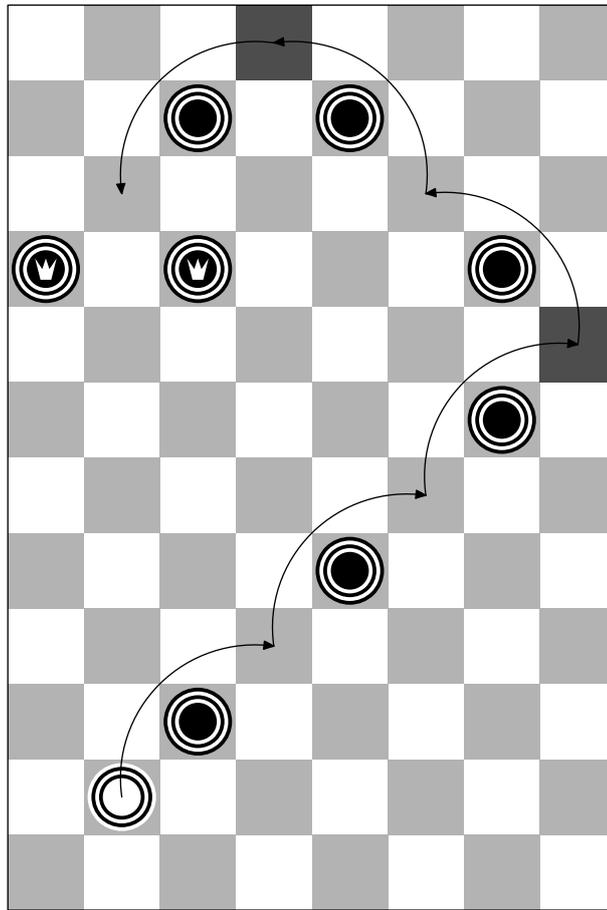


## Задача А. Достойный финал

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

Халк и Танос, решив, что на сегодня довольно драк, решили сыграть в шашки, чтобы выяснить, кто круче. Однако играют они по особым правилам.

- Поле имеет размеры  $h \times w$ , его строки пронумерованы снизу вверх числами от 1 до  $h$ , а столбцы — слева направо числами от 1 до  $w$ .
- Клетки, у которых номера столбца и строки имеют одинаковую четность, покрашены в черный цвет, а остальные — в белый.
- В некоторых черных клетках находятся белые или черные шашки. Ни в какой клетке не находится более одной шашки.
- Игроки ходят по очереди, за ход можно либо ничего не сделать, либо походить шашкой своего цвета: Халк — белой, Танос — черной.
- Если игрок ходит шашкой, он совершает ею последовательность *взятий*. А именно, пусть шашка игрока находится в клетке  $A$ , в соседней по углу клетке  $B$  находится шашка противника, а следующая в том же направлении клетка  $C$  пустая. Тогда игрок может *взять* шашкой в клетке  $A$  шашку в клетке  $B$ : при этом шашка игрока перемещается с  $A$  на  $C$ , а «перепрыгнутая» шашка противника снимается с поля, то есть клетка  $B$  оказывается пустой. Если после взятия походившая шашка способна взять ещё какую-нибудь шашку противника, то игрок может продолжать совершать *взятия* своей шашкой, пока желает.
- В игре есть дополнительное правило: нельзя за один ход более двух раз менять направление, в котором шашка движется, совершая взятия. Таким образом, в приведенной на рисунке ниже ситуации белая шашка не могла за свой ход взять ни одну из двух шашек, в которых нарисована корона: чтобы взять левую из них, ей необходимо было бы выпрыгнуть из поля, а чтобы взять правую — в третий раз сменить направление прыжков (первые два изменения направления произошли в выделенных клетках).



После нескольких ходов у Халка осталась всего одна пашка, однако сдаваться он не собирается. Помогите ему за один ход взять как можно больше пашек Таноса!

### Формат входных данных

В первой строке находится три целых числа  $h, w, n$  — высота, ширина поля и количество черных пашек ( $1 \leq h, w, n \leq 250\,000$ ).

Во второй строке находится два целых числа  $p, q$  — номера строки и столбца, на пересечении которых находится белая пашка ( $1 \leq p \leq h, 1 \leq q \leq w$ ).

В каждой из следующих  $n$  строк находится по два целых числа  $r_i, s_i$  — номера строки и столбца, на пересечении которых находится  $i$ -я черная пашка ( $1 \leq r_i \leq h, 1 \leq s_i \leq w$ ).

Гарантируется, что ни у какой черной пашки пара координат не совпадает с парой координат другой черной или белой пашки. Гарантируется, что все пашки находятся в клетках черного цвета.

### Формат выходных данных

Выведите одно целое число — наибольшее количество черных пашек, которое за один ход может взять белой пашкой Халк, следуя указанным выше правилам.

### Пример

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

## Задача В. Карточная игра

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

Это интерактивная задача.

В свободное от битв время, Тони Старк любит играть с Вижном в карты. Однако, обычные карточные игры им быстро надоели, и они решили придумать свою собственную.

У них есть колода из  $n$  карт с написанными на них числами от 1 до  $n$ , которые лежат на столе рубашкой вверх. Перед началом игры Вижн тщательно тасует эту колоду, а после этого в тайне от Тони переворачивает произвольные  $k$  карт рубашкой вниз. После этого задача Тони — не смотря на колоду, разделить ее на две непустых так, чтобы в них было поровну карт, лежащих рубашкой вниз. Для этого Тони (участник) может делать следующие действия:

- *Reverse* — перевернуть карту, лежащую наверху колоды;
- *Move* — отложить карту, лежащую наверху колоды, в отдельную колоду;
- *Up  $i$*  — переместить карту, находящуюся на  $i$ -м месте, считая сверху, наверх колоды. После этого верхние  $i - 1$  карт смещаются вниз;
- *Finish* — сообщение о том, что больше никаких действий с колодой производиться не будет. Это сообщение предполагает, что среди оставшихся в колоде карт и отложенных количество карт, лежащих рубашкой вниз, одинаково.

В ответ на каждое действие Вижн (жюри) сообщает Тони количество карт в отложенной колоде, лежащих рубашкой вниз.

Тони понимает, что Вижн намного умнее него благодаря камню Бесконечности, однако, все равно хочет победить. Поэтому он попросил вас помочь ему победить в этой игре.

### Протокол взаимодействия

Во время взаимодействия вашей программы с программой жюри происходит следующее: сначала на вход вашей программе дается два числа  $n$  и  $k$  ( $1 \leq n \leq 100$ ,  $0 \leq k \leq n$ ;  $n, k$  — четные), а затем несколько раз повторяются следующие действия:

- Ваша программа сообщает программе жюри одно из четырех действий, описанных в условии;
- Если это одно из первых трех действий, программа жюри сообщает количество карт, лежащих рубашкой вниз в отложенной колоде после выполнения этого действия;
- Если же это действие *Finish*, программа жюри проверяет, что количество карт, лежащих рубашкой вниз, в обоих колодах совпадает и завершает взаимодействие с вашей программой (соответственно, после этого действия и ваша программа должна завершиться).

Вашей программе разрешается выполнить не более 1000 действий.

### Замечание

После каждого действия вашей программы выводите символ перевода строки. Если вы используете «`writeln`» в Паскале, «`cout << ... << endl`» в C++, «`System.out.println`» в Java или «`print`» в Python, сброс потока вывода у вас происходит автоматически, дополнительно делать «`flush`» не обязательно. Если вы используете другой способ вывода, рекомендуется делать «`flush`», но все равно обязательно требуется выводить символ перевода строки.

Ниже приведены наиболее типичные причины получения тех или иных сообщений об ошибке.

Если ваша программа соблюдает протокол, но в итоге сдается, хотя изначально можно было выиграть, вы получите вердикт «Wrong Answer».

Если ваша программа выводит некорректно отформатированные сообщения программе жюри, то вы получите результат «Presentation Error» либо «Wrong Answer».

Если ваша программа нарушила протокол и ждет ввода в то же время, когда его ждет и программа жюри, то вы получите результат «Idleness Limit Exceeded». Обратите внимание, что к такому же результату может привести и то, что вы не переводите строку после каждого выведенного сообщения или выводите не тем способом, который описан в начале раздела, и не делаете «flush».

## Пример

стандартный ввод	стандартный вывод
6 4	Move
1	Reverse
1	Move
1	Up 4
1	Reverse
1	Move
2	Finish

## Замечание

В тестовом примере представлено возможное развитие событий:

- В колоде 6 карт, из них 4 рубашкой вниз;
- Откладываем верхнюю карту в отдельную колоду, она оказывается рубашкой вниз, в ответе жюри сообщает, что теперь в отложенной колоде одна карта лежит рубашкой вниз (соответственно, в колоде осталось 3 карты рубашкой вниз);
- Переворачиваем новую верхнюю карту в колоде;
- Теперь перекладываем ее в отдельную колоду, но она оказывается рубашкой вверх, возвращаемое значение не меняется (в колоде осталось 2 карты рубашкой вниз, так как до переворачивания верхняя карта лежала рубашкой вниз);
- Переносим четвертую (последнюю в текущей колоде) карту наверх;
- Переворачиваем ее;
- Откладываем ее в сторону, она оказывается рубашкой вниз. Теперь слева осталось 2 карты рубашкой вниз, и справа тоже (как нам только что сообщило жюри). Поэтому, можно заканчивать игру, мы победили.

## Задача С. Просчет событий

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

Ожидая Таноса на Титане, Доктор Стрэндж не терял время зря — он сел просчитывать вероятность победы в войне с помощью Глаза Агамотто, содержащего камень времени, пятый камень Бесконечности. Для этого он выделил  $n$  действий, которые могут сделать Мстители и  $q$  действий, про которые надо узнать, можно ли их сделать. Каждое действие Стрэндж обозначил числом, уникально описывающим его — действия, которые Мстители могут сделать, он обозначил числами  $a_1, a_2, \dots, a_n$ , а действия, про которые надо узнать возможность их выполнения —  $b_1, b_2, \dots, b_q$ .

Маг знает, что временной континуум устроен так, что если можно сделать действие, обозначенное числом  $x$  и действие, обозначенное числом  $y$ , то можно сделать и действия, обозначенные числами  $x \vee y$  и  $x \wedge y$  (где  $\vee$  и  $\wedge$  — побитовые операции «или» и «и» соответственно). Поэтому теперь про каждое из событий  $b_1, b_2, \dots, b_q$  осталось понять, можно ли их получить описанным выше способом. Помогите Стрэнджу справиться с этим заданием, ведь времени до прибытия Таноса на Титан осталось совсем немного. Обратите внимание, что одно действие можно совершать любое количество раз.

### Формат входных данных

В первой строке входного файла содержится число  $n$  — количество действий, которые могут выполнить Мстители ( $1 \leq n \leq 100\,000$ ).

В следующей строке содержится  $n$  чисел  $a_i$  — числа, описывающие эти действия ( $0 \leq a_i \leq 10^9$ ). Гарантируется, что все числа попарно различны.

В третьей строке содержится число  $q$  — количество действий, про которые надо узнать их возможность выполнения ( $1 \leq q \leq 100\,000$ ).

В последней строке содержится  $q$  чисел  $b_j$  — числа, описывающие эти действия ( $0 \leq b_j \leq 10^9$ ).

### Формат выходных данных

В  $i$ -й строке выходного файла выведите «YES», если действие, описанное числом  $b_i$ , можно выполнить и «NO» в противном случае.

### Пример

стандартный ввод	стандартный вывод
3	YES
1 3 4	NO
6	YES
1 2 3 4 5 6	YES
	YES
	NO

### Замечание

Числа 1, 3 и 4 можно получить не задействуя операций  $\vee$  и  $\wedge$ , а  $5 = 4 \vee 1$ .

## Задача D. Помогите спасти Землю!

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

Отряд супергероев спешит на помощь защитникам Земли на битву с Таносом. Супергерои сейчас находятся на Титане, и у них есть только один космический шатл, в который помещается  $k$  пассажиров и один пилот. Отряд состоит из одного мудрого предводителя и еще  $n$  супергероев. К сожалению, некоторые пары супергероев недолюбливают друг друга, могут повздорить, или даже того хуже — нанести вред друг другу. Но когда рядом находится предводитель, супергерои никогда не будут конфликтовать, потому что предводитель это жестко пресекает.

Решено, что шатлом будет управлять предводитель. Теперь его интересует вопрос, получится ли переправить всех супергероев на Землю. Процесс переправки будет происходить следующим образом:

- Не более чем  $k$  супергероев садятся в шатл вместе с предводителем. Среди оставшихся на Титане супергероев, не должно быть ни одной конфликтующей пары, иначе возможны печальные последствия.
- Шатл летит на Землю. Во время полета, супергерои на корабле находятся рядом с предводителем, поэтому не будут драться.
- На Земле из шатла высаживаются все супергерои, которые в нем находились. При этом среди супергероев, находящихся в этот момент на Земле, может быть конфликтующая пара. Они не подерутся, потому что рядом находится предводитель.
- Не более чем  $k$  супергероев садятся в шатл. Среди оставшихся на Земле супергероев не должно быть конфликтующей пары.
- Шатл летит на Титан.
- На Титане из шатла высаживаются все супергерои. Среди находящихся в этот момент на Титане супергероев, может быть конфликтующая пара.

Пункты повторяются по циклу, пока все супергерои и предводитель не окажутся на Земле. Предводитель может лететь с планеты на планету и вовсе без супергероев, если это необходимо. Если на планете нет предводителя, на ней не должно быть конфликтующей пары. Других ограничений на конфликтующие пары нет.

Вам дан список конфликтующих пар. Помогите предводителю определить порядок действий для транспортировки всех супергероев на Землю, либо сообщите, что это невозможно. Обратите внимание, что минимизировать количество полетов не требуется. Однако, предводителя интересует способ, содержащий не более 100 000 перелетов. Можно доказать при данных ограничениях, что если существует способ доставить всех на Землю, существует способ доставить всех на Землю за не более, чем 100 000 перелетов.

### Формат входных данных

В первой строке даны три целых числа  $n$ ,  $m$  и  $k$  — количество супергероев, количество конфликтующих пар и количество супергероев, помещающихся в шатл ( $1 \leq n \leq 15$ ,  $0 \leq m \leq n \cdot (n - 1)/2$ ,  $1 \leq k \leq 15$ ).

В следующих  $m$  строках даны пары целых чисел  $a_i$ ,  $b_i$  — номера супергероев, которые конфликтуют ( $1 \leq a_i, b_i \leq n$ ,  $a_i \neq b_i$ ).

Гарантируется, что каждая пара супергероев встречается не более одного раза, в том числе если во входных данных есть пара  $a_i$ ,  $b_i$ , тогда во входных данных не может быть пары  $b_i$ ,  $a_i$ .

## Формат выходных данных

Если невозможно переправить всех супергероев и предводителя на Землю, в единственной строке выведите  $-1$ .

Иначе, в первой строке выведите целое число  $t$  — количество перелетов, которое должен совершить шатл, чтобы на Земле оказались все супергерои и предводитель ( $t \leq 100\,000$ ). В  $i$ -й из следующих  $t$  строк выведите список супергероев, которые будут находиться в шатле во время  $i$ -го перелета. В начале выведите  $q_i$  — размер этого списка, а затем  $q_i$  различных чисел — номера супергероев, которые будут находиться в шатле во время  $i$ -го перелета.

## Пример

стандартный ввод	стандартный вывод
3 2 1	7
1 2	1 2
2 3	0
	1 3
	1 2
	1 1
	0
	1 2

## Замечание

Номер полета	На Титане	В полете	На Земле
—	1, 2, 3		
1	1, 3	2 →	
—	1, 3		2
2	1, 3	←	2
—	1, 3		2
3	1	3 →	2
—	1		2, 3
4	1	2 ←	3
—	1, 2		3
5	2	1 →	3
—	2		1, 3
6	2	←	1, 3
—	2		1, 3
7		2 →	1, 3
—			1, 2, 3

## Задача Е. Поиск корабля

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Стражи Галактики получили сигнал бедствия с разрушенного Таносом корабля. Им нужно как можно скорее обнаружить корабль и спасти находящегося на нем Тора.

Способ перемещения корабля весьма необычен. Существует  $n$  различных точек во Вселенной, пронумерованных от 1 до  $n$ , между которыми происходит перемещение корабля. Также существует  $m$  кротовых нор,  $i$ -я из которых соединяет собой точки с номерами  $a_i$  и  $b_i$ . Обратите внимание, что кротовая нора может соединять точку с самой собой, а также между двумя точками может быть более одной кротовой норы. Перемещение на корабле по каждой из кротовых нор занимает ровно один час.

Изначально корабль находился в точке с номером  $s$ . Стражи думают, что Тор пытался направить корабль к одной из точек, но по пути вынужден был остановиться. У них есть  $q$  предположений,  $i$ -е из которых состоит в следующем: Тор, начав свой путь в точке с номером  $s$ , двигался по одному из кратчайших путей в точку с номером  $v_i$ , однако, переместившись по кротовым норам  $k_i$  раз, корабль остановился и перестал двигаться дальше. Кратчайший путь — это путь, который содержит минимальное возможное количество кротовых нор.

Помогите Стражам определить для каждого предположения, может ли оно быть верно. А именно, для каждого предположения определите, верно ли, что существует хотя бы одна точка, в которой может сейчас находиться Тор, если да, то правда ли, что эта точка определена однозначно, и если да, то какая это точка.

### Формат входных данных

Первая строка входных данных содержит три целых числа  $n$ ,  $m$  и  $s$  — количество точек, кротовых нор и номер стартовой точки соответственно ( $1 \leq n, m \leq 50\,000$ ;  $1 \leq s \leq n$ ).

Каждая из следующих  $m$  строк содержит по два целых числа  $a_i$  и  $b_i$  — номера точек, соединенных  $i$ -й кротовой норой ( $1 \leq a_i, b_i \leq n$ ).

Следующая строка содержит единственное число  $q$  — количество предположений Стражей ( $1 \leq q \leq 50\,000$ ).

Каждая из следующих  $q$  строк содержит по два целых числа  $v_i$  и  $k_i$  — предполагаемый номер точки, куда направлялся Тор, и количество совершенных им перемещений по кротовым норам ( $1 \leq v_i \leq n$ ;  $1 \leq k_i \leq m$ ).

### Формат выходных данных

Для каждого из предположений выведите в отдельной строке результат проверки этого предположения, а именно:

- Если описанной в предположении ситуации быть не могло (например, если не существует пути от точки  $s$  до точки  $v_i$ ), выведите число 0;
- Если описанная в предположении ситуация могла быть, и в данный момент существует более одной точки, в которой может находиться корабль, выведите число  $-1$ ;
- В противном случае, то есть если положение корабля может быть определено однозначно, выведите номер точки, в которой согласно этому предположению должен располагаться корабль.

## Пример

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

## Задача F. Возвращение к домашней работе

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	10 секунд
Ограничение по памяти:	1024 мегабайта

Человек-паук только что вернулся домой после очередного задания. Он ужасно хочет спать, но у него еще есть невыполненная домашняя работа по математике на завтра. Вот, в чем она состоит: Вам дается строка  $s_0$  — начальное значение строки  $s$ , состоящей из символов 0, 1, 2 и 3. Далее, вам нужно обрабатывать запросы четырех типов:

1. Вам дается целое число  $p_i$  и строка  $w_i$ . Вы должны вставить строку  $w_i$  в текущую строку  $s$  перед символом на позиции  $p_i$ , если  $pos_i < |s|$  и после всех символов, если  $p_i = |s|$ .
2. Вам даются два целых числа  $p_i$  и  $l_i$ . Вы должны удалить из текущей строки  $s$  подстроку длины  $l_i$ , начинающуюся с символа на позиции  $p_i$ .
3. Вам даются два целых числа  $p_i$  и  $l_i$ . Вы должны развернуть в обратном порядке подстроку строки  $s$ , начинающуюся с символа на позиции  $p_i$ , длины  $l_i$ .
4. Вам даются три целых числа  $p_i$ ,  $l_i$  и  $k_i$ . Вы должны  $k_i$  раз продублировать подстроку, начинающуюся с символа на позиции  $p_i$ , имеющую длину  $l_i$ , сразу после этой подстроки. Иными словами, вы должны сразу после этой подстроки дописать эту подстроку  $k_i$  раз.

Символы в строке нумеруются с 0. Смотрите пояснение к примеру для лучшего понимания.

В каждый момент времени вас интересует длина наибольшей неубывающей подпоследовательности строки  $s$ . Наибольшей неубывающей подпоследовательностью называется максимальное по размеру множество чисел  $a_1 < a_2 < \dots < a_{m-1} < a_m$ , для которых верно, что  $s_{a_1} \leq s_{a_2} \leq \dots \leq s_{a_{m-1}} \leq s_{a_m}$ .

### Формат входных данных

В первой строке дана непустая строка  $s_0$ , состоящая из символов 0...3 — начальное значение строки  $s$  ( $|s_0| \leq 100\,000$ ). В следующей строке дано целое число  $q$  — количество запросов ( $0 \leq q \leq 15\,000$ ). В следующих  $q$  строках дано описание запросов. Если строка начинается с 1, то это запрос первого типа, далее дано целое число  $p_i$  и непустая строка  $w_i$ , состоящая из символов 0...3 ( $0 \leq p_i \leq |s|$ ,  $1 \leq |w_i| \leq 100\,000$ ). Если строка начинается с 2, то это запрос второго типа, далее даны два целых числа  $p_i$  и  $l_i$  ( $1 \leq l_i < |s|$ ,  $0 \leq p_i \leq |s| - l_i$ ). Если строка начинается с 3, то это запрос третьего типа, далее даны два целых числа  $p_i$  и  $l_i$  ( $1 \leq l_i < |s|$ ,  $0 \leq p_i \leq |s| - l_i$ ). Если строка начинается с 4, то это запрос четвертого типа, далее даны три целых числа  $p_i$ ,  $l_i$  и  $k_i$  ( $1 \leq l_i < |s|$ ,  $0 \leq p_i \leq |s| - l_i$ ,  $1 \leq k_i \leq 10^{15}$ ).

Гарантируется, что в любой момент времени строка  $s$  не пуста, и ее длина не превышает  $10^{15}$ . Гарантируется, что сумма длин всех  $w_i$  не превышает 100 000.

### Формат выходных данных

Выведите  $q + 1$  целое число — длину наибольшей неубывающей подпоследовательности  $s$  до всех запросов и после каждого из запросов.

### Пример

стандартный ввод	стандартный вывод
0103020	4
4	6
1 5 01	5
2 1 3	4
3 2 3	8
4 0 3 2	

## **Замечание**

Вот значения  $s$ . Выделены символы, образующие одну из возможных наибольших неубывающих подпоследовательностей.

**0103020**

**010300120**

**000120**

**002100**

**002002002100**