

## Задача А. Перерыв на обед

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Супершион Лэнс Стерлинг находится в городе и отправляется на задание. Для удобства, введем в городе прямоугольную систему координат, единичный отрезок которой имеет длину один метр. Сейчас Лэнс находится в тайном убежище в точке с координатами  $(x_s, y_s)$ , ему требуется добраться до точки с координатами  $(x_t, y_t)$ . Он может перемещаться со скоростью один метр в секунду в любом направлении. Однако, он очень проголодался, поэтому решил перед заданием заскочить перекусить.

Всего в городе есть  $n$  заведений, в которых Лэнс может пообедать. Заведение номер  $i$  находится в точке  $(x_i, y_i)$ , и Лэнс знает, что он проведет в нем ровно  $t_i$  секунд. Помогите ему определить минимальное время, которое потребуется, чтобы добраться до точки назначения, зайдя по пути перекусить в одно из заведений.

### Формат входных данных

В первой строке даны четыре целых числа  $x_s, y_s, x_t$  и  $y_t$  — координаты тайного убежища и точки назначения ( $|x_s|, |y_s|, |x_t|, |y_t| \leq 1000$ ).

В следующей строке дано одно целое число  $n$  — количество заведений, где Лэнс может пообедать ( $1 \leq n \leq 1000$ ).

В следующих  $n$  строках дано по три целых числа  $x_i, y_i$  и  $t_i$  — координаты  $i$ -го заведения и количество секунд, которое придется потратить, чтобы пообедать в этом заведении ( $|x_i|, |y_i| \leq 1000$ ;  $1 \leq t_i \leq 1000$ ).

### Формат выходных данных

Выведите одно вещественное число — минимальное количество секунд, которое потребуется Лэнсу, чтобы добраться из убежища до точки назначения, зайдя по пути перекусить. Ответ будет считаться правильным, если его абсолютная или относительная погрешность не будет превышать  $10^{-9}$ .

### Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи	Информация о проверке
1	20	$y_s, y_t, y_i = 0$		первая ошибка
2	20	$n = 1$		первая ошибка
3	60	Нет дополнительных ограничений	1, 2	первая ошибка

### Примеры

стандартный ввод	стандартный вывод
0 0 10 0 1 5 0 3	13.000000000000000
0 -5 0 -3 1 0 5 10	28.000000000000000
0 0 5 5 2 3 3 2 3 4 1	8.236067977499790

## Задача В. Починка массива

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

У Уолтера Беккета был замечательный отсортированный массив, однако, после множества экспериментов произошло непредвиденное: а именно, массив перестал быть отсортированным!

Казалось бы, что сложного в том, чтобы отсортировать массив? Но Уолтер и здесь решил провести эксперимент. Он хочет отсортировать массив используя только две операции:

- Взять любой элемент массива и переместить его в **конец** массива.
- Взять любой элемент массива и переместить его в **начало** массива.

Таким образом, если массив изначально содержал элементы  $a_1, a_2, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_n$  и был выбран  $i$ -й элемент, то если применить первую операцию, массив станет выглядеть как  $a_1, a_2, \dots, a_{i-1}, a_{i+1}, \dots, a_n, a_i$ , а в случае применения второй операции — как  $a_i, a_1, a_2, \dots, a_{i-1}, a_{i+1}, \dots, a_n$ .

Оказалось, что с помощью этих двух операций всегда можно отсортировать массив, что Уолтер и сделал со своим массивом. Но теперь Уолтер дал вам новый массив и попросил найти наименьшее количество таких операций, необходимых, чтобы отсортировать новый массив.

### Формат входных данных

В первой строке содержится одно целое число  $n$  — длина массива, который вам дал Уолтер ( $1 \leq n \leq 300\,000$ ).

Во второй строке заданы  $n$  целых чисел  $a_i$ , разделенных пробелами — элементы массива ( $1 \leq a_i \leq 10^9$ ).

### Формат выходных данных

Выведите единственное число — минимальное число операций, которые нужно применить к данному массиву, чтобы он стал отсортированным.

### Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи	Информация о проверке
1	10	$n \leq 10$		полная
2	10	$n \leq 300$ и все $a_i$ — различны		первая ошибка
3	15	$n \leq 5\,000$ и все $a_i$ — различны	2	первая ошибка
4	20	Все $a_i$ — различны	2, 3	первая ошибка
5	10	$n \leq 300$	1, 2	первая ошибка
6	15	$n \leq 5\,000$	1, 2, 3, 5	первая ошибка
7	20	Без дополнительных ограничений	1, 2, 3, 4, 5, 6	первая ошибка

## Примеры

стандартный ввод	стандартный вывод
5 3 1 2 4 5	2
5 5 4 3 2 1	4
6 2 3 1 6 4 5	2

## Замечание

В первом тесте можно переставить 2 в начало, а затем 1 в начало и массив будет отсортирован за две операции.

Во втором тесте можно оставить 5 на месте, а все остальные элементы по очереди переставить в начало. А можно оставить 1 на месте, а все остальные элементы переставить в конец. В обоих случаях придется потратить минимум четыре операции.

В третьем тесте достаточно переставить 1 в начало, а 6 в конец. Итого две операции.

## Задача С. Кодовый замок

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

На очередном задании Лэнс Стерлинг в очередной раз столкнулся с запертым сейфом. «Да сколько можно, снова взламывать», — подумал про себя супершпион.

Данный сейф закрывается на кодовый замок, который может быть представлен в виде таблицы  $n \times n$ . В этой таблице есть клетки трех типов:

- Пустые
- Клетки, содержащие *центральные элементы*
- Клетки, содержащие *поворотные ручки*

Причем, из-за конструктивных особенностей сейфа, в каждом столбце и каждой строке таблицы содержится не более одного центрального элемента.

Каждую из ручек можно ориентировать одним из двух способов: вертикально или горизонтально. Для того, чтобы замок открылся, ручки нужно ориентировать таким образом, чтобы от каждой ручки можно было добраться до центрального элемента, переходя только по ручкам ориентированным таким же образом, в направлении ориентации. Иными словами, если ручка ориентирована вертикально, должен существовать центральный элемент, находящийся на одной вертикали с этой ручкой, и все клетки между этим центральным элементом и этой ручкой должны быть заняты ручками, ориентированными вертикально. Аналогично для всех горизонтальных ручек.

Лэнсу нельзя терять ни минуты, поэтому он попросил помощи с открытием сейфа у вас. Помогите ему определить ориентации ручек, при которых сейф откроется, либо сообщите, что это невозможно.

### Формат входных данных

В первой строке дано одно целое число  $n$  — размер таблицы ( $1 \leq n \leq 500$ ).

Далее в  $n$  строках дано по  $n$  символов — описание таблицы. Символ «.» соответствует пустой клетке, символ «O» (заглавная английская буква O, ASCII код 79) — центральному элементу, а «+» — поворотной ручке.

Гарантируется, что в каждой строке и каждом столбце таблицы содержится не более одного центрального элемента.

### Формат выходных данных

Если не существует способа ориентировать ручки, чтобы сейф открылся, выведите «No». Иначе, выведите «Yes», а в следующих  $n$  строках по  $n$  символов — описание таблицы, аналогичное описанию во входных данных. Ручки, которые нужно ориентировать вертикально, должны быть обозначены символами «|», а ручки, которые нужно ориентировать горизонтально — символами «-».

Если существует несколько различных ответов, выведите любой из них.

### Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Обозначим за  $C$  количество центральных элементов в таблице.

Подзадача	Баллы	Ограничения	Необходимые подзадачи	Информация о проверке
1	10	$C \leq 1, n \leq 100$		полная
2	20	$C \leq 5, n \leq 100$	1	первая ошибка
3	30	$n \leq 100$	1, 2	первая ошибка
4	40	Без дополнительных ограничений	1, 2, 3	первая ошибка

## Примеры

стандартный ввод	стандартный вывод
3 0++ +. + ++0	Yes 0-     .   --0
4 .. +. ... ..0. ..+.	No
3 .+. 0++ .0.	No

## Задача D. Побег с горной базы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

После того как Лэнс Стерлинг украдет чемодан с горной базы, ему предстоит вернуться в штаб. Самый безопасный способ это сделать — съехать на лыжах по горе и сесть в вертолёт, который заранее будет стоять на одной из плоских полей, расположенных на горе.

Всего на горе есть  $n$  полей, которые пронумерованы от 1 до  $n$  в порядке уменьшения высоты. Поляна с номером 1 находится выше всего, а на каждую другую поляну ведет тропа с ровно одной поляны, которая находится выше. Лэнс может скатываться на лыжах только по тропам и только с более высокой поляны на более низкую.

У Лэнса в распоряжении есть  $k$  вертолетов. Он собирается расставить их на некоторых полянах. Лэнс еще не знает, на какой из полей он окажется сразу после побега. Поэтому, хочет расставить вертолеты таким образом, чтобы количество полей, с которых он может добраться до какого-нибудь вертолета, было максимально.

Сейчас они с Уолтером прорабатывают план, и Лэнс заинтересовался, чему равно максимальное количество полей, с которых можно будет добраться до вертолета, если расставить вертолеты оптимальным образом.

### Формат входных данных

В первой строке даны два числа  $n$  и  $k$  — количество полей на горе и вертолётов в распоряжении у Лэнса ( $1 \leq k \leq n \leq 300\,000$ ).

В следующей строке даны  $n - 1$  целых чисел  $p_2, p_3, \dots, p_n$ . Число  $p_i$  означает, что тропа, ведущая на поляну  $i$ , начинается на поляне  $p_i$  ( $1 \leq p_i < i$ ).

### Формат выходных данных

Выведите одно число — максимальное количество полей, с которых можно будет добраться до вертолета при оптимальной расстановке вертолетов.

### Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Дополнительные ограничения	Необх. подзадачи	Информация о проверке
1	10	$n \leq 20$			первая ошибка
2	20	$n \leq 300$		1	первая ошибка
3	10	$n \leq 2000$	До всех полей можно добраться от первой, проехав по не более чем 20 тропам	1	первая ошибка
4	30	$n \leq 5000$		1, 2, 3	первая ошибка
5	30	$n \leq 300000$		1, 2, 3, 4	первая ошибка

### Пример

стандартный ввод	стандартный вывод
7 2 1 2 3 2 5 1	6