

Задача А. Смерть

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	1024 мегабайта

Смерть — Четвертый всадник Апокалипсиса, и за этим всадником следует ад. Однако, даже этот всадник готов пощадить некоторые города и оставить их жителей в живых.

Карта страны, которую изучает Смерть, представляет собой клетчатый прямоугольник размера $n \times m$. Каждая клетка — город, и в каждом городе живут люди, подчиняющиеся одному определенному лорду. Смерть хочет пощадить несколько городов так, чтобы выполнялись два правила:

- из любого выжившего города можно добраться в любой другой, перемещаясь только по выжившим городам, соседним по стороне
- в выживших городах проживают люди, подчиняющиеся не более, чем двум различным лордам

Теперь Смерть заинтересовало максимальное количество городов, которые он может пощадить.

Формат входных данных

В первой строке входного файла задано два целых числа n и m ($1 \leq n, m \leq 10^3$) — размеры страны. Следующие n строк содержат по m чисел каждая — номера лордов, которым подчиняются люди в соответствующих городах. Номера лордов — натуральные числа, не превышающие 10^6 .

Формат выходных данных

Выведите одно натуральное число — максимальное количество городов, которые смогут выжить после набега смерти.

Система оценки

Решения, работающие в случаях, в которых n и m не превышают 20, будут оцениваться в 33 баллов.

Решения, работающие в случаях, в которых n и m не превышают 80, будут оцениваться в 66 баллов.

Пример

стандартный ввод	стандартный вывод
5 5 1 1 2 2 1 1 2 2 1 1 1 2 3 3 1 1 3 3 2 4 1 2 2 4 1	17

Замечание

В первом примере надо оставить некоторых людей, подчиняющихся 1 и 2 лордам.

Задача В. Захват вселенной

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Самая закрытая олимпиада страны кончилась трагедией! Самый главный знаток «SEERC», создатель популярного сервиса «SnurkMems», а в свободное время составитель планов по захвату вселенной, не смог сделать правильную таблицу результатов. Чтобы отойти от этой тяжелейшей трагедии, он решил забросить всю свою обыкновенную работу и полностью сконцентрироваться на составлении планов по захвату вселенной.

Как оказалось, захват вселенной — дело совсем не сложное. Для этого надо решить следующую задачу. Дано натуральное число M и S — перестановка всех чисел от 0 до $2^M - 1$. Для успешного захвата вселенной нужно выбрать какую-то непустую подпоследовательность подряд идущих чисел из S такую, чтобы существовал способ поменять числа на 2-х различных местах из исходной последовательности S , чтобы после этого побитовый **XOR** выбранной подпоследовательности равнялся $2^M - 1$.

Наш герой хочет знать про все возможные планы захвата вселенной, поэтому ему интересно число таких подпоследовательностей подряд идущих чисел из S , что условия, описанные выше, выполняются, и вселенную можно будет захватить. Помогите ему найти это число, чтобы наш герой снова вернулся к своей обычной работе и «SnurkMems» продолжал нас радовать полезной актуальной информацией.

Формат входных данных

В первой строке дано число M ($1 \leq M \leq 20$).

Во второй строке даны 2^M чисел из S — перестановка всех чисел от 0 до $2^M - 1$.

Формат выходных данных

В единственной строке выведите одно число — ответ на задачу.

Система оценки

Всего в задаче 10 групп тестов, каждая оценивается в 10 баллов, в каждой следующей m больше, чем в предыдущей и соответственно каждая оценивается только если все предыдущие пройдены.

Чтобы вы примерно ориентировались в баллах сообщаем, что группы, в которых $1 \leq M \leq 14$, будут оцениваться в 50 баллов.

Примеры

стандартный ввод	стандартный вывод
2 0 1 2 3	9
3 3 7 0 4 6 1 5 2	33
4 13 0 15 12 4 8 7 3 11 14 6 10 1 5 9 2	133

Замечание

В первом примере, если взять подпоследовательность $(1, 2, 3)$, с исходным **XOR** равным 0, можно заменить числа 0 и 3, после чего последовательность будет представлять из себя $(1, 2, 0)$, и её **XOR** станет равным 3. И вообще наш герой может фактически выиграть для каждой выбранной подпоследовательности подряд идущих чисел, кроме всей последовательности.

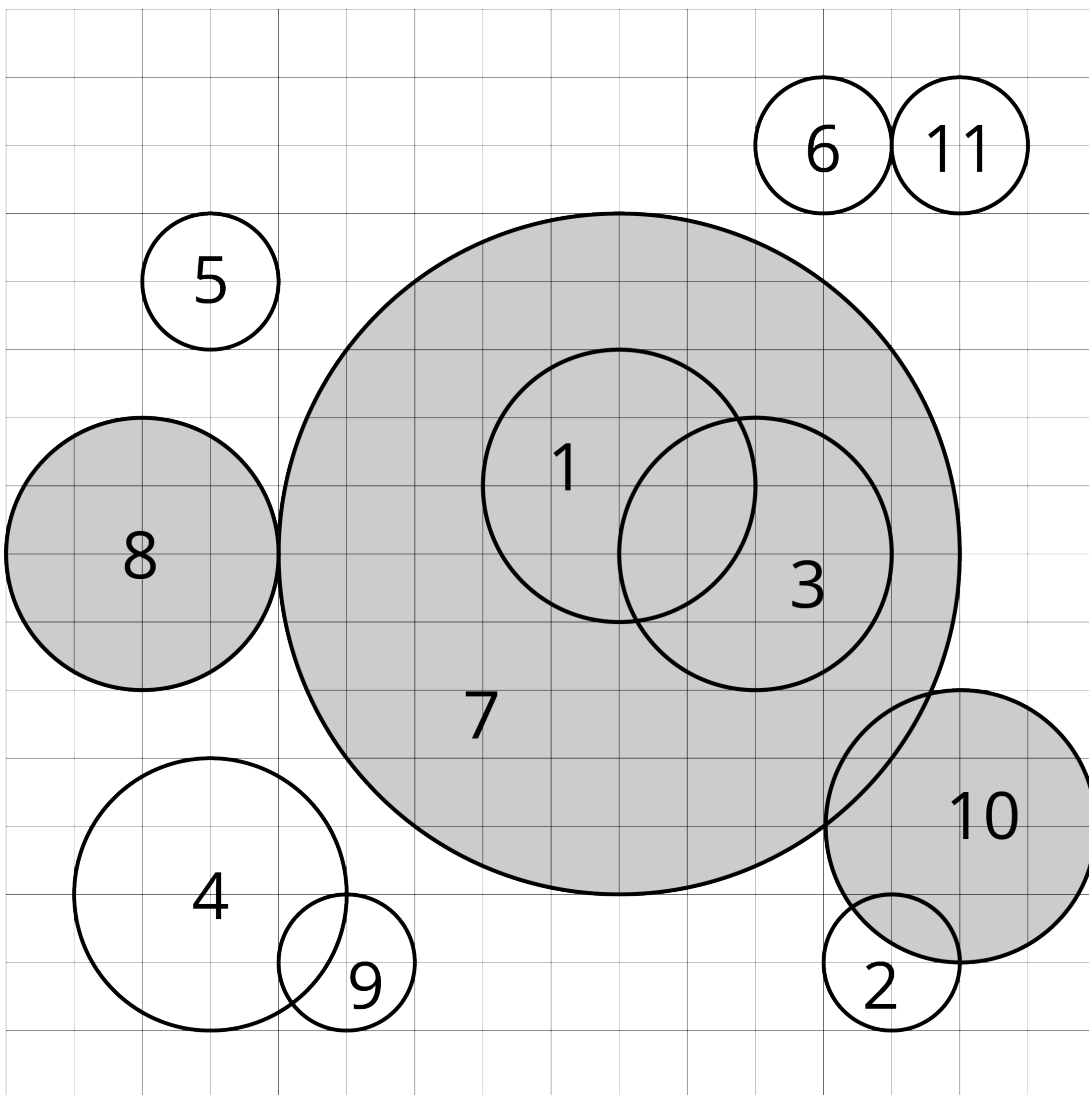
Во втором примере, если выбрать весь массив $(3, 7, 0, 4, 6, 1, 5, 2)$, наш герой не может изменить **XOR** подпоследовательности (который равен 0), независимо от того, какие два элемента меняются местами.

Задача C. Circle selection - Выбор кругов

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 1024 мегабайта

Дано n кругов c_1, c_2, \dots, c_n на координатной плоскости. Будем повторять следующие операции:

1. Выберем круг c_i с максимальным радиусом. Если есть несколько кругов с максимальным радиусом, выбирается круг с минимальным номером (наименьшим i).
2. Удалим круг c_i и все круги, которые имеют общие точки с c_i . Точка считается принадлежащей кругу, если она лежит внутри этого круга или на его границе.
3. Шаги 1 и 2 повторяются, пока есть хотя бы один круг.



Будем говорить, что круг c_i был удален кругом c_j , если c_j — круг, выбранный на шаге 1 описанного алгоритма, после чего на шаге 2 был удален круг c_i . Для каждого круга выясните, каким кругом он был удален.

Формат входных данных

Первая строка содержит целое число n , обозначающее количество кругов ($1 \leq n \leq 3 \cdot 10^5$). Каждая из следующих n строк содержит три целых числа x_i, y_i, r_i , задающих x-координату, y-координату и радиус круга c_i ($-10^9 \leq x_i, y_i \leq 10^9$, $1 \leq r_i \leq 10^9$).

Формат выходных данных

Выведите n целых чисел a_1, a_2, \dots, a_n на первой строке, для числа a_i должно выполняться условие, что круг c_i был удален кругом c_{a_i} .

Система оценки

17 баллов: $n \leq 5000$

14 баллов: $n \leq 3 \cdot 10^5$, $y_i = 0$ для всех кругов

16 баллов: $n \leq 3 \cdot 10^5$, каждый круг пересекается не более чем с одним другим кругом

23 балла: $n \leq 3 \cdot 10^5$, у всех кругов одинаковый радиус

19 баллов: $n \leq 10^5$

11 баллов: $n \leq 3 \cdot 10^5$

Пример

стандартный ввод	стандартный вывод
11	7 2 7 4 5 6 7 7 4 7 6
9 9 2	
13 2 1	
11 8 2	
3 3 2	
3 12 1	
12 14 1	
9 8 5	
2 8 2	
5 2 1	
14 4 2	
14 14 1	

Замечание

Рисунок в условии иллюстрирует первый пример.

Задача D. Ghiță, Lică Sămădăul и Buză Spartă

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

«Ghiță» очень любит программирование. Его любимые занятия — играть с перестановками и проводить время со своей женой Зинаидой. На свою 10-летнюю годовщину свадьбы Зинаида подарила ему очень красивую перестановку, ведь она знала, что это лучший подарок, который может получить «Ghiță». Пусть P_j — это j -элемент перестановки для каждого $1 \leq j \leq N$.

«Ghiță» был так рад такому подарку, что начал вычислять значение Q_i для каждого i что $1 \leq i \leq N$. Q_i — это число возрастающих подпоследовательностей на префиксе длины i в перестановке.

Более формально для каждого $1 \leq i \leq N$, Q_i равняется числу последовательностей j_1, j_2, \dots, j_k , что $1 \leq j_1 < j_2 < \dots < j_{k-1} < j_k \leq i$ и $P_{j_1} < P_{j_2} < \dots < P_{j_k}$.

Q конечно не перестановка, но тоже крутая штука, поэтому «Ghiță» положил её рядом с P .

Все было нормально, пока не появился «Lică Sămădăul». Он хотел использовать систему эпиднадзора «Ghiță» для аморальных целей, а «Ghiță», будучи честным человеком, не помог ему. Разгневанный ответом «Ghiță», «Lică Sămădăul» нанял «Buză Spartă», чтобы помочь ему украсть самые ценные активы «Ghiță»: его перестановку и его жену. И так он и сделал.

На следующий день «Ghiță» выяснил, что P отсутствует, и теперь единственная возможность для «Ghiță» восстановить перестановку — это использовать массив Q , который у него есть. Как вы уже догадались, ваша задача — помочь «Ghiță» восстановить перестановку P , используя массив Q .

Формат входных данных

В первой строке ввода записано одно число N ($1 \leq N \leq 70\,000$).

Во второй строке через пробел записаны N целых чисел Q_1, Q_2, \dots, Q_N .

Размер входного файла не превышает **115 МБ**.

Мы советуем вам самостоятельно проверять время работы и использование памяти читающей части вашей программы, чтобы убедиться, что возможная неэффективность вашей программы не связана с этой частью.

Формат выходных данных

В первой и единственной строке выведите P — украденную перестановку.

Гарантируется, что существует ровно один возможный ответ (только один P имеет заданный Q).

Система оценки

№	Баллы	Ограничения	T = длине Q_i	Размер входных данных
0	0	Тесты из условия	—	—
1	10	$N \leq 9$	—	—
2	15	$N \leq 400$	$T \leq 18$	—
3	18	$N \leq 700$	—	—
4	17	$N \leq 40\,000$	$T \leq 171$	4.5 МБ
5	11	$N \leq 70\,000$	$T \leq 258$	10 МБ
6	7	$N \leq 70\,000$	$T \leq 314$	16 МБ
7	16	$N \leq 70\,000$	—	85 МБ
8	6	$N \leq 70\,000$	—	115 МБ

Примеры

стандартный ввод	стандартный вывод
4 1 2 5 6	3 2 4 1
6 1 3 5 9 11 21	1 6 3 4 2 5

Замечание

В первом примере $N = 4$ и $P = \{3, 2, 4, 1\}$

$Q_1 = 1$, так как $\{3\}$ — единственная возрастающая подпоследовательность $\{3\}$

$Q_2 = 2$, потому что $\{3\}$ и $\{2\}$ — единственные возрастающие подпоследовательности $\{3, 2\}$

$Q_3 = 5$, потому что $\{3\}, \{3, 4\}, \{2\}, \{2, 4\}, \{4\}$ — единственные возрастающие подпоследовательности $\{3, 2, 4\}$

$Q_4 = 6$, потому что $\{3\}, \{3, 4\}, \{2\}, \{2, 4\}, \{4\}, \{1\}$ — единственные возрастающие подпоследовательности $\{3, 2, 4, 1\}$.