ними. Чтобы принять окончательное решение, она решила назначить обоим мальчикам свидание в одно и то же время. Маша хочет выбрать два памятника на пешеходной улице, около которых мальчики будут ее ждать. При этом она хочет выбрать такие памятники, чтобы мальчики не увидели друг друга. Маша знает, что из-за тумана мальчики увидят друг друга только в том случае, если они будут на расстоянии не более  $r$  метров.

Маше заинтересовалась, а сколько способов есть выбрать два различных памятника для организации свиданий.

### Формат входных данных

В первой строке входного файла находятся два целых числа  $n$  и  $r$  ( $2 \leq n \leq 300\,000$ ,  $1 \leq r \leq 10^9$ ) — количество памятников и максимальное расстояние, на котором мальчики могут увидеть друг друга.

Во второй строке задано  $n$  положительных чисел  $d_1, \dots, d_n$ , где  $d_i$  — расстояние от  $i$ -го памятника до начала улицы. Все памятники находятся на разном расстоянии от начала улицы. Памятники приведены в порядке возрастания расстояния от начала улицы ( $1 \leq d_1 < d_2 < \dots < d_n \leq 10^9$ ).

### Формат выходных данных

Выведите одно число — число способов выбрать два памятника для организации свиданий.

### Пример

<code>che.in</code>	<code>che.out</code>
4 4 1 3 5 8	2

### Пояснение

В приведенном примере Маша может выбрать памятники 1 и 4 или памятники 2 и 4.

## Задача С. Гномы и Одинокая гора

Имя входного файла: `dwarfs.in`  
 Имя выходного файла: `dwarfs.out`  
 Ограничение по времени: 2 секунды  
 Ограничение по памяти: 256 мегабайт

Гномы продолжают искать золото предков в недрах Одинокой горы. Недра Одинокой горы представляют собой  $n$  пещер, некоторые из которых соединены двусторонними переходами. При этом из каждой пещеры в любую другую можно попасть по переходам, причем это можно сделать единственным способом.

Гномы разделились на два отряда, которые начали свои поиски с пещер  $u_0$  и  $v_0$ , соответственно. Гномы каждого из отрядов перемещаются вместе. На обследование пещеры у отряда гномов уходит ровно одна минута, после чего каждый отряд быстро перемещается по переходу в одну из соседних пещер. При этом гномы никогда не заходят в пещеру, если они или другой отряд в ней уже побывали. Оба отряда никогда не заходят в одну и ту же пещеру. Если хотя бы один из отрядов гномов не может переместиться в соответствии с этими правилами, оба отряда сразу прекращают поиски сокровищ.

Чтобы как можно лучше обследовать недра Одинокой горы, гномы хотят, чтобы поиски продолжались как можно дольше. По заданной карте пещер в Одинокой горе и начальному положению отрядов гномов определите, какое максимальное время могут продолжаться поиски сокровищ.

### Формат входных данных

В первой строке число  $n$  ( $2 \leq n \leq 200\,000$ ) — число пещер в Одинокой горе.

В следующих  $n-1$  строках заданы переходы между пещерами. В каждой строке записаны номера двух пещер  $v$  и  $u$ , соединенных переходом ( $1 \leq v, u \leq n$ ).

В следующей строке заданы номера пещер  $v_0$  и  $u_0$ , в которых исходно находятся два отряда гномов ( $1 \leq v_0, u_0 \leq n, v_0 \neq u_0$ ).

### Формат выходных данных

Выведите максимальное число минут, которое могут продолжаться поиски сокровищ.

### Примеры

<code>dwarfs.in</code>	<code>dwarfs.out</code>	Пояснение
<pre>6 1 2 2 3 3 4 4 5 5 6 4 5</pre>	2	
<pre>8 1 2 2 3 3 4 2 5 5 6 3 7 7 8 1 8</pre>	4	

## Задача D. Покраска забора

Имя входного файла: `fence.in`  
Имя выходного файла: `fence.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Том Соьер уговорил  $n$  своих друзей помочь ему в нелегком деле покраски забора, окружающего дом тетушки Полли. Забор представляет собой  $k$  последовательных досок, пронумерованных от 1 до  $k$ , причем после  $k$ -й доски опять идет первая.

Друзья Тома очень привередливы,  $i$ -й друг согласен участвовать в покраске только в том случае, если ему дадут покрасить участок из ровно  $a_i$  последовательных досок. Кисточка у Тома только одна, поэтому друзья будут красить по очереди и сразу весь отведенный им отрезок. Тому остается лишь выбрать порядок, в котором приглашать друзей, а также выбрать для каждого желаемое количество последовательных досок.

При этом каждый из друзей Тома готов красить как еще неокрашенную доску забора, так и доску, которую уже покрасил один из его предшественников. Тем не менее, друзья получают больше удовольствия от покраски неокрашенной доски. Том хочет выбрать число  $x$  и распределить отрезки забора для покраски таким образом, чтобы каждый из его друзей покрасил хотя бы  $x$  неокрашенных досок. Том очень любит своих друзей и хочет, чтобы каждый из них получил от процесса покраски забора максимальное удовольствие, поэтому он пытается максимизировать  $x$ .

Помогите Тому понять, сколько радости он сможет доставить своим друзьям.

### Формат входных данных

Первая строка входного файла содержит два целых числа  $n$  ( $1 \leq n \leq 10^5$ ) и  $k$  ( $1 \leq k \leq 10^9$ ). Следующая строка содержит  $n$  целых чисел — значения  $a_i$  ( $1 \leq a_i \leq k$ ).

### Формат выходных данных

Выведите одно число — максимальное возможное значение  $x$ .

### Примеры

<code>fence.in</code>	<code>fence.out</code>
2 100 5 10	5
4 10 7 8 3 5	2

### Пояснение

В первом примере  $x = 5$ , так как один из друзей просто не хочет красить больше пяти досок. Он придет первым, покрасит свои пять, после чего еще 10 неокрашенных досок достанется второму другу Тома. Оставшиеся 85 досок Тому придется красить самому.

Во втором примере достичь  $x = 2$  можно, например, так. Сначала третий друг красит доски с 4 по 6 (3 неокрашенных доски). Затем четвертый друг красит доски с 1 по 5 (3 неокрашенных доски). Затем второй друг красит доски с 1 по 8 (2 неокрашенных доски). Наконец, первый друг красит доски с 6 по 10 и с 1 по 2 (2 неокрашенных доски, заметим, что забор идет по циклу и эти доски образуют последовательный отрезок).

## Задача Е. Телефонные номера

Имя входного файла:	<code>numbers.in</code>
Имя выходного файла:	<code>numbers.out</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Неотъемлемой частью программного обеспечения любого мобильного телефона является записная книжка. В самых первых мобильных телефонах в ней можно было хранить лишь имена абонентов и их телефонные номера. В более современных моделях в ней можно также хранить множество другой полезной информации — электронный адрес человека, его фотографию, ссылки на его страницы в социальных сетях и дату его рождения. Однако, при реализации даже самой простой версии этой программы, позволяющей работать только с телефонными номерами, разработчики иногда сталкиваются с некоторыми сложностями.

Будем считать, что любой телефонный номер состоит из 11 цифр и делится на три части, каждая из которых является числом без ведущих нулей. Первая часть состоит из одной, двух или трех цифр и является кодом страны, в которой этот телефон зарегистрирован. Вторая часть может состоять из трех, четырех или пяти цифр, и может являться или кодом региона, в котором зарегистрирован номер, или кодом мобильного оператора, которому этот номер принадлежит. Третья часть состоит из всех оставшихся цифр номера и является номером конкретного абонента.

При отображении телефонного номера на экране телефона, части этого номера принято отделять друг от друга различными символами так, чтобы номер было проще прочесть и запомнить. Так, перед кодом страны обычно ставится символ «+», код региона или оператора берется в скобки, номер абонента разделяется символами «-» на несколько частей. При этом, то, на сколько частей он разбивается, напрямую зависит от количества цифр в нем:

- если номер абонента состоит из трех цифр, то он представляет собой одну часть, состоящую из трех цифр;
- если номер абонента состоит из четырех цифр, то он разбивается на две части, каждая из которых состоит из двух цифр;
- если номер абонента состоит из пяти цифр, то он разбивается на две части, первая из которых состоит из трех цифр, а вторая — из двух;
- если номер абонента состоит из шести цифр, то он разбивается на три части, каждая из которых состоит из двух цифр;
- если номер абонента состоит из семи цифр, то он разбивается на три части, первая из которых состоит из трех цифр, а все остальные — из двух.

Естественно, что человек, заносивший новый номер в записную книжку своего телефона, не станет задумываться об этих правилах, а просто введет его как последовательность из 11 цифр. Однако, перед отображением номера на экране, программное обеспечение телефона должно выяснить, какая часть этого номера является кодом страны, какая — кодом региона или оператора, а какая — номером абонента, и отформатировать номер по правилам, описанным выше. Чтобы сделать эту задачу разрешимой, в память телефона записывается информация о том, какие в данный момент существуют коды государств и какие в этих государствах существуют коды операторов или регионов. Вам необходимо реализовать программу, которая, имея эту информацию, будет правильно форматировать номера, сохраненные в записной книжке телефона.

### Формат входных данных

Первая строка файла содержит одно целое число  $n$  ( $1 \leq n \leq 100$ ) — количество государств, информация про телефонные коды которых записана в память телефона. Далее следуют  $n$  описаний этих государств, разделенных переводами строк.

Первая строка описания каждого государства содержит два целых числа  $c$  и  $k$  ( $1 \leq c \leq 999$ ,  $1 \leq k \leq 100$ ) — телефонный код этого государства и количество операторов или регионов,

существующих в этом государстве. Следующие  $k$  строк описания этого государства содержат целые числа, каждое из которых не меньше 100 и не больше 99999 — коды операторов или регионов, зарегистрированных в этом государстве.

Следующая строка входного файла содержит одно целое число  $m$  ( $1 \leq m \leq 10\,000$ ) — количество телефонных номеров, которые необходимо отформатировать. Следующие  $m$  строк содержат сами номера — строки, состоящие ровно из 11 цифр.

Гарантируется, что ни один данный вам номер невозможно разбить на код государства, код оператора или региона и номер абонента более, чем одним способом.

### Формат выходных данных

Выведите номера, данные вам во входном файле, отформатированными по правилам, описанным в условии. Каждый номер необходимо вывести в отдельной строке. Номера необходимо выводить в том же порядке, в котором они были перечислены во входном файле.

Вместо номеров, корректного разбиения которых на код государства, код оператора или региона и номер абонента не существует, необходимо вывести слово «Incorrect».

### Примеры

numbers.in	numbers.out
2	+7(981)826-64-56
7 3	+351(963)284-63
981	+7(812)247-25-57
3517	Incorrect
812	+7(3517)96-03-26
351 3	+351(34712)239
34712	+351(1234)20-13
1234	Incorrect
963	
8	
79818266456	
35196328463	
78122472557	
01234567890	
73517960326	
35134712239	
35112342013	
78120203040	

## Задача F. Вращающаяся пластина

Имя входного файла: `polygon.in`  
Имя выходного файла: `polygon.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Мальчик Вася очень любит геометрию. Кроме того, ему очень нравится забивать гвозди в доску. Сегодня он изучает свою любимую металлическую пластину, которую он собирается прибить к деревянной доске.

Пластина имеет форму, ограниченную многоугольником без самопересечений и самокасаний. В первой вершине многоугольника пластина имеет маленькую петлю.

Сначала Вася кладет пластину на доску, лежащую на горизонтальном столе, и прибывает ее гвоздем к пластине через петлю в первой вершине. Вбитый гвоздь жестко закрепляет положение первой вершины, однако позволяет пластине вращаться вокруг нее.

Теперь Вася хочет забить в доску еще один гвоздь, который бы ограничил вращение пластины, но он еще не решил куда. Он наметил  $m$  возможных позиций, куда он хотел бы забить гвоздь, и для каждой из них хочет узнать, на какой угол он сможет поворачивать свою пластину, если забьет в доску гвоздь в этой точке. При этом как только пластина касается гвоздя, дальше ее поворачивать нельзя.

### Формат входных данных

В первой строке входного файла задано число  $n$  ( $3 \leq n \leq 2000$ ) — количество вершин многоугольника. В следующих  $n$  строках заданы координаты вершин многоугольника в порядке обхода.

В следующей строке задано число  $m$  ( $1 \leq m \leq 2000$ ) — количество точек, которые Вася рассматривает как возможные положения второго гвоздя. В следующих  $m$  строках заданы координаты этих точек. Все эти точки находятся снаружи от исходного положения пластины.

Все координаты во входном файле целые и не превосходят  $10^6$  по модулю.

### Формат выходных данных

Выходной файл должен содержать  $m$  строк.

В  $i$ -й строке выведите два вещественных числа  $\alpha_i$  и  $\beta_i$ , где  $\alpha_i$  — максимальный угол в градусах, на который можно повернуть пластину по часовой стрелке, если Вася забьет гвоздь в  $i$ -ю точку, а  $\beta_i$  — максимальный угол в градусах, на который в этом случае можно повернуть пластину против часовой стрелки.

Если гвоздь не мешает пластине поворачиваться, выведите  $\alpha_i = \beta_i = 360$ .

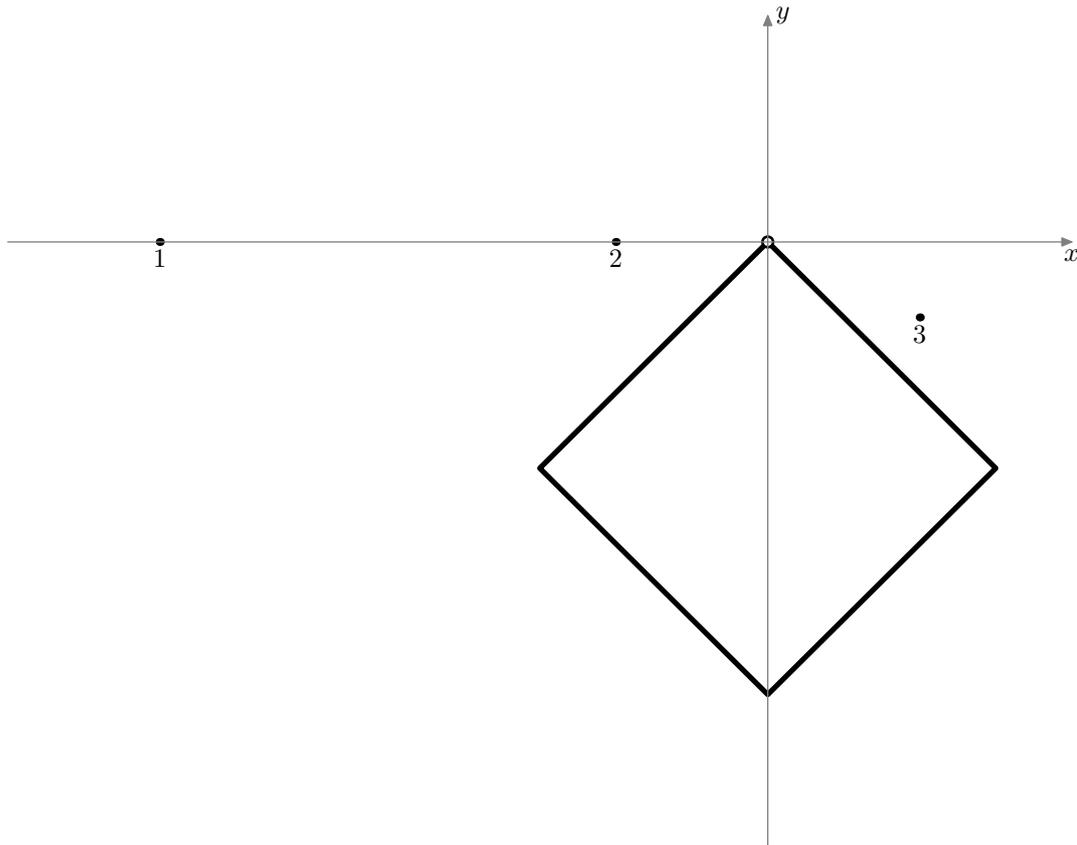
Ответ считается верным, если его абсолютная или относительная погрешность относительно правильного не превосходит  $10^{-6}$ .

### Примеры

<code>polygon.in</code>	<code>polygon.out</code>
4	360.000000000000 360.000000000000
0 0	45.000000000000 225.000000000000
-3 -3	251.565051177078 18.434948822922
0 -6	
3 -3	
3	
-8 0	
-2 0	
2 -1	

Пояснение к примеру на следующей странице.

## Пояснение



На рисунке выше изображен тест из примера. Прямоугольник показывает начальное положение пластины. Точками показаны позиции, в которые Вася планирует забить гвоздь.

Гвоздь, забитый в первую точку, не мешает пластине поворачиваться.

Гвоздь, забитый во вторую точку, позволяет повернуть пластину на  $45^\circ$  по часовой стрелке или на  $225^\circ$  против часовой стрелки.

## Задача G. Призы

Имя входного файла: prizes.in  
Имя выходного файла: prizes.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Организаторы Всероссийской командной олимпиады школьников по программированию всегда ответственно относятся ко всем этапам проведения соревнования. Недавно организаторам были доставлены футболки для участников олимпиады. Они были сложены в ящик, который является кубом со стороной в один метр. Ящик был поставлен в углу комнаты прямоугольной формы размером  $m \times n$  метров. Чтобы никто случайно не забрал ящик, на его верхней грани красной краской написали слово «ВКОШП».

Сегодня организаторам вдруг понадобилось переставить этот ящик в противоположный угол комнаты. Но, к сожалению, ящик оказался настолько тяжелым, что никто не мог сдвинуть его с места. Выяснилось, что все, что можно сделать с ящиком — перекатить его через ребро нижней грани. Соответствующее ребро при этом остается на том же месте, а нижней становится другая смежная с этим ребром грань.

Перед организаторами олимпиады встала следующая задача. Им необходимо перекатить ящик из угла комнаты, в котором он стоит, в противоположный угол. При этом даже перекачивать ящик очень тяжело, поэтому организаторы решили минимизировать количество перекачиваний ящика.

Но когда они уже собрались начать транспортировку, обнаружилась еще одна проблема. Красная надпись «ВКОШП» каждый раз, касаясь пола, оставляет на нем следы. Поэтому среди всех вариантов транспортировки, минимизирующих количество перекачиваний, организаторы решили выбрать тот, при котором надпись «ВКОШП» окажется на нижней грани куба минимальное число раз.

Помогите организаторам — посчитайте, сколько раз надпись «ВКОШП» коснется пола при оптимальном перекачивании куба с футболками.

### Формат входных данных

В первой строке задано два числа  $n$  и  $m$  ( $1 \leq n, m \leq 10^9$ ) — размеры комнаты в метрах.

### Формат выходных данных

Выведите одно число — сколько раз надпись «ВКОШП» окажется на нижней грани при оптимальном перемещении ящика.

### Примеры

	prizes.in	prizes.out
1 2		0
3 3		1

### Пояснение

В первом примере необходимо одно перекачивание, надпись, которая исходно была на верхней грани, окажется на боковой грани, но пола не коснется.

Во втором примере необходимо четыре перекачивания. В любом случае хотя бы один раз надпись «ВКОШП» коснется пола. Один из способов сделать перекачивания так, чтобы это произошло один раз, следующий. Сначала два раза перекатим куб в одном направлении (он окажется в соседнем углу комнаты). Сейчас надпись «ВКОШП» находится на нижней грани и касается пола. Затем перекатим куб еще два раза в перпендикулярном направлении. Теперь куб находится в требуемом положении.

## Задача Н. Защищенное соединение

Имя входного файла: `secure.in`  
Имя выходного файла: `secure.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

В свете недавних новостей о прослушке каналов связи, два непримиримых интернет-гиганта Урагании «Laim.UR» и «Xenda» решили подписать соглашение об установлении защищенного канала связи между дата-центрами друг друга. В Урагании  $n$  городов, но, к сожалению, ни в одном городе нет дата-центров обоих гигантов. Поэтому для формирования защищенного канала придется прокладывать междугородные линии связи.

Специалисты компаний определили  $m$  пар городов, которые можно соединить, проложив сегмент канала связи, и оценили стоимость создания такого сегмента для каждой из этих пар.

Результирующий канал может состоять из нескольких сегментов. Он должен начинаться в одном из городов, где находится дата-центр первой компании, может проходить через промежуточные города и должен заканчиваться в городе, где находится дата-центр второй компании.

Теперь необходимо определить минимальную стоимость защищенного канала, соединяющего два дата-центра компаний.

### Формат входных данных

В первой строке находятся целые числа  $n$  и  $m$  ( $2 \leq n \leq 5000$ ,  $1 \leq m \leq 10^5$ ) — количество городов и количество пар городов, которые можно соединить сегментом канала связи.

Во второй строке находятся  $n$  целых чисел  $a_i$  ( $0 \leq a_i \leq 2$ ). Если  $a_i = 0$ , то в  $i$ -м городе нет дата-центра ни одного из гигантов. Если  $a_i = 1$ , то в  $i$ -м городе есть дата-центр «Laim.UR», а если  $a_i = 2$ , то в  $i$ -м городе находится дата-центр «Xenda». Гарантируется, что среди этих чисел есть как минимум одна единица и одна двойка.

В каждой из следующих  $m$  строк находится по три целых числа —  $s_i$ ,  $t_i$  и  $c_i$ , которые означают, что города  $s_i$  и  $t_i$  ( $1 \leq s_i, t_i \leq n$ ,  $s_i \neq t_i$ ) можно соединить сегментом канала связи стоимостью  $c_i$  ( $1 \leq c_i \leq 10^5$ ). Каждую пару городов можно соединить не более чем одним сегментом канала.

### Формат выходных данных

Если соединить защищенным каналом связи два дата-центра разных интернет-гигантов возможно, то выведите в выходной файл три числа:  $x$ ,  $y$  и  $d$ , означающие, что между городами  $x$  и  $y$  возможно провести канал связи суммарной стоимостью  $d$ . В городе  $x$  должен находиться дата-центр «Laim.UR», в городе  $y$  — дата-центр «Xenda». Если существует несколько оптимальных ответов, выведите любой. Если провести искомый канал невозможно, выведите  $-1$ .

### Примеры

<code>secure.in</code>	<code>secure.out</code>
6 7 1 0 1 2 2 0 1 3 3 1 2 4 2 3 3 2 4 2 1 6 5 3 5 6 5 6 1	3 4 5
4 2 1 0 0 2 1 3 3 2 4 2	-1

### Пояснение

В первом примере оптимально построить канал связи из двух сегментов:  $3 - 2$  и  $2 - 4$ .

## Задача I. Перетягивание каната

Имя входного файла: `sochi.in`  
Имя выходного файла: `sochi.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

В 2086 году в программу зимней олимпиады решено было добавить соревнования по перетягиванию каната на льду. Для проведения финала соревнования организаторы нашли  $n$  кусков каната. Для повышения зрелищности соревнования решено было сделать связать некоторые из этих кусков в один как можно более длинный канат.

Когда начались работы по связыванию, выяснилось, что на узел, связывающий два куска каната между собой, уходит по  $d$  сантиметров каната с каждого из связываемых концов. Также, оказалось, что связывать так, что получающиеся узлы находятся близко друг к другу, невозможно: расстояние между соседними узлами должно быть хотя бы  $d$  сантиметров. Например, если  $d = 10$ , то после связывания кусков каната длиной 25 и 50 сантиметров, получается канат длиной 55 сантиметров, в 15 сантиметрах от одного из краев которого находится узел.

До начала соревнований остается мало времени, поэтому они обратились за помощью к вам. Помогите организаторам выяснить, канат какой максимальной длины удастся получить.

### Формат входных данных

В первой строке заданы числа  $n$  ( $1 \leq n \leq 100\,000$ ) и  $d$  ( $1 \leq d \leq 1000$ ) — количество кусков каната и длина каната, уходящая на завязывание узла.

Во второй строке заданы  $n$  чисел  $a_i$  ( $1 \leq a_i \leq 1000$ ) — длины имеющихся кусков каната.

### Формат выходных данных

Выведите единственное число — максимальную длину каната, которую можно получить.

### Примеры

<code>sochi.in</code>	<code>sochi.out</code>
2 10 25 50	55
5 2 4 5 6 7 8	14

## Задача J. Загранпаспорт

Имя входного файла: `travel.in`  
Имя выходного файла: `travel.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Осень 2243-го года. Государства в прошлом, главным территориальным образованием является *автономия*. В результате терраформирования территория бывшей Евразии теперь представляет собой прямоугольник, он разбит на  $w \times h$  квадратных автономий, которые организованы в виде сетки из  $w$  автономий по ширине и  $h$  по высоте. Некоторые автономии входят в содружество Крипто, они представляют собой криптоанархистские технократические общества и не признают бюрократии. Остальные автономии входят в конфедерацию Бюро, бюрократия в них доведена до высшего совершенства.

Для перемещения между автономиями необходим загранпаспорт. При въезде в автономию содрущества Крипто предъявлять документы не нужно, но при въезде в автономию конфедерации Бюро пограничная служба проверяет документ, заполняет специальные формы и проставляет в загранпаспорт путешественника штамп своей автономии. Один штамп занимает одну страницу паспорта, разные штампы ставятся на разные страницы.

Опытный путешественник по имени Вениамин хочет получить загранпаспорт нового образца. Чтобы это сделать досрочно, ему нужно заполнить все страницы своего старого паспорта штампами пограничных служб. На данный момент у него осталось  $n$  пустых страниц, соответственно ему нужно ровно  $n$  раз въехать в автономию Бюро. Он хочет сделать это как можно быстрее.

Вениамин путешествует таким образом, что за один день он переезжает из автономии, в которой он находится, в автономию, имеющую с ней общую сторону. Помогите Вениамину спланировать свое путешествие так, чтобы заполнить все оставшиеся страницы загранпаспорта и при этом сделать как можно меньше переездов между автономиями. Вениамин начинает свое путешествие в родной автономии, которая принадлежит содружеству Крипто, а закончить путешествие может в любой автономии.

### Формат входных данных

В первой строке входного файла содержатся три целых числа:  $w$ ,  $h$  и  $n$  ( $1 \leq w, h \leq 500$ ,  $1 \leq n \leq 1000$ ).

В каждой из следующих  $h$  строк содержится по  $w$  символов, они задают карту Евразии. Символ «А» обозначает автономию содрущества Крипто, а «Т» — автономию конфедерации Бюро. Автономия, в которой Вениамин начинает свое путешествие, обозначена символом «V».

Карта дана с севера на юг по строкам и с запада на восток по столбцам, таким образом, первый символ второй строки входного файла описывает самую северо-западную автономию.

Гарантируется, что в Евразии есть хотя бы одна автономия Бюро.

### Формат выходных данных

В выходной файл выведите одну строку из символов «N», «E», «S», «W» — план путешествия Вениамина. Эти символы означают, что Вениамину следует поехать на север, восток, юг или запад, соответственно. Число перемещений в плане необходимо минимизировать, в процессе путешествия Вениамин должен ровно  $n$  раз въехать в автономию Бюро. Если возможных оптимальных планов путешествия несколько, можно вывести любой.

### Пример

<code>travel.in</code>	<code>travel.out</code>
5 3 6 AAATA VAATA AAAAT	EEENSNSN
3 1 2 TVT	WEE

## Задача К. Вирус

Имя входного файла: `virus.in`  
Имя выходного файла: `virus.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

В лаборатории биоинформатики ученые проводят эксперименты по распространению искусственно созданных вирусов. Для эксперимента используется специальная лабораторная установка, представляющая собой таблицу из  $n \times m$  ячеек. В каждую ячейку помещается живая клетка. Ученые заражают вирусом некоторые клетки, всего исходно заражается не более 8 клеток.

Каждую секунду среди незараженных клеток, имеющих зараженную клетку в соседней по стороне ячейке, ровно одна клетка заражается вирусом.

Ученые заинтересовались, какие конфигурации зараженных клеток могут получиться через  $t$  секунд. Для начала они хотят посчитать число таких конфигураций. Помогите им это сделать.

### Формат входных данных

В первой строке входного файла находятся целые числа  $n$ ,  $m$  и  $t$  ( $1 \leq n, m \leq 100$ ,  $1 \leq t \leq 6$ ) — размеры таблицы и количество секунд.

Каждая из следующих  $n$  строк содержит  $m$  символов. Символ «.» означает, что в изначальной конфигурации клетка не заражена, а символ «\*» — что заражена. Количество «\*» в таблице не превышает 8.

Гарантируется, что незараженных клеток в исходной конфигурации не меньше  $t$ .

### Формат выходных данных

Выведите количество различных возможных конфигураций таблицы после  $t$  секунд.

### Примеры

<code>virus.in</code>	<code>virus.out</code>
2 2 1 *. ..	2
2 2 2 *. ..	3
2 2 3 *. ..	1