

## Задача 1. Посадка в самолет

Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

В самолетах авиакомпании Битавиа кресла расположены в  $n$  рядов, при этом в каждом ряду по шесть мест, между третьим и четвертым местом находится проход. Некоторые пассажиры регистрируются заранее онлайн, другие пассажиры регистрируются на стойке регистрации в аэропорту.

При онлайн-регистрации пассажир может выбрать любое место и не может его затем менять. Например, при  $n = 6$  рассадка в самолете после онлайн-регистрации может выглядеть так (крестиками отмечены занятые места):

	1	2	3	4	5	6
1	✗					
2						
3					✗	
4	✗					
5						
6			✗	✗		

На стойку регистрации придут  $m$  пассажиров. По правилам Битавиа нужно рассадить их в самолете таким образом, чтобы итоговая рассадка в самолете была симметрична относительно прохода. То есть, если в некотором ряду на первом кресле сидит пассажир, то в том же ряду на шестом кресле тоже должен сидеть пассажир. То же самое справедливо для второго и пятого, третьего и четвертого кресел, соответственно. При этом пересаживать пассажиров, прошедших онлайн-регистрацию нельзя. В исходную рассадку, показанную на рисунке выше, можно добавить семь пассажиров, удовлетворив условие симметрии, например, следующим образом:

	1	2	3	4	5	6
1	✗					✗
2	✗					✗
3		✗			✗	
4	✗					✗
5			✗	✗		
6			✗	✗		

Вам дана рассадка пассажиров после онлайн-регистрации. Требуется рассадить  $m$  пассажиров так, чтобы итоговая рассадка в самолете была симметрична относительно прохода, или определить, что это невозможно.

### Формат входных данных

В первой строке содержатся два целых числа  $n$  и  $m$  — количество рядов в самолете и количество пассажиров, которые придут на стойку регистрации ( $1 \leq n \leq 1000$ ,  $0 \leq m \leq 6000$ ).

В следующих  $n$  строках задана изначальная рассадка в самолете после онлайн-регистрации. В каждой строке содержится по шесть символов, при этом  $i$ -й символ  $j$ -й строки равен «X» (заглавная английская X), если  $i$ -е место в  $j$ -м ряду уже занято и «.» (точка) иначе.

### Формат выходных данных

Если искомой рассадки не существует, выведите «Impossible».

Иначе выведите  $n$  строк по шесть символов — итоговую рассадку в самолете. При этом  $i$ -й символ  $j$ -й строки должен быть равен «X», если место занято, и «.» (точка), если свободно. Если существует несколько решений, разрешается вывести любое.

### Система оценки

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

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	15	$m = 0$		первая ошибка
2	16	Изначально в самолете все места свободны		первая ошибка
3	17	$m = 1$		первая ошибка
4	18	Изначально в самолете занято ровно одно место		первая ошибка
5	34	нет	1–4	первая ошибка

### Примеры

стандартный ввод	стандартный вывод
1 0 X.XX.X	X.XX.X
2 1 X.XX.X ..X...	X.XX.X ..XX..
3 2 X.XX.X ..... X..X.X	Impossible
1 103 .X.XXX	Impossible
6 7 X..... ..... ....X. X..... ..... ..XX..	X....X X....X .X..X. X....X ..XX.. ..XX..

### Замечание

Выше приведены пять примеров входных данных.

- 1) В первом примере  $m = 0$ , а рассадка в самолете симметрична, поэтому итоговая рассадка совпадает с исходной.

- 2) Во втором примере есть только один способ рассадить пассажиров симметрично.
- 3) В третьем примере существовало бы решение, при  $m = 1$ , но при  $m = 2$  не существует способа рассадить всех пассажиров симметрично.
- 4) В четвертом примере требуется рассадить больше пассажиров чем свободных мест в самолете.
- 5) Пятый пример соответствует ситуации, рассмотренной на рисунках в тексте условия. В этом примере существует несколько решений, приведено одно из них.

## Задача 2. Битоническая последовательность

Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Последовательность  $[b_1, b_2, \dots, b_k]$  называется битонической, если выполнены неравенства  $b_1 < b_2 < \dots < b_i > \dots > b_k$  для некоторого  $1 \leq i \leq k$ .

Например, последовательности  $[1]$ ,  $[1, 2, 3, 2]$ ,  $[1, 4, 10]$ ,  $[3, 2]$  являются битоническими, а последовательности  $[1, 1]$ ,  $[2, 1, 3]$  — нет.

Задана последовательность  $[a_1, a_2, \dots, a_n]$ . Требуется количество пар  $(l, r)$  таких, что  $1 \leq l \leq r \leq n$  и последовательность  $[a_l, a_{l+1}, \dots, a_r]$  является битонической.

### Формат входных данных

Первая строка ввода содержит число  $n$  ( $1 \leq n \leq 300\,000$ ).

Вторая строка ввода содержит  $n$  целых чисел:  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq n$ ).

### Формат выходных данных

Выведите одно число — количество пар  $(l, r)$ , таких, что  $1 \leq l \leq r \leq n$  и последовательность  $[a_l, a_{l+1}, \dots, a_r]$  является битонической.

### Система оценки

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

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	27	$n \leq 500$		первая ошибка
2	14	$n \leq 5000$	1	первая ошибка
3	20	все числа $a_i$ различны		первая ошибка
4	39	—	1–3	первая ошибка

### Примеры

стандартный ввод	стандартный вывод
5 1 1 2 3 1	11
3 1 1 1	3

### Замечание

В первом примере подходят следующие пары:

- $(1, 1)$ , последовательность  $[1]$
- $(2, 2)$ , последовательность  $[1]$
- $(2, 3)$ , последовательность  $[1, 2]$
- $(2, 4)$ , последовательность  $[1, 2, 3]$
- $(2, 5)$ , последовательность  $[1, 2, 3, 1]$
- $(3, 3)$ , последовательность  $[2]$
- $(3, 4)$ , последовательность  $[2, 3]$
- $(3, 5)$ , последовательность  $[2, 3, 1]$
- $(4, 4)$ , последовательность  $[3]$
- $(4, 5)$ , последовательность  $[3, 1]$
- $(5, 5)$ , последовательность  $[1]$

## Задача 3. Игра с таблицей

Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Дана таблица  $A$  из  $h$  строк и  $w$  столбцов, в каждой ячейке которой записано целое число. Строки пронумерованы от 1 до  $h$  сверху вниз, столбцы пронумерованы от 1 до  $w$  слева направо.

Разрешается применять к этой таблице следующие операции:

- выбрать столбец таблицы и удалить его (столбцы слева и справа от него становятся соседними);
- выбрать строку таблицы и удалить ее (строки сверху и снизу от нее становятся соседними).

Эти операции разрешается применить произвольное число раз в любом порядке.

Определите, возможно ли при помощи этих операций получить из исходной таблицу с суммой чисел, равной заданному числу  $s$ , и если да, то какие операции и в каком порядке необходимо применить.

### Формат входных данных

Первая строка ввода содержит числа  $h$  и  $w$  — размеры таблицы ( $1 \leq h, w \leq 15$ ).

Каждая из следующих  $h$  строк содержит по  $w$  целых чисел — таблицу  $A$  ( $0 \leq A_{i,j} \leq 10^9$ ).

В последней строке ввода находится число  $s$  — необходимая сумма ( $1 \leq s \leq 10^{18}$ ).

### Формат выходных данных

Если получить таблицу с суммой чисел  $s$  из исходной невозможно, выведите строку «NO».

Иначе:

- В первой строке выведите строку «YES».
- Во второй строке выведите единственное число  $k$  — количество операций с таблицей, которые необходимо применить, чтобы получить из неё таблицу с суммой чисел  $s$ .
- В каждой из следующих  $k$  строк выведите по два целых числа  $t_j, i_j$ , где  $t_j = 1$ , если очередная операция производится со строкой, и  $t_j = 2$ , если она производится со столбцом таблицы. Число  $i_j$  должно быть равно номеру строки или столбца, соответственно, в *исходной* нумерации, с которой эта операция производится.

### Система оценки

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

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	17	$h = 1$		первая ошибка
2	6	сумма чисел в $i$ -й строке не превосходит $i$		первая ошибка
3	10	$h \leq 3$	1	первая ошибка
4	13	$h, w \leq 10$		первая ошибка
5	13	$h, w \leq 12$	4	первая ошибка
6	12	$a_{i,j} \leq 6$		первая ошибка
7	29		1–6	первая ошибка

## Примеры

стандартный ввод	стандартный вывод
3 3 1 2 3 2 3 1 3 1 2 8	YES 2 1 3 2 3
2 3 2 2 2 2 2 2 5	NO
5 5 1 2 1 4 5 2 5 4 1 2 4 2 4 3 1 5 5 3 2 4 1 2 4 5 2 34	YES 3 1 4 1 5 2 1

## Замечание

В первом примере изначально дана следующая таблица:

1	2	3
2	3	1
3	1	2

Удалив третью строку и столбец получим таблицу с суммой чисел 8:

1	2	3
2	3	1
3	1	2

 $\rightarrow$ 

1	2	3
2	3	1

 $\rightarrow$ 

1	2
2	3

Во втором примере можно показать, что разрешенными операциями невозможно получить таблицу с суммой чисел 5 из исходной.

В третьем примере изначально дана таблица:

1	2	1	4	5
2	5	4	1	2
4	2	4	3	1
5	5	3	2	4
1	2	4	5	2

Удалив последние две строки и первый столбец, получим таблицу с суммой чисел 34:

1	2	1	4	5
2	5	4	1	2
4	2	4	3	1
5	5	3	2	4
1	2	4	5	2

 $\rightarrow$ 

1	2	1	4	5
2	5	4	1	2
4	2	4	3	1
5	5	3	2	4

 $\rightarrow$ 

1	2	1	4	5
2	5	4	1	2
4	2	4	3	1

 $\rightarrow$ 

2	1	4	5
5	4	1	2
2	4	3	1

## Задача 4. Выбор столицы

Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

Дано неориентированное дерево — связный граф из  $n$  вершин без циклов, и число  $k$ . Зафиксируем некоторую вершину  $s$  дерева и назовем ее столицей.

Ориентируем ребра дерева в направлении от столицы. Иными словами, ориентируем ребро  $(u, v)$  в направлении  $u \rightarrow v$ , если при подвешивании дерева за вершину  $s$  вершина  $u$  является родителем вершины  $v$ . Заметим, что при таком ориентировании ребер каждая вершина достижима из столицы.

Определим расстояние до вершины  $v$  графа как минимальное количество ребер на пути из  $s$  в  $v$ . Назовем *доступностью* вершины  $s$  максимальное из расстояний до всех вершин.

Разрешается добавить в дерево не более  $k$  дополнительных ориентированных ребер.

Для каждой вершины  $s$  дерева определите, какой минимальной *доступности* можно достичь, если выбрать вершину  $s$  в качестве столицы.

Обратите внимание, что в некоторых подзадачах требуется вывести ответ только для первой вершины.

### Формат входных данных

Первая строка содержит три целых числа  $n$ ,  $k$  и  $t$  ( $2 \leq n \leq 2 \cdot 10^5$ ,  $1 \leq k \leq n - 1$ ,  $n \cdot k \leq 2 \cdot 10^5$ ,  $0 \leq t \leq 1$ ) — количество вершин дерева, ограничение на максимальное количество добавленных ребер и число  $t$ , равное 0, если нужно вывести ответ только для вершины с номером 1, и равное 1 иначе.

Каждая из следующих  $n - 1$  строк содержит два целых числа  $u_i, v_i$  ( $1 \leq u_i, v_i \leq n$ ) — ребра дерева.

Гарантируется, что заданные ребра образуют дерево.

### Формат выходных данных

В случае, если  $t = 0$ , выведите единственное целое число: минимальную *доступность*, которую можно достичь, выбрав вершину с номером 1 в качестве столицы, и добавив не более  $k$  дополнительных ориентированных ребер.

В случае, если  $t = 1$ , выведите  $n$  чисел:  $i$ -е число равняется минимальной *доступности*, которую можно достичь, выбрав вершину  $i$  в качестве столицы, и добавив не более  $k$  дополнительных ориентированных ребер.

### Система оценки

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

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	5	$u_i = i, v_i = i + 1, t = 0$		первая ошибка
2	5	$k = 1, n \leq 2000, t = 0$		первая ошибка
3	10	$k = 1, t = 0$	2	первая ошибка
4	5	$u_i = i, v_i = i + 1$	1	первая ошибка
5	5	$n \leq 16$		первая ошибка
6	10	$n \leq 50$	5	первая ошибка
7	10	$n \leq 400$	5, 6	первая ошибка
8	10	$n \leq 2000$	5, 6, 7	первая ошибка
9	25	$n \cdot k \leq 50000$	2, 5, 6, 7, 8	первая ошибка
10	15	нет	1–9	первая ошибка

## Примеры

стандартный ввод	стандартный вывод
5 2 1 1 2 1 3 2 4 2 5	1 1 2 2 2
3 1 0 1 2 2 3	1

## Замечание

На рисунке приведены иллюстрации к первому примеру. Пунктирными линиями обозначены добавленные ребра. Для вершин 1 и 2 минимальная *доступность* равняется 1, а для вершин 3, 4 и 5 минимальная *доступность* равняется 2.

