

Содержание

Прошлое и динамика	2
Задача A1. Любители Кошек [0.5 sec, 256 mb]	2
Задача A2. Лифт [0.5 sec, 256 mb]	3
Задача A3. Гвоздики [0.5 sec, 256 mb]	4
Задача A4. Свертка [0.5 sec, 256 mb]	5
Деревья	6
Задача B1. Простейшее BST: upper_bound [0.5 sec, 256 mb]	6
Задача B2. К-ый максимум [0.5 sec, 256 mb]	7
Задача B3. Неявный Ключ [0.5 sec, 256 mb]	8
Жадность и динамика	9
Задача C1. Число [0.5 sec, 256 mb]	9
Задача C2. Белоснежка и n гномов [0.5 sec, 256 mb]	10
Задача C3. Разложение на кубы [0.5 sec, 256 mb]	11
Задача C4. Сильные интеллектом [0.5 sec, 256 mb]	12
Строки и динамика	13
Задача D1. К-я строка [0.5 sec, 256 mb]	13
Задача D2. Телефонные номера [0.5 sec, 256 mb]	14
Задача D3. Сравнения подстрок [0.5 sec, 256 mb]	16
Задача D4. Свобода выбора [0.5 sec, 256 mb]	17
Дерево отрезков и сканирующая прямая	18
Задача E1. Сумма [0.5 sec, 256 mb]	18
Задача E2. Перестановки [0.5 sec, 256 mb]	19
Паросочетания и потоки	20
Задача F1. Pairs. Паросочетание [0.5 sec, 256 mb]	20
Задача F2. Замощение доминошками [0.5 sec, 256 mb]	21

Задача F3. Такси [0.5 sec, 256 mb]	22
Задача F4. Улиточки [0.5 sec, 256 mb]	23
Задача F5. Molecule. Химия!!! [0.5 sec, 256 mb]	24

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же stdin), вывести ответ нужно в **стандартный поток вывода** (он же stdout).

Прошлое и динамика

Задача A1. Любители Кошек [0.5 sec, 256 mb]

В университетском клубе любителей кошек зарегистрировано n членов. Естественно, что некоторые из членов клуба знакомы друг с другом. Нужно сосчитать, сколькими способами можно выбрать из них троих, которые могли бы свободно общаться (то есть, любые два из которых знакомы между собой).

Формат входных данных

В первой строке входного файла заданы числа n и m ($1 \leq n \leq 1000$, $1 \leq m \leq 30000$), где m обозначает общее число знакомств. В последующих m строках идут пары чисел a_i b_i , обозначающие, что a_i знаком с b_i . Информация об одном знакомстве может быть записана несколько раз, причем даже в разном порядке (как (x, y) , так и (y, x)).

Формат выходных данных

В выходной файл необходимо вывести количество способов выбрать троих попарно знакомых друг с другом людей из клуба.

Пример

stdin	stdout
3 3 1 2 2 3 3 1	1

Задача А2. Лифт [0.5 sec, 256 mb]

Высокое здание, состоящее из N этажей, оснащено только одним лифтом. Парковка находится ниже фундамента здания, что соответствует одному этажу ниже первого. Этажи пронумерованы от 1 до N снизу вверх. Про каждый этаж известно количество человек, желающих спуститься на лифте на парковку. Пусть для i -го этажа эта величина равна A_i . Известно, что лифт не может перевозить более C человек одновременно, а также то, что на преодоление расстояния в один этаж (не важно вверх или вниз) ему требуется P секунд. Какое наибольшее количество человек лифт может перевезти на парковку за T секунд, если изначально он находится на уровне парковки?

Формат входных данных

В первой строке входного файла содержатся целые числа N, C, P, T ($1 \leq N \leq 100$, $1 \leq C \leq 10^9$, $1 \leq P \leq 10^9$, $1 \leq T \leq 10^9$). Вторая строка содержит последовательность N целых чисел A_1, A_2, \dots, A_N ($0 \leq A_i \leq 10^9$). Сумма всех значений последовательности не превосходит 10^9 .

Формат выходных данных

Выведите наибольшее количество человек, которое лифт успеет перевезти на парковку.

Пример

stdin	stdout
4 5 2 15 0 1 2 3	3
4 5 2 18 0 1 2 3	5
3 2 1 9 1 1 1	3

Задача А3. Гвоздики [0.5 sec, 256 mb]

На прямой дощечке вбиты гвоздики. Любые два гвоздика можно соединить ниточкой. Требуется соединить какие-то пары гвоздиков ниточками так, чтобы к каждому гвоздику была привязана хотя бы одна ниточка, а суммарная длина всех ниточек была минимальна.

Формат входных данных

В первой строке входного файла записано число N — количество гвоздиков ($2 \leq N \leq 100$). В следующей строке записано N чисел — координаты всех гвоздиков (неотрицательные целые числа, не превосходящие 10 000).

Формат выходных данных

В выходной файл нужно вывести единственное число — минимальную суммарную длину всех ниточек.

Пример

stdin	stdout
5 4 10 0 12 2	6

Задача А4. Свертка [0.5 сек, 256 mb]

Петя хочет сократить запись последовательности, состоящей из заглавных латинских букв. Для этого он может свернуть ее повторяющиеся подпоследовательности. Например, последовательность АААААААААВАВАВССD может быть записана как 10(A)2(BA)B2(C)D.

Формальной определение свернутой последовательности и соответствующей ей операции развертки дается следующим образом:

- Последовательность, которая содержит единственный символ от 'A' до 'Z' представляет из себя свернутую последовательность. При развертке такой последовательности получается она сама.
- Если S и Q — свернутые последовательности, то SQ также свернутая последовательность. Если при развертке строки S получается строка S' , а при развертке Q получается Q' , то при развертке SQ получается строка $S'Q'$.
- Если S — свернутая последовательность, то $X(S)$ также свернутая последовательность, где X это десятичное представление целого числа большего единицы. Если при развертке строки S получается строка S' , то при развертке $X(S)$ получается строка S' , повторенная X раз.

Петя хочет свернуть заданную последовательность таким образом, чтобы результат содержал наименьшее число символов.

Формат входных данных

Входной файл содержит непустую строку, состоящую из заглавных латинских букв. Длина строки не превышает 100 символов.

Формат выходных данных

В выходной файл выведите одну строку, содержащую наименьшую последовательность развертка которой даст строку, заданную во входном файле.

Если ответов несколько — выведите любой из них.

Пример

stdin	stdout
AAAAAAAAAABABABCCD	9(A)3(AB)CCD
NEERCYESYESYESNEERCYESYESYES	2(NEERC3(YES))

Деревья

Задача В1. Простейшее BST: upper_bound [0.5 sec, 256 mb]

В этой задаче вам нужно написать простейшее BST по явному ключу и отвечать им на запросы:

- + x – добавить в дерево x (если x уже есть, ничего не делать).
- > x – вернуть минимальный элемент больше x или 0, если таких нет.

Формат входных данных

В каждой строке содержится один запрос.

Все x целые от 1 до 10^9 , количество запросов от 1 до 300 000.

Гарантируется, что все x выбраны равномерным распределением.

Формат выходных данных

Для каждого запроса вида «> x » выведите в отдельной строке ответ.

Пример

stdin	stdout
+ 1	3
+ 3	3
+ 3	0
> 1	2
> 2	
> 3	
+ 2	
> 1	

Подсказка по решению

Случайные данные! Не нужно ничего специально балансировать.

```
struct node {
    node *l, *r;
    int x;
};
```

Задача В2. К-ый максимум [0.5 sec, 256 mb]

Напишите программу, реализующую структуру данных, позволяющую добавлять и удалять элементы, а также находить k -й максимум.

Формат входных данных

Первая строка входного файла содержит натуральное число n — количество команд ($n \leq 100\,000$). Последующие n строк содержат по одной команде каждая. Команда записывается в виде двух чисел c_i и k_i — тип и аргумент команды соответственно ($|k_i| \leq 10^9$). Поддерживаемые команды:

- $+1$ (или просто 1): Добавить элемент с ключом k_i .
- 0 : Найти и вывести k_i -й максимум.
- -1 : Удалить элемент с ключом k_i .

Гарантируется, что в процессе работы в структуре не требуется хранить элементы с равными ключами или удалять несуществующие элементы. Также гарантируется, что при запросе k_i -го максимума, он существует.

Формат выходных данных

Для каждой команды нулевого типа в выходной файл должна быть выведена строка, содержащая единственное число — k_i -й максимум.

Пример

stdin	stdout
11	7
+1 5	5
+1 3	3
+1 7	10
0 1	7
0 2	3
0 3	
-1 5	
+1 10	
0 1	
0 2	
0 3	

Подсказка по решению

Псевдокод от AVL: `def rotate(v): return node(v.l.x, v.l.l, node(v.x,v.l.r,v.r))`

Напишите сюда, пожалуйста, **AVL-дерево** или **Treap**.

Без полноценного дерева с перебалансировками не обойтись.

Дерево нужно по **явному** ключу. Чтобы находить k -й элемент, понимать спускаться влево, или вправо, достаточно хранить размеры поддеревьев (и пересчитывать).

Задача В3. Неявный Ключ [0.5 сек, 256 mb]

Научитесь быстро делать две операции с массивом:

- `add i x` — добавить после i -го элемента x ($0 \leq i \leq n$)
- `del i` — удалить i -й элемент ($1 \leq i \leq n$)

Формат входных данных

На первой строке n_0 и m ($1 \leq n_0, m \leq 10^5$) — длина исходного массива и количество запросов. На второй строке n_0 целых чисел от 0 до $10^9 - 1$ — исходный массив. Далее m строк, содержащие запросы. Гарантируется, что запросы корректны: например, если просят удалить i -й элемент, он точно есть.

Формат выходных данных

Выведите конечное состояние массива.

На первой строке количество элементов, на второй строке сам массив.

Примеры

stdin	stdout
3 4	3
1 2 3	9 2 8
del 3	
add 0 9	
add 3 8	
del 2	

Подсказка по решению

Сюда нужно писать AVL или Treap.

Жадность и динамика

Задача C1. Число [0.5 sec, 256 mb]

Вася написал на длинной полоске бумаги большое число и решил похвастаться своему старшему брату Пете этим достижением. Но только он вышел из комнаты, чтобы позвать брата, как его сестра Катя вбежала в комнату и разрежала полоску бумаги на несколько частей. В результате на каждой части оказалось одна или несколько идущих подряд цифр.

Теперь Вася не может вспомнить, какое именно число он написал. Только помнит, что оно было очень большое. Чтобы утешить младшего брата, Петя решил выяснить, какое максимальное число могло быть написано на полоске бумаги перед разрезанием. Помогите ему!

Формат входных данных

Входной файл содержит одну или более строк, каждая из которых содержит последовательность цифр. Количество строк во входном файле не превышает 100, каждая строка содержит от 1 до 100 цифр. Гарантируется, что хотя бы в одной строке первая цифра отлична от нуля.

Формат выходных данных

Выведите в выходной файл одну строку — максимальное число, которое могло быть написано на полоске перед разрезанием.

Примеры

stdin	stdout
2 20 004 66	66220004
3	3

Задача C2. Белоснежка и n гномов [0.5 sec, 256 mb]

«Ну не гномы, а наказание какое-то!», — подумала Белоснежка, в очередной раз пытаюсь уложить гномов спать. Одного уложишь — другой уже проснулся! И так всю ночь.

У Белоснежки n гномов, и все они очень разные. Она знает, что для того, чтобы уложить спать i -го гнома нужно a_i минут, и после этого он будет спать ровно b_i минут. Помогите Белоснежке узнать, может ли она получить хотя бы минутку отдыха, когда все гномы будут спать, и если да, то в каком порядке для этого нужно укладывать гномов спать.

Например, пусть есть всего два гнома, $a_1 = 1, b_1 = 10, a_2 = 10, b_2 = 20$. Если Белоснежка сначала начнет укладывать первого гнома, то потом ей потребуется целых 10 минут, чтобы уложить второго, а за это время проснется первый. Если же она начнет со второго гнома, то затем она успеет уложить первого и получит целых 9 минут отдыха.

Формат входных данных

Первая строка входного файла содержит число n ($1 \leq n \leq 10^5$), вторая строка содержит числа a_1, a_2, \dots, a_n , третья — числа b_1, b_2, \dots, b_n ($1 \leq a_i, b_i \leq 10^9$).

Формат выходных данных

Выведите в выходной файл n чисел — порядок, в котором нужно укладывать гномов спать. Если Белоснежке отдохнуть не удастся, выведите число -1 . Если решений несколько, выведите любое.

Пример

stdin	stdout
2 1 10 10 20	2 1
2 10 10 10 10	-1
3 1 4 1 5 3 4	2 1 3

Задача С3. Разложение на кубы [0.5 sec, 256 mb]

Дано целое число от n , представить его в виде суммы минимального числа кубов.

Формат входных данных

В этой задаче мультитест. Каждая строка входных данных содержит одно целое n от 1 до 50 000. Количество n в одном тесте от 1 до 10 000.

Формат выходных данных

Для каждого n на отдельной строке минимальное количество кубов в разложении.

Примеры

stdin	stdout
7	7
8	1
10	3
43	3

Замечание

$$7 = 1^3 + 1^3 + 1^3 + 1^3 + 1^3 + 1^3 + 1^3$$

$$8 = 2^3$$

$$10 = 2^3 + 1^3 + 1^3$$

$$43 = 2^3 + 2^3 + 3^3$$

Задача C4. Сильные интеллектом [0.5 сек, 256 mb]

Даны n людей. Каждый человек характеризуется силой a_i и интеллектом b_i .

Требуется выбрать как можно меньше людей, чтобы их суммарная сила была хотя бы A , а суммарный интеллект хотя бы B .

Формат входных данных

В первой строке дано три целых числа n , A и B ($1 \leq n \leq 1000, 1 \leq A \leq 10^9, 1 \leq B \leq 100$) — количество людей, сила и интеллект, которые требуется набрать. В каждой из следующих n строк заданы целые числа a_i и b_i ($1 \leq a_i \leq 10^9, 1 \leq b_i \leq 100$) — сила и интеллект человека с номером i .

$$1 \leq n \leq 1000, A \leq 10^9.$$

Формат выходных данных

В первой строке выведите k — количество людей в выбранном множестве. Во второй строке выведите k целых чисел — номера выбранных людей *в порядке возрастания*. Люди нумеруются от 1 до n в порядке входных данных. Если ответов несколько, выведите любой. Если невозможно выбрать такое множество людей, в единственной строке выведите -1 .

Примеры

stdin	stdout
2 5 5 1 1 2 2	-1
5 5 3 3 1 2 1 1 1 1 1 1 2	3 1 2 3
5 5 3 3 1 2 1 2 2 1 1 1 2	2 1 3

Подсказка по решению

Проблемы с памятью? Правильное решение не требует упихивания.

Строки и динамика

Задача D1. К-я строка [0.5 sec, 256 mb]

Реализуйте структуру данных, которая поддерживает следующие операции:

- добавить в словарь строку S ;
- найти в словаре k -ю строку в лексикографическом порядке.

Изначально словарь пуст.

Формат входных данных

Первая строка входного файла содержит натуральное число N — количество команд ($N \leq 10^5$). Последующие N строк содержат по одной команде каждая.

Команда записывается либо в виде числа k , либо в виде строки S , которая может состоять только из строчных латинских букв. При запросе k -й строки гарантируется, что она существует. Гарантируется, что сумма длин всех добавляемых строк не превышает 10^5 .

Формат выходных данных

Для каждого числового запроса k выходной файл должен содержать k -ю в лексикографическом порядке строчку из словаря на момент запроса. Гарантируется, что суммарная длина строк в выходном файле не превышает 10^5 .

Примеры

stdin	stdout
7	tolstoy
pushkin	gogol
lermontov	
tolstoy	
gogol	
gorkiy	
5	
1	

Подсказка по решению

Бор.

Задача D2. Телефонные номера [0.5 sec, 256 mb]

Задача старая. Ниже оригинальная легенда.

Вы встречаетесь с огромным количеством телефонных номеров, которые со временем становятся всё длиннее и длиннее. Вам приходится запоминать эти номера. Одним из простых способов запоминания является сопоставление букв каждой цифре, как показано на следующем рисунке:

1 ij	2 abc	3 def
4 gh	5 kl	6 mn
7 prs	8 tuv	9 wxу
0 oqz		

Таким образом, каждому слову или группе слов может быть сопоставлен уникальный номер, так что можно запоминать слова вместо телефонных номеров. Есть особый шарм в том, чтобы найти простую взаимосвязь между словом, используемым для запоминания телефонного номера, и владельцем этого номера. Так, телефонный номер 941837296 вашего друга, играющего в шахматы, может быть прочитан как WHITEPAWN (белая пешка), а номер 2855304 Вашего любимого учителя может быть прочитан как BULLDOG (бульдог). Напишите программу, находящую самую короткую последовательность слов (имеющую наименьшее количество слов), которая соответствует заданному номеру телефона и заданному списку слов. Соответствие описано на рисунке выше.

Формат входных данных

Ввод состоит из нескольких тестов. Первая строка каждого теста содержит номер телефона, к которому нужно подобрать мнемонику. Номер состоит не более чем из 100 цифр. Вторая строка содержит общее количество слов в словаре (максимум 50 000). Каждая из оставшихся строк содержит одно слово, состоящее не более чем из 50 строчных латинских букв. Общий размер ввода не превосходит 300 килобайт. Последняя строка ввода содержит число -1 .

Формат выходных данных

Каждая строка вывода должна содержать кратчайшую последовательность слов, найденную вашей программой. Слова должны быть разделены одиночными пробелами. Если решения нет, выведите «No solution.». Если существует несколько решений, имеющих одинаковое количество слов, можете выбрать любое из них.

Примеры

stdin	stdout
7325189087	reality our
5	No solution.
it	
your	
reality	
real	
our	
4294967296	
5	
it	
your	
reality	
real	
our	
-1	

Подсказка по решению

Бор. И динамика.

Задача D3. Сравнения подстрок [0.5 sec, 256 mb]

Дана строка. Нужно уметь отвечать на запросы вида: равны ли подстроки $[a..b]$ и $[c..d]$.

Формат входных данных

Сперва строка S (не более 10^5 строчных латинских букв). Далее число M — количество запросов.

В следующих M строках запросы a,b,c,d . $0 \leq M \leq 10^5, 1 \leq a \leq b \leq |S|, 1 \leq c \leq d \leq |S|$

Формат выходных данных

M строк. Выведите Yes, если подстроки совпадают, и No иначе.

Пример

stdin	stdout
trololo	Yes
3	Yes
1 7 1 7	No
3 5 5 7	
1 1 1 5	

Подсказка по решению

Просто хеши.

Задача D4. Свобода выбора [0.5 sec, 256 mb]

Даны две строки, состоящих из заглавных латинских букв. Нужно найти их наибольшую общую подстроку. Полное условие можно посмотреть на [тимусе](#).

Формат входных данных

На первой строке число n ($1 \leq n \leq 10^5$).

На второй и третьей строках находятся по n заглавных английских букв.

Формат выходных данных

Максимальную по длине общую подстроку. Если оптимальных ответов несколько, выведите любой.

Примеры

stdin	stdout
28 VOTEFORTHEGREATALBANIAFORYOU CHOOSETHEGREATALBANIANFUTURE	THEGREATALBANIA

Подсказка по решению

Примениить хеши. Разбирали на практике.

19-й – антихеш тест

Дерево отрезков и сканирующая прямая

Задача E1. Сумма [0.5 sec, 256 mb]

Дан массив из N элементов, нужно научиться находить сумму чисел на отрезке.

Формат входных данных

Первая строка содержит два целых числа N и K — число чисел в массиве и количество запросов. ($1 \leq N \leq 100\,000$), ($0 \leq K \leq 100\,000$). Следующие K строк содержат запросы

- “A i x ” — присвоить i -му элементу массива значение x ($1 \leq i \leq n$, $0 \leq x \leq 10^9$)
- “Q l r ” — найти сумму чисел в массиве на позициях от l до r . ($1 \leq l \leq r \leq n$)

Изначально в массиве живут нули.

Формат выходных данных

На каждый запрос вида Q l r нужно вывести единственное число — сумму на отрезке.

Примеры

stdin	stdout
5 9	0
A 2 2	2
A 3 1	1
A 4 2	2
Q 1 1	0
Q 2 2	5
Q 3 3	
Q 4 4	
Q 5 5	
Q 1 5	

Замечание

Обыкновенное дерево отрезков.

Задача E2. Перестановки [0.5 sec, 256 mb]

Вася выписал на доске в каком-то порядке все числа от 1 по N , каждое число ровно по одному разу. Количество чисел оказалось довольно большим, поэтому Вася не может окинуть взглядом все числа. Однако ему надо всё-таки представлять эту последовательность, поэтому он написал программу, которая отвечает на вопрос — сколько среди чисел, стоящих на позициях с x по y , по величине лежат в интервале от k до l . Сделайте то же самое.

Формат входных данных

В первой строке лежит два натуральных числа — $1 \leq N \leq 100\,000$ — количество чисел, которые выписал Вася и $1 \leq M \leq 100\,000$ — количество вопросов, которые Вася хочет задать программе. Во второй строке дано N чисел — последовательность чисел, выписанных Васей. Далее в M строках находятся описания вопросов. Каждая строка содержит четыре целых числа $1 \leq x \leq y \leq N$ и $1 \leq k \leq l \leq N$.

Формат выходных данных

Выведите M строк, каждая должна содержать единственное число — ответ на Васин вопрос.

Пример

stdin	stdout
4 2	1
1 2 3 4	3
1 2 2 3	
1 3 1 3	

Паросочетания и потоки

Задача F1. Pairs. Паросочетание [0.5 sec, 256 mb]

Двудольным графом называется граф (V, E) , $E \subset V \times V$ такой, что его множество вершин V можно разбить на два подмножества A и B , для которых $\forall (e_1, e_2) \in E \ e_1 \in A, e_2 \in B$ и $A, B \subset E, A \cap B = \emptyset$.

Паросочетанием в двудольном графе называется любой его набор несмежных ребер, то есть такой набор $S \subset E$, что для любых двух ребер $e_1 = (u_1, v_1), e_2 = (u_2, v_2)$ из S выполнено $u_1 \neq u_2$ и $v_1 \neq v_2$.

Ваша задача — найти максимальное паросочетание в двудольном графе, то есть паросочетание с максимально возможным числом ребер.

Формат входных данных

В первой строке записаны два целых числа n и m ($1 \leq n, m \leq 250$) — число вершин в A и число вершин в B .

Далее следуют n строк с описаниями ребер. i -я вершина из A описана в $i + 1$ -й строке файла. Каждая из этих строк содержит номера вершин из B , соединенных с i -й вершиной A . Вершины в A и B нумеруются независимо (с единицы). Список завершается числом 0.

Формат выходных данных

Первая строка выходного файла должна содержать одно целое число l — количество ребер в максимальном паросочетании. Далее должны следовать l строк, в каждой из которых должны быть два целых числа u_j и v_j — концы ребер паросочетания в A и B , соответственно.

Пример

stdin	stdout
2 2	2
1 2 0	1 1
2 0	2 2

Задача F2. Замощение доминошками [0.5 сек, 256 mb]

Дано игровое поле размера $n \times m$, некоторые клетки которого уже замощены. Замостить свободные соседние клетки поля доминошкой размера 1×2 стоит a условных единиц, а замостить свободную клетку поля квадратиком размера 1×1 — b условных единиц.

Определите, какая минимальная сумма денег нужна, чтобы замостить всё поле.

Формат входных данных

Первая строка входного файла содержит 4 целых числа n, m, a, b ($1 \leq n, m \leq 100, |a| \leq 1000, |b| \leq 1000$). Каждая из последующих n строк содержит по m символов: символ "." (точка) обозначает занятую клетку поля, а символ "*" (звёздочка) — свободную.

Формат выходных данных

В выходной файл выведите одно число — минимальную сумму денег, имея которую можно замостить свободные клетки поля (и только их).

Пример

stdin	stdout
2 3 3 2 .** .*.	5

Задача F3. Такси [0.5 sec, 256 mb]

Управлять службой такси — совсем не простое дело. Помимо естественной необходимости централизованного управления машинами для того, чтобы обслуживать заказы по мере их поступления и как можно быстрее, нужно также планировать поездки для обслуживания тех клиентов, которые сделали заказы заранее.

В вашем распоряжении находится список заказов такси на следующий день. Вам необходимо минимизировать число машин такси, необходимых чтобы выполнить все заказы.

Для простоты будем считать, что план города представляет собой квадратную решетку. Адрес в городе будем обозначать парой целых чисел: x -координатой и y -координатой. Время, необходимое для того, чтобы добраться из точки с адресом (a, b) в точку (c, d) , равно $|a - c| + |b - d|$ минут. Машина такси может выполнить очередной заказ, либо если это первый ее заказ за день, либо она успевает приехать в начальную точку из предыдущей конечной хотя бы за минуту до указанного срока. Обратите внимание, что выполнение некоторых заказов может закончиться после полуночи.

Формат входных данных

В первой строке входного файла записано число заказов M ($0 < M < 500$). Последующие M строк описывают сами заказы, по одному в строке. Про каждый заказ указано время отправления в формате `hh:mm` (в интервале с `00:00` по `23:59`), координаты (a, b) точки отправления и координаты (c, d) точки назначения. Все координаты во входном файле неотрицательные и не превосходят 200. Заказы записаны упорядоченными по времени отправления.

Формат выходных данных

В выходной файл выведите единственное целое число — минимальное количество машин такси, необходимых для обслуживания всех заказов.

Пример

stdin	stdout
2 08:00 10 11 9 16 08:07 9 16 10 11	1
2 08:00 10 11 9 16 08:06 9 16 10 11	2

Задача F4. Улиточки [0.5 sec, 256 mb]

Две улиточки Маша и Петя сейчас находятся в на лужайке с абрикосами и хотят добраться до своего домика. Лужайки пронумерованы числами от 1 до n и соединены дорожками (может быть несколько дорожек соединяющих две лужайки, могут быть дорожки, соединяющие лужайку с собой же). В виду соображений гигиены, если по дорожке проползла улиточка, то вторая по той же дорожке уже ползти не может. Помогите Пете и Маше добраться до домика.

Формат входных данных

В первой строке файла записаны четыре целых числа — n , m , s и t (количество лужаек, количество дорог, номер лужайки с абрикосами и номер домика). В следующих m строках записаны пары чисел. Пара чисел (x, y) означает, что есть дорожка с лужайки x до лужайки y (из-за особенностей улиток и местности дорожки односторонние). Ограничения: $2 \leq n \leq 10^5$, $0 \leq m \leq 10^5$, $s \neq t$.

Формат выходных данных

Если существует решение, то выведите YES и на двух отдельных строчках сначала последовательность лужаек для Машеньки (дам нужно пропускать вперед), затем путь для Пети. Если решения не существует, выведите NO. Если решений несколько, выведите любое.

Пример

stdin	stdout
3 3 1 3	YES
1 2	1 3
1 3	1 2 3
2 3	

Замечание

Дан оргграф, найти два непересекающихся по ребрам пути из s в t , вывести вершины найденных путей.

Задача F5. Molecule. Химия!!! [0.5 sec, 256 mb]

Вася и Сережа играют в следующую игру. В некоторых клетках клетчатого листка Сережа рисует один из символов 'H', 'O', 'N' или 'C', после чего Вася должен провести между некоторыми находящимися в соседних клетках символами линии так, чтобы получилось корректное изображение химической молекулы. К сожалению, Сережа любит рисовать много символов, и Вася не может сразу определить, возможно ли вообще нарисовать линии нужным способом. Помогите ему написать программу, которая даст ответ на этот вопрос.

В этой задаче проведенные между символами химических элементов линии будем считать корректным изображением молекулы, если они удовлетворяют следующим условиям:

- каждая линия соединяет символы, нарисованные в соседних (по стороне) клетках,
- между каждой парой символов проведено не более одной линии,
- от каждого элемента отходит ровно столько линий, какова валентность этого элемента (1 для H, 2 для O, 3 для N, 4 для C),
- пустые клетки ни с чем не соединены, и
- **хотя бы в одной клетке** нарисован какой-то символ.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m ($1 \leq n, m \leq 50$) — размеры листочка, на котором рисует Сережа. Далее следуют n строк по m символов в каждой, задающих конфигурацию химических элементов, которую нарисовал Сережа; пустые клетки задаются символом '.'.

Формат выходных данных

В выходной файл выведите одно слово: 'Valid', если линии провести требуемым образом можно, и 'Invalid', если нельзя.

Пример

stdin	stdout
3 4 HOH. NCOH OO..	Valid
3 4 HOH. NCOH OONH	Invalid