

Открытая олимпиада школьников (информатика) 2024-25

Задания заключительного этапа для 9 класса

1. Кодирование информации. Системы счисления (1 балл)

[WOW!]

Известно, что запись числа в шестнадцатеричной системе состоит из идущих подряд последовательностей четырех цифр KXYZ, то есть выглядит как KXYZKXYZ...KXYZ₁₆ (K, X, Y, Z – цифры 16-ричной системы счисления). После перевода этой записи в двоичную систему счисления оказалось, что она состоит из двоичных чисел W, разделённых двумя идущими подряд нолями, то есть выглядит как W00...00W00W₂, причем чисел W сколько же, сколько последовательностей KXYZ в исходном числе (W также может содержать в своей записи 00). Определите количество различных вариантов последовательностей цифр KXYZ, для которых это возможно. Два варианта последовательностей будут считаться различными, если отличаются значения хотя бы одной цифры, например варианты 1234, 1134 и 1324 – это три различных варианта. В ответе укажите целое число.

2. Кодирование информации. Объем информации (3 балла)

[Алиса и Боб]

Алиса решила написать генератор простых текстов на английском языке. Для хранения текста в памяти она решила использовать неравномерное кодирование, основываясь на частоте букв в английском алфавите. Она воспользовалась простой таблицей:

Буква	E	T	A	O	I	N	S	H	R	D	L	C	U
Частотность, %	12,7	9,06	8,17	7,51	6,97	6,75	6,33	6,09	5,99	4,25	4,03	2,78	2,76
Буква	M	W	F	G	Y	P	B	V	K	X	J	Q	Z
Частотность, %	2,41	2,36	2,23	2,02	1,97	1,93	1,49	0,98	0,77	0,15	0,15	0,1	0,05

Кодирование Алисы построено следующим образом:

- В начале каждого кода символа находится специальный бит, обозначающий начало кода.
- Затем идёт бит, отвечающий за то, является ли буква заглавной.
- Кодом длиной i бит она кодирует 2^i букв с наибольшей популярностью, которым ещё не назначены более короткие коды. Например, буквы E и T получают однобитные коды, следующие 4 буквы – двухбитные коды, и так далее.

Её друг Боб справедливо заметил, что так как она генерирует простые тексты, в них используется не более 1500 различных слов (Боб также игнорирует регистр букв в слове) и предложил использовать минимальное, одинаковое для всех слов количество бит, а также для каждого слова записывать 1 бит метainформации – начинается ли слово с заглавной буквы. Определите, при какой минимальной средней длине слова способ Боба будет эффективнее способа Алисы (для хранения всего текста потребуется строго меньше бит). Средней длиной слова будем считать количество символов, поделенное на количество слов и округлённое до ближайшего целого числа. В ответ запишите одно число – искомую среднюю длину.

3. Основы логики (2 балла)

[Под знаком вопроса]

Дано логическое уравнение с тремя логическими переменными A, B и C:

$$(A \ ? \ B) \ ? \ (B \ ? \ C) \ ? \ (A \ ? \ C) = 0.$$

В данном уравнении на месте символа ? может стоять знак операции конъюнкции или дизъюнкции. Сколько существует способов заменить ? так, чтобы уравнение имело ровно 4 решения?

4. Алгоритмизация и программирование. Формальные исполнители (1 балл)

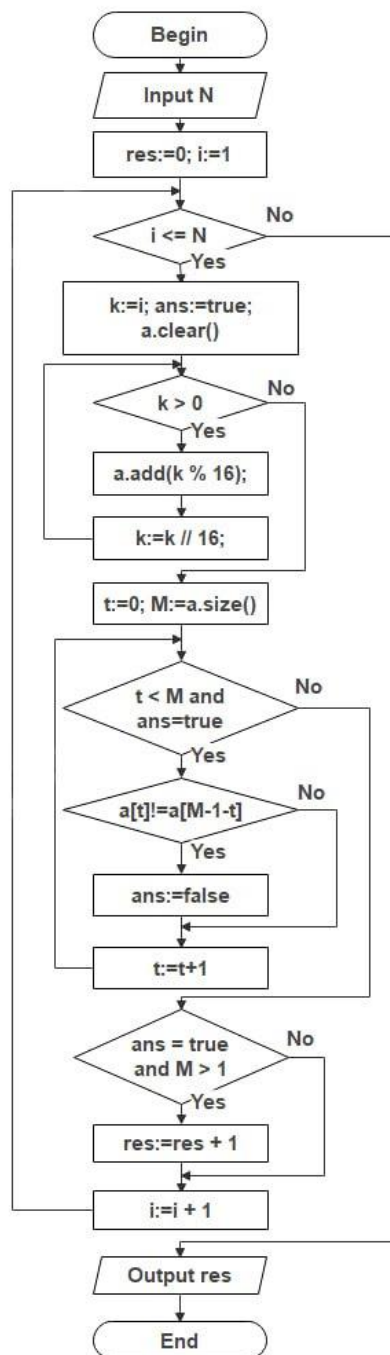
[Простые замены]

Дана последовательность нолей и единиц вида 00110011..., длиной 5000 символов. За один ход можно либо заменить 3 подряд идущих символа, начиная с текущего, на противоположные (0 на 1, 1 на 0), либо перейти к следующему символу. В случае если от некоторого символа до конца строки меньше 3 символов, замена производится для всех оставшихся до конца строки символов (например, строку 100 можно превратить в строку 111, сделав переход к следующему символу, и затем заменив 00 на 11). Для первого хода, текущим считается первый символ строки. Сколько раз понадобится совершить ход с заменой, чтобы превратить эту строку в состоящую только из единиц?

5. Алгоритмизация и программирование. Анализ алгоритма, заданного в виде блок-схемы (3 балла)

[Поиск мина и макс]

Дана блок-схема алгоритма:



Операция % обозначает взятие остатка от деления, операция // обозначает целочисленное деление. Операция **a.clear()** создаёт пустой список чисел, операция **a.size()** возвращает количество чисел в **a**, операция **a.add(x)** дописывает **x** в конец списка **a**. На вход было подано некоторое натуральное число **N**, после чего алгоритм вывел 278. Найдите минимальное и максимальное **N**, при которых это возможно. В ответ запишите через пробел два числа – минимальное и максимальное значения.

6. Телекоммуникационные технологии (2 балла)

[Последствия невнимательности]

Маска сети для IPv4 адресации – это 4-байтное число, которое делит IP адрес на адрес сети (первая часть) и адрес узла (вторая часть). У всех адресов одной IP-сети совпадают первые части и отличаются вторые. Для части IP адреса, соответствующей адресу сети, в маске сети содержатся двоичные единицы, а для части IP адреса, соответствующей адресу узла, в маске сети содержатся двоичные нули.

Недавно Олег узнал о масках подсети, однако, пока ему рассказывали, он был не слишком внимателен, и запомнил, что чтобы получить адрес сети, нужно применить побитовую операцию «исключающее или» к адресу узла и маске. В результате из адреса узла **X** и трёх различных корректных масок **A**, **B** и **C** Олег получил следующие «адреса сетей» – 40.102.219.169, 40.102.196.89 и 40.102.59.169. Определите адрес узла, который использовал Олег. В ответ запишите 4 числа без ведущих нулей, разделённых точками – адрес узла.

Пример записи ответа: 192.168.0.1

7. Технологии обработки информации в электронных таблицах, технологии сортировки и фильтрации данных (2 балла)

[Есть тут кто?]

В ячейки таблицы A1, A4, B1, B2, C1, D1 поместили некоторые формулы (см. изображение).

	A	B	C	D	E
1	=COUNTIF(B2:AD30; "=0")	=A2	=A3	=IF(AND((B\$1>C\$1); (C\$1>1)); C\$1-1; IF(C\$1<10; C\$1+1; C\$1-1))	
2		=MOD(MOD((B1+A2); 100); 11)			
3					
4	=IF(AND((\$A2>\$A3); (\$A3>1)); \$A3-1; IF(\$A3<10; \$A3+1; \$A3-1))				
5					

Формулу из ячейки B2 скопировали во все ячейки диапазона B2:AD30. Формулу из ячейки A4 скопировали во все ячейки диапазона A4:A30. Формулу из ячейки D1 скопировали во все ячейки диапазона D1:AD1.

В каждую из ячеек A2 и A3 либо поместили число от 1 до 10, либо оставили ячейку пустой. В результате в ячейке A1 отобразилось значение 206, число в ячейке B2 оказалось нечётным, число в ячейке B3 совпало с числом в ячейке B2. Определите для каждой из ячеек A2 и A3, поместили ли туда число (и тогда какое) или оставили её пустой. Если ячейка осталась пустой, напишите -, в противном случае укажите число. В ответ запишите через пробел два значения – значение в ячейке A2 и значение в ячейке A3.

Пример записи ответа: 12 -

Примечание: таблица соответствия имён используемых функций.

	Google Sheets	Excel	LibreOffice
Условное вычисление	IF	ЕСЛИ	IF
Логическое И	AND	И	AND
Остаток от деления	MOD	ОСТАТ	MOD
Число ячеек, для которых верно условие	COUNTIF	СЧЁТЕСЛИ	COUNTIFS

Задание 8 (3 балла) и Задание 9 (4 балла). Технологии программирования

В отдельном архиве опубликованы тесты, решения жюри и проверяющие программы для задач по технологиям программирования. Архив содержит 4 директории: по два варианта каждой из двух задач. В поддиректории tests каждой задачи содержатся тесты и ответы решения жюри в файлах "???" и "???.a" соответственно, где "???" -- двузначное число с, возможно, ведущим нулем. В поддиректории solutions каждой задачи содержатся решения жюри на различных языках программирования, а также примеры неверных решений (см. файлы с расширением *.desc, чтобы классифицировать решения). В корневой директории архива находится программа, использовавшаяся для проверки ответа участника (check.cpp). В поддиректории statements содержатся условия задач. Апелляции заданий по программированию должны быть основаны на предоставленных тестах.

Наименования директорий с задачами (по вариантам): задача 8 – robot-pixel-art-v1, robot-pixel-art-v2; задача 9 – robot-connection-v1, robot-connection-v2.

Задача 9А1. Покраска холста

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Радиуправляемый робот умеет перемещаться по клетчатому полю размером $n \times t$ (n строк и t столбцов) и красить клетки в один из четырех цветов (красный, зеленый, синий и белый). Будем обозначать клетку на пересечении i -й сверху строки и j -го слева столбца как (i, j) .

Робот выполняет *команды* пользователя, при этом перемещаясь по полю в соответствии с заданными *настройками* и *ограничениями*.

Настройки представляют собой матрицу S размера $n \times t$, каждый элемент которой — либо $\emptyset$, либо пара из направления (вправо, вверх, влево, вниз) и цвета. Если $S_{i,j} = (d, c)$, то после того, как робот красит клетку (i, j) в какой-либо цвет, он сразу же смещается на одну клетку в направлении d и красит ее в цвет c . Если для новой покрашенной клетки $S_{i',j'} \neq \emptyset$, процесс продолжается по тем же правилам.

Ограничения бывают двух типов:

- ограничение на максимальное разрешенное число клеток цвета c ;
- запрет наличия на поле квадрата 2×2 , покрашенного цветами $\begin{pmatrix} c_{1,1} & c_{1,2} \\ c_{2,1} & c_{2,2} \end{pmatrix}$.

Пользователю доступны следующие *команды* для взаимодействия с роботом:

- «color i j c » — закрасить клетку (i, j) в цвет c , после чего выполнять действия в соответствии с настройками; процесс останавливается, когда
 - очередное перемещение привело робота за границу поля;
 - очередное перемещение привело робота в клетку, которую он уже красил в процессе выполнения текущей команды;
 - для очередной клетки $S_{i',j'} = \emptyset$;
 - с очередной покраской перестанет выполняться какое-то из ограничений.

Обратите внимание, что если первая же покраска клетки (i, j) в цвет c приводит к нарушению какого-то из ограничений, робот остановится сразу же, не покрасив ни одну клетку.

- «limit c x » — выставить ограничение на максимальное число клеток цвета c в x . Если в настоящий момент на поле уже больше x клеток цвета c , команда игнорируется и ограничение не меняется. Для каждого цвета в каждый момент времени действует только последнее введенное на него ограничение на число клеток.
- «block $c_{1,1}$ $c_{1,2}$ $c_{2,1}$ $c_{2,2}$ » — запретить появление квадратов 2×2 , раскрашенных цветами $\begin{pmatrix} c_{1,1} & c_{1,2} \\ c_{2,1} & c_{2,2} \end{pmatrix}$. Если в настоящий момент на поле уже есть квадрат, раскрашенный таким образом, команда игнорируется и ограничение не добавляется.
- «allow $c_{1,1}$ $c_{1,2}$ $c_{2,1}$ $c_{2,2}$ » — аналогично, отменить запрет на раскрашенные соответствующим образом квадраты 2×2 , если такой запрет сейчас есть.
- «settings i j d c » — выставить настройки для клетки (i, j) в значение (d, c) , где d — направление перемещения, а c — цвет, в который затем будет раскрашена клетка, в которую робот переместится.
- «settings i j none» — выставить настройки для клетки (i, j) в значение $\emptyset$, соответствующее отсутствию перемещения после покраски клетки (i, j) .

Еще раз повторим, что робот никогда не красит одну и ту же клетку дважды во время исполнения одной команды, а также останавливается до момента первого нарушения какого-либо ограничения. Например, если $S_{1,1} = (\rightarrow, \text{red})$, $S_{1,2} = (\downarrow, \text{blue})$, $S_{2,2} = (\leftarrow, \text{green})$ и $S_{2,1} = (\uparrow, \text{red})$, то при поступлении команды «color 1 1 blue», робот покрасит (1, 1) в синий, (1, 2) в красный, (2, 2) в синий и (2, 1) в зеленый. Затем робот остановится, так как клетка (1, 1) уже была покрашена при исполнении этой команды.

Изначально все настройки равны $\emptyset$, никакие ограничения не введены, а все клетки поля покрашены в белый цвет. Вам дан список из q команд пользователя, которые были последовательно отправлены роботу. Выведите раскраску поля после применения всех этих команд.

Формат входных данных

В первой строке записано целое число t ($1 \leq t \leq 1000$) — число наборов входных данных в тесте.

В первой строке каждого набора данных даны три целых положительных числа n , m и q — размеры поля и число команд. Гарантируется, что сумма $n \cdot m \cdot q$ по всем наборам входных данных не превосходит 10^5 .

В следующих q строках дано описание команд, посланных роботу в формате, описанном в условии. Направления задаются строками «right» (вправо), «up» (вверх), «left» (влево), «down» (вниз). Цвета задаются заглавными буквами 'R' (красный), 'G' (зеленый), 'B' (синий) и 'W' (белый).

Формат выходных данных

Для каждого набора входных данных выведите итоговую раскраску поля, полученную после обработки всех команд.

Пример

стандартный ввод	стандартный вывод
2	R G
3 2 11	G B
settings 1 1 down R	R W
settings 2 1 down R	W W W R
color 1 1 R	G R B R
limit B 0	W B B W
settings 2 1 right B	
color 2 1 G	
block R W G R	
color 2 2 R	
settings 2 2 up G	
limit B 1	
color 2 2 B	
3 4 10	
limit R 1	
settings 2 1 right R	
settings 2 2 right B	
settings 2 3 down B	
settings 3 3 left B	
settings 3 2 up B	
color 2 1 G	
limit R 3	
settings 2 4 up R	
color 2 4 R	

Замечание

В первом примере из условия

1. Команда «color 1 1 R» красит (1, 1) в красный, после чего из-за $S_{1,1} = (\downarrow, \text{red})$ клетка (2, 1) тоже красится в красный, а из-за $S_{2,1} = (\downarrow, \text{red})$ затем и (3, 1) красится в красный.

2. При выполнении «color 2 1 G» робот красит (2,1) в зеленый. $S_{2,1}$ в этот момент уже равно $(\rightarrow, \text{blue})$, поэтому после этого клетка (2,2) должна быть покрашена в синий, но это бы нарушило ограничение «limit B 0», поэтому процесс останавливается до этого.

3. После этого поле выглядит как

RW
GW
RW

4. Затем «color 2 2 R» должен покрасить (2,2) в красный, но это бы привело к получению квадрата с цветами RWGR, который запрещен, поэтому выполнение команды сразу останавливается.

5. Последняя команда покраски «color 2 2 B» отработывает корректно, так как до этого было разрешено иметь на поле не больше 1 синей клетки. После чего, в соответствии с $S_{2,2} = (\uparrow, \text{green})$, клетка (1,2) красится в зеленый.

6. Итоговый рисунок:

RG
GB
RW

Задача 9А2. Покраска холста

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Радиуправляемый робот умеет перемещаться по клетчатому полю размером $n \times t$ (n строк и t столбцов) и красить клетки в один из четырех цветов (красный, зеленый, синий и белый). Будем обозначать клетку на пересечении i -й сверху строки и j -го слева столбца как (i, j) .

Робот выполняет *команды* пользователя, при этом перемещаясь по полю в соответствии с заданными *настройками* и *ограничениями*.

Настройки представляют собой матрицу S размера $n \times t$, каждый элемент которой — либо $\emptyset$, либо пара из координат клетки и цвета. Если $S_{i,j} = ((i', j'), c)$, то после того, как робот красит клетку (i, j) в какой-либо цвет, он сразу же перемещается в клетку (i', j') и красит ее в цвет c . Если для новой покрашенной клетки $S_{i',j'} \neq \emptyset$, процесс продолжается по тем же правилам.

Ограничения бывают двух типов:

- ограничение на минимальное требуемое число клеток цвета c ;
- запрет наличия на поле квадрата 2×2 , покрашенного цветами $\begin{pmatrix} c_{1,1} & c_{1,2} \\ c_{2,1} & c_{2,2} \end{pmatrix}$.

Пользователю доступны следующие *команды* для взаимодействия с роботом:

- «fill i j with c » — закрасить клетку (i, j) в цвет c , после чего выполнять действия в соответствии с настройками; процесс останавливается, когда
 - очередное перемещение привело робота за границу поля;
 - очередное перемещение привело робота в клетку, которую он уже красил в процессе выполнения текущей команды;
 - для очередной клетки $S_{i',j'} = \emptyset$;
 - с очередной покраской перестанет выполняться какое-то из ограничений.

Обратите внимание, что если первая же покраска клетки (i, j) в цвет c приводит к нарушению какого-то из ограничений, робот остановится сразу же, не покрасив ни одну клетку.

- «at-least x c » — выставить ограничение на минимальное число клеток цвета c в x . Если в настоящий момент на поле меньше x клеток цвета c , команда игнорируется и ограничение не меняется. Для каждого цвета в каждый момент времени действует только последнее введенное на него ограничение на число клеток.
- «no-squares $c_{1,1}$ $c_{1,2}$ $c_{2,1}$ $c_{2,2}$ » — запретить появление квадратов 2×2 , раскрашенных цветами $\begin{pmatrix} c_{1,1} & c_{1,2} \\ c_{2,1} & c_{2,2} \end{pmatrix}$. Если в настоящий момент на поле уже есть квадрат, раскрашенный таким образом, команда игнорируется и ограничение не добавляется.
- «allow-squares $c_{1,1}$ $c_{1,2}$ $c_{2,1}$ $c_{2,2}$ » — аналогично, отменить запрет на раскрашенные соответствующим образом квадраты 2×2 , если такой запрет сейчас есть.
- «move i j to i' j' c » — выставить настройки для клетки (i, j) в значение $((i', j'), c)$, где (i', j') — клетка, в которую надо переместиться, а c — цвет, в который затем надо ее покрасить.
- «no-move i j » — выставить настройки для клетки (i, j) в значение $\emptyset$, соответствующее отсутствию перемещения после покраски клетки (i, j) .

Еще раз повторим, что робот никогда не красит одну и ту же клетку дважды во время исполнения одной команды, а также останавливается до момента первого нарушения какого-либо ограничения. Например, если $S_{1,1} = ((1,2), \text{red})$, $S_{1,2} = ((2,2), \text{blue})$, $S_{2,2} = ((2,1), \text{green})$ и $S_{2,1} = ((1,1), \text{red})$, то при поступлении команды «fill 1 1 with blue», робот покрасит (1,1) в синий, (1,2) в красный, (2,2) в синий и (2,1) в зеленый. Затем робот остановится, так как клетка (1,1) уже была покрашена при исполнении этой команды.

Изначально все настройки равны $\emptyset$, никакие ограничения не введены, а все клетки поля покрашены в белый цвет. Вам дан список из q команд пользователя, которые были последовательно отправлены роботу. Выведите раскраску поля после применения всех этих команд.

Формат входных данных

В первой строке записано целое число t ($1 \leq t \leq 1000$) — число наборов входных данных в тесте.

В первой строке каждого набора данных даны три целых положительных числа n , m и q — размеры поля и число команд. Гарантируется, что сумма $n \cdot m \cdot q$ по всем наборам входных данных не превосходит 10^5 .

В следующих q строках дано описание команд, посланных роботу в формате, описанном в условии. Цвета задаются строками «red» (красный), «green» (зеленый), «blue» (синий) и «white» (белый).

Формат выходных данных

Для каждого набора входных данных выведите итоговую раскраску поля, полученную после обработки всех команд. Для обозначения красного, зеленого или синего цвета используйте первую букву его английской записи ('r', 'g' или 'b'); для обозначения белого цвета используйте символ '.'.

Пример

стандартный ввод	стандартный вывод
2 3 2 11 move 1 1 to 2 1 red move 2 1 to 3 1 red fill 1 1 with red at-least 3 white move 2 1 to 2 2 blue fill 2 1 with green no-squares red white green red at-least 0 white fill 2 2 with red move 2 2 to 1 2 green fill 2 2 with blue 3 4 10 at-least 6 white move 2 1 to 2 2 red move 2 2 to 2 3 blue move 2 3 to 3 3 blue move 3 3 to 3 2 blue move 3 2 to 2 2 green fill 2 1 with green at-least 0 white move 2 4 to 1 4 red fill 2 4 with red	rg gb r. ...r grbr .bb.

Замечание

В первом примере из условия

1. Команда «`fill 1 1 with red`» красит $(1,1)$ в красный, после чего из-за $S_{1,1} = ((2,1), \text{red})$ клетка $(2,1)$ тоже красится в красный, а из-за $S_{2,1} = ((3,1), \text{red})$ затем и $(3,1)$ красится в красный.
2. При выполнении «`fill 2 1 with green`» робот красит $(2,1)$ в зеленый. $S_{2,1}$ в этот момент уже равно $((2,2), \text{blue})$, поэтому после этого клетка $(2,2)$ должна быть покрашена в синий, но это бы нарушило ограничение «`at-least 3 white`», поэтому процесс останавливается до этого.

3. После этого поле выглядит как

r.
g.
r.

4. Затем «`fill 2 2 with R`» должен покрасить $(2,2)$ в красный (ограничение на число белых уже снято), но это бы привело к получению квадрата с цветами `r.gr`, который запрещен, поэтому выполнение команды сразу останавливается.
5. Последняя команда покраски «`fill 2 2 with B`» выполняется. После чего, в соответствии с $S_{2,2} = ((1,2), \text{green})$, клетка $(1,2)$ красится в зеленый.

6. Итоговый рисунок:

rg
gb
r.

Задача 9B1. Кластеры роботов

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

На производстве расположены n роботов, пронумерованных от 1 до n . Любая пара роботов может быть либо соединена проводом, либо нет. Всего на производстве m проводов и i -й из них сейчас соединяет роботов с номерами a_i и b_i .

Вы находитесь рядом с роботом номер 1. Роботом номер t можно *управлять*, если он связан с первым некоторой цепочкой проводов, то есть

- либо если $t = 1$;
- либо если существуют такие $i_1, \dots, i_k$, что $i_1 = 1$, $i_k = t$, и любые два соседних в этой последовательности робота i_j и i_{j+1} связаны проводом.

Любому из роботов, которыми можно управлять, можно послать команду «извлеки второй конец подключенного к себе провода и подключи его к другому роботу». Иными словами, если роботом номер t можно управлять, и есть провод, соединяющий его с роботом номер x , то можно заменить провод (t, x) на провод (t, y) для любого $y \neq t$, еще не связанного проводом с t . Обратите внимание, что после этого вы можете потерять управление над t -м роботом, если единственная связь t -го с первым проходила через x -й.

Ваша задача — получить управление над максимально возможным количеством роботов на производстве. Найдите это количество и самую короткую последовательность действий, приводящую к такому исходу.

Формат входных данных

В первой строке записано целое число T ($1 \leq T \leq 1000$) — количество наборов входных данных в тесте.

Первая строка каждого набора входных данных содержит целые числа n и m ($1 \leq n \leq 10^5$; $0 \leq m \leq 1.5 \cdot 10^5$) — количество роботов и количество проводов, соответственно.

В следующих m строках дано описание проводов: в i -й строке даны целые числа a_i и b_i ($1 \leq a_i, b_i \leq n$; $a_i \neq b_i$) — номера роботов, соединенных i -м проводом. Гарантируется, что никакие два провода не соединяют одну и ту же пару роботов.

Также гарантируется, что сумма n по всем наборам входных данных не превосходит 10^5 и сумма m по всем наборам входных данных не превосходит $1.5 \cdot 10^5$.

Формат выходных данных

Для каждого набора входных данных выведите в первой строке через пробел максимальное количество роботов, управление над которыми можно получить, и количество действий k , которое для этого понадобится.

В следующих k строках выведите сами описания действий по одному на каждой строке. Описание действия должно состоять из трех целых чисел t , x и y ($1 \leq t, x, y \leq n$; $x \neq y$), означающих, что робот номер t меняет провод (t, x) на провод (t, y) .

Если возможных ответов несколько, выведите любой. Обратите внимание, что k при этом обязано быть **наименьшим**, при котором можно подключить максимальное количество роботов.

Пример

стандартный ввод	стандартный вывод
5	1 0
2 0	2 0
2 1	4 1
1 2	3 2 4
5 3	6 2
1 2	2 3 4
2 3	6 5 7
1 3	7 3
7 5	1 2 3
1 2	6 5 7
2 3	6 4 2
4 5	
5 6	
4 6	
7 7	
1 2	
3 4	
3 5	
3 6	
4 5	
4 6	
5 6	

Задача 9B2. Кластеры роботов

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

На производстве расположены n роботов, пронумерованных от 1 до n . Любая пара роботов может быть либо соединена проводом, либо нет. Известно, что i -й робот соединен с k_i другими роботами с номерами $v_{i,1}, \dots, v_{i,k_i}$. Провода двухсторонние, то есть если i связан с j , то j связан с i (иными словами, (i, j) и (j, i) — это один и тот же провод).

Вы находитесь рядом с роботом номер n . Роботом номер t можно *управлять*, если он связан с n -м некоторой цепочкой проводов, то есть

- либо если $t = n$;
- либо если существуют такие $i_1, \dots, i_k$, что $i_1 = n$, $i_k = t$, и любые два соседних в этой последовательности робота i_j и i_{j+1} связаны проводом.

Любому из роботов, которыми можно управлять, можно послать команду «извлеки второй конец подключенного к себе провода и подключи его к другому роботу». Иными словами, если роботом номер t можно управлять, и есть провод, соединяющий его с роботом номер x , то можно заменить провод (t, x) на провод (t, y) для любого $y \neq t$, еще не связанного проводом с t . Обратите внимание, что после этого вы можете потерять управление над t -м роботом, если единственная связь t -го с n -м проходила через x -й.

Ваша задача — получить управление над максимально возможным количеством роботов на производстве. Найдите это количество и самую короткую последовательность действий, приводящую к такому исходу.

Формат входных данных

В первой строке записано целое число T ($1 \leq T \leq 1000$) — количество наборов входных данных в тесте.

Первая строка каждого набора входных данных содержит одно целое число n ($1 \leq n \leq 10^5$) — количество роботов.

Затем следуют n строк, в i -й из которых записаны числа k_i ($0 \leq k_i \leq n - 1$) и $v_{i,1}, \dots, v_{i,k_i}$ ($1 \leq v_{i,j} \leq n$; $v_{i,j} \neq i$) — количество проводов, подключенных к i -му роботу, и номера роботов на противоположных концах этих проводов. Гарантируется, что данные корректны: никакие два провода не соединяют одну и ту же пару роботов, и если $j \in v_i$, то $i \in v_j$.

Также гарантируется, что сумма n по всем наборам входных данных не превосходит 10^5 и сумма количества проводов по всем наборам входных данных не превосходит $1.5 \cdot 10^5$.

Формат выходных данных

Для каждого набора входных данных выведите в первой строке через пробел максимальное количество роботов, управление над которыми можно получить, и количество действий k , которое для этого понадобится.

В следующих k строках выведите сами описания действий в формате « $y + t - x$ » ($1 \leq t, x, y \leq n$; $x \neq y$). Каждая такая строка соответствует замене провода (t, x) на провод (t, y) .

Если возможных ответов несколько, выведите любой. Обратите внимание, что k при этом обязано быть **наименьшим**, при котором можно подключить максимальное количество роботов.

Пример

стандартный ввод	стандартный вывод
5	1 0
2	2 0
0	4 1
0	1 + 4 - 3
2	6 2
1 2	4 + 2 - 1
1 1	3 + 6 - 5
5	7 3
0	3 + 7 - 1
0	2 + 6 - 5
2 4 5	1 + 6 - 4
2 3 5	
2 3 4	
7	
1 2	
2 1 7	
0	
2 5 6	
2 4 6	
2 4 5	
1 2	
7	
1 7	
0	
3 4 5 6	
3 3 5 6	
3 3 4 6	
3 3 4 5	
1 1	