

Задача 1. Космическая станция

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

Давным-давно марсиане построили и вывели в космос Межпланетарную космическую станцию, которая предназначалась для исследования Солнечной системы. Но с момента ее запуска уже прошло много времени, и срок эксплуатации станции подходит к концу. По этой причине было принято непростое решение: станция должна быть выведена из строя, а все ее части должны быть отправлены обратно на Марс для дальнейшей утилизации и переработки.

Межпланетарная космическая станция состоит из n модулей, образующих древовидную структуру. Все модули пронумерованы целыми числами от 1 до n . Модуль с номером 1 является *корневым* модулем. Все остальные модули имеют *непосредственного предка* p_i , к которому они пристыкованы. *Предками* модуля i будем называть непосредственного предка этого модуля (т. е. модуль p_i), а также всех предков модуля p_i . Дополнительно будем считать, что каждый модуль является предком для самого себя. Иными словами, предками модуля i являются все модули на кратчайшем пути от этого модуля до корневого. *Поддеревом* модуля i назовем множество всех модулей, для которых i является предком (не обязательно непосредственным). Модуль i будем считать *концевым*, если он не является непосредственным предком никакого модуля.

Задача по выводу станции из строя была поручена двум талантливым инженерам — Тсетноку и Ачадазу. Но прежде чем заняться делом, инженеры решили сыграть в такую игру. Для начала они выбирают некоторый модуль с номером x . Затем они делают ходы по очереди, при этом первым ходит Тсетнок. За один ход игрок может выбрать любой **концевой** модуль w , который находится в поддереве модуля x , не совпадает с x , и при этом x не является непосредственным предком w . Пусть непосредственный предок модуля w — это модуль v . Тогда игрок отстыковывает модуль w от модуля v и пристыковывает его к непосредственному предку модуля v . Можно заметить, что после хода игрока станция по-прежнему сохраняет свою древовидную структуру, а модуль w остается в поддереве модуля x .

Начальство Тсетнока и Ачадаза хочет, чтобы они поскорее приступили к работе, а не играли вместо этого в игры с важными и дорогостоящими космическими станциями. Поэтому было принято решение рассказать инженерам, кто выигрывает при оптимальной игре для любого изначально выбранного модуля x . Предполагается, что им станет неинтересно играть, если они будут заранее знать победителя, и Тсетнок и Ачадаз наконец-то займутся полезным делом.

Для реализации этого хитрого плана осталось лишь написать программу, которая по заданной структуре станции сможет найти победителя. Справитесь ли Вы с этим заданием?

Формат входных данных

В первой строке входных данных находится целое число n ($2 \leq n \leq 2 \cdot 10^5$) — количество модулей, из которых состоит Межпланетарная космическая станция.

Во второй строке входных данных находится $n - 1$ целое число p_2, p_3, \dots, p_n ($1 \leq p_i < i$) — непосредственный предок i -го модуля.

Формат выходных данных

Выведите строку, состоящую из n символов. x -й символ в строке должен быть равен «Т», если при начальном выборе модуля x победит Тсетнок. Иначе x -й символ в строке должен быть равен «А».

Система оценки

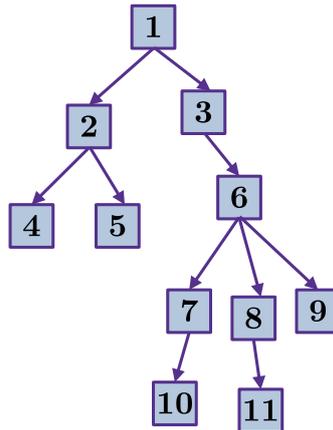
| № | Дополнительные ограничения | Баллы за подзадачу | Необходимые подзадачи |
|---|--------------------------------|--------------------|-----------------------|
| 1 | $n \leq 10$ | 13 | |
| 2 | $n \leq 1000$ | 29 | 1 |
| 3 | $p_i = 1$ | 3 | |
| 4 | $p_i = i - 1$ | 9 | |
| 5 | Нет дополнительных ограничений | 46 | 1 - 4 |

Примеры

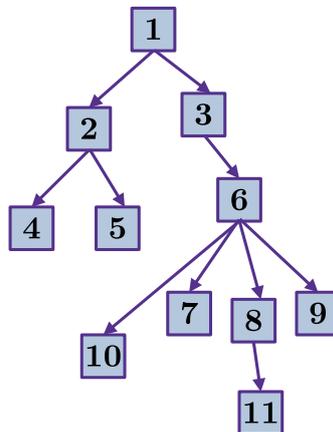
| стандартный ввод | стандартный вывод |
|---------------------------|-------------------|
| 11 1 1 2 2 3 6 6 6 7 8 | TATAAAAAAAAA |
| 3 1 2 | TAA |

Замечание

Рассмотрим первый пример из условия. В этом примере космическая станция будет выглядеть следующим образом:



Пусть $x = 3$. В поддереве модуля 3 находятся модули 3, 6, 7, 8, 9, 10 и 11. Первым ходом Тсетнок может выбрать модуль 9, 10 или 11, поскольку остальные модули не являются концевыми. Если Тсетнок выберет модуль 10, то он должен отстыковать этот модуль от модуля 7 и пристыковать его к модулю 6. В результате получится следующая ситуация:



Далее Ачадаз может выбирать модули 7, 9, 10 или 11. Если он, к примеру, выберет модуль 9, то он будет должен отстыковать его от модуля 6 и пристыковать к модулю 3.

В этой ситуации можно убедиться, что при оптимальной игре выиграет Тсетнок, поэтому в выходных данных третий символ должен быть равен «Т».

Теперь рассмотрим случай $x = 2$. В этом случае в поддереве модуля 2 находится два концевых модуля — 4 и 5. Но Тсетнок их выбрать не может, поскольку они пристыкованы к модулю $x = 2$. Следовательно, он не может сделать ход и проигрывает. Поэтому второй символ в выходных данных должен быть равен «А».

Задача 2. Новая игра Пети и Petya++

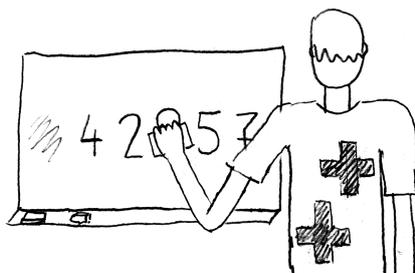
| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 1 секунда |
| Ограничение по памяти: | 256 мегабайт |

Мальчик Петя со своим другом, роботом Petya++, любит играть в различные игры. Недавно ребята придумали новую игру и решили сыграть в нее.

Правила игры описаны ниже. В начале на доске записывается массив из n чисел a_i слева направо. Игроки делают ходы по очереди, причем первым делает ход Петя, а Petya++ ходит вторым. За один ход игрок может стереть любое число с доски. Игра заканчивается, когда остается ровно k чисел. Тогда ребята смотрят на самое левое число, которое не было стерто. Это число является *счетом игры*. Задача Пети — сделать так, чтобы счет игры был как можно больше, а задача Petya++ — чтобы он был как можно меньше.

Ребята успели сыграть в эту игру несколько раз, после чего Petya++ заявил, что знает оптимальную стратегию. Петя не поверил своему другу и решил устроить ему испытание: он записывал на доске разные массивы a и просил Petya++ угадать, каков будет счет игры, если учесть, что игроки действуют оптимально.

К сожалению, Петя сам не знает ответа, поэтому он попросил Вас написать программу, которая умеет вычислять счет игры по заданным начальным числам.



Формат входных данных

В первой строке входных данных находится два целых числа n и k ($1 \leq n \leq 2 \cdot 10^5$, $1 \leq k \leq n$) — количество чисел, которые записаны на доске, и количество чисел, которые должны остаться в конце игры.

Во второй строке входных данных находится n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — записанный на доске массив.

Формат выходных данных

Выведите одно целое число — счет игры, если игроки действуют оптимальным образом.

Система оценки

| № | Дополнительные ограничения | Баллы за подзадачу | Необходимые подзадачи |
|---|--------------------------------|--------------------|-----------------------|
| 1 | $n \leq 8$ | 4 | |
| 2 | $n \leq 20$ | 18 | 1 |
| 3 | $a_i \leq 2$ | 29 | |
| 4 | Нет дополнительных ограничений | 49 | 1 - 3 |

Примеры

| стандартный ввод | стандартный вывод |
|--------------------|-------------------|
| 6 4 1 4 2 8 5 7 | 2 |
| 6 3 1 4 2 8 5 7 | 4 |

Замечание

В первом примере Петя стирает число 1. Затем Petya++ выгоднее всего стереть число 4. После этого на доске остается всего четыре числа, поэтому игра заканчивается. Самое левое число на доске сейчас равно двум.

Во втором примере Петя может стереть число 1 первым ходом. Petya++ невыгодно теперь стирать 4, поскольку в этом случае Петя своим следующим ходом сотрет 2, и счет игры станет равным 8. Поэтому Petya++ своим первым ходом стирает число 8. Далее Петя стирает любое число, кроме 4. На доске остается три числа, самое левое из которых — 4, поэтому счет игры равен четырем.

Задача 3. Модели компьютеров

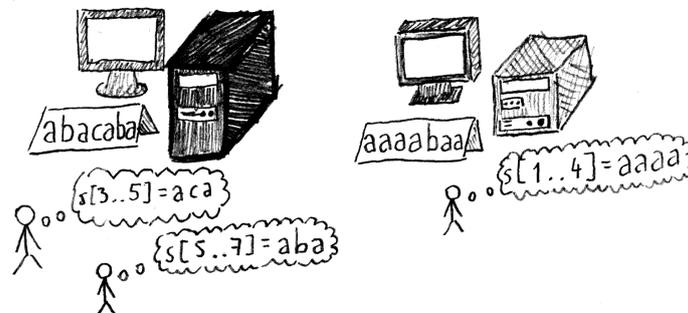
| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 256 мегабайт |

У марсиан часто возникают различные вычислительные задачи во всех без исключения сферах деятельности. Чтобы решать эти задачи быстро и эффективно, им нужно огромное количество компьютеров. Компания «Black Hole Semiconductors, Inc.», один из крупнейших поставщиков компьютерной техники на Марсе, производит огромное количество устройств, чтобы удовлетворить интересы своих заказчиков.

На данный момент компания должна обслужить n заказчиков. Марсиане считают, что «как компьютер назовешь, так он и будет считать», поэтому у каждого заказчика есть свои особые требования на название модели компьютера. Название модели каждого компьютера — это некоторая строка s , состоящая из m строчных букв латинского алфавита. i -й заказчик требует, чтобы на позициях от l_i до r_i в названии модели компьютера содержалась подстрока w_i . Иными словами, если i -му заказчику будет доставлен компьютер с названием модели s , то тогда должно выполняться $s_{l_i} s_{l_i+1} \dots s_{r_i} = w_i$.

Конечно, компания может выпустить отдельную модель компьютеров для каждого заказчика. Но есть проблема: разработка новых моделей отнимает много времени и средств. А когда модель уже разработана, то возможно легко производить сколь угодно много компьютеров такой модели. Поэтому «Black Hole Semiconductors, Inc.» хочет обслужить всех своих заказчиков таким образом, чтобы минимизировать количество моделей, которые им придется разработать.

Вы работаете в отделе исследований компании «Black Hole Semiconductors, Inc.» Руководство поручило Вам важную задачу: придумать названия моделей компьютеров таким образом, чтобы удовлетворить пожелания всех клиентов, и при этом количество моделей было минимально.



Формат входных данных

В первой строке входных данных находится два целых числа n и m ($1 \leq n \leq 500$, $1 \leq m \leq 10^3$) — количество заказчиков и длина названия модели компьютера.

В каждой из следующих n строк входных данных находится два целых числа l_i , r_i , а также строка w_i , состоящая из строчных букв латинского алфавита ($1 \leq l_i \leq r_i \leq m$, $|w_i| = r_i - l_i + 1$) — требование на название модели, которое выдвигает i -й заказчик.

Формат выходных данных

В первой строке выходных данных выведите целое число k ($1 \leq k \leq n$) — количество моделей компьютеров, которые должна выпустить компания.

В каждой из следующих k строк выведите строку s_i , состоящую из m строчных букв латинского алфавита — название i -й модели компьютера.

Количество выведенных Вами моделей компьютеров должно быть минимально и должно удовлетворять условию задачи, т. е. для каждого заказчика должна найтись модель, которая удовлетворяет его требованиям.

Если существует несколько способов придумать названия моделей компьютеров, чтобы их количество было минимальным, то разрешается вывести любой из них.

Система оценки

| № | Дополнительные ограничения | Баллы за подзадачу | Необходимые подзадачи |
|---|--|--------------------|-----------------------|
| 1 | $n \leq 2, m \leq 100$ | 4 | |
| 2 | $n \leq 10, m \leq 100$ | 11 | 1 |
| 3 | $n \leq 15, m \leq 100$ | 21 | 1, 2 |
| 4 | $l_1 \leq l_2 \leq \dots \leq l_n, r_1 \geq r_2 \geq \dots \geq r_n$ | 6 | |
| 5 | Нет дополнительных ограничений | 58 | 1 - 4 |

Примеры

| стандартный ввод | стандартный вывод |
|---|--|
| 5 7 3 3 a 5 7 aba 3 5 aca 1 5 abaca 1 4 aaaa | 2 abacaba aaaabaa |
| 8 18 6 10 brest 11 17 vitebsk 1 5 gomel 6 11 grodno 1 5 minsk 12 18 mogilev 3 3 n 10 10 w | 3 minskbrestvitebskz gomelgrodnomogilev wwwwwwwwwwwwwww |
| 5 9 1 9 aaaaaaaaaa 2 8 aaaaaaa 3 7 aaaaa 4 6 aaa 5 5 a | 1 aaaaaaaaa |

Замечание

Рассмотрим первый пример. В нем компьютеры модели «abacaba» подходят под требования первого, второго, третьего и четвертого заказчиков, компьютеры модели «aaaabaa» — под требования первого и пятого заказчиков. Таким образом, каждый заказчик получит компьютеры. Можно показать, что с помощью всего одной модели компьютеров нельзя удовлетворить требования всех заказчиков, поэтому минимальное количество моделей равно двум.

Задача 4. Экспериментальная установка

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

Марсианские ученые продолжают свои эксперименты с антинейтрино, чтобы создать новые, более эффективные двигатели для межгалактических полетов. Чтобы проводить эксперименты, ученые создали установку, в которой n антинейтрино расположены в ряд. Каждый антинейтрино имеет *направление*: он направлен либо вправо, либо влево.

Если антинейтрино в установке будут находиться под воздействием излучения, то они время от времени будут изменять свое состояние. Можно считать, что каждое изменение состояния происходит в разные моменты времени и имеет один из двух вариантов:

- Некоторый антинейтрино, который направлен вправо (кроме самого правого) делает своего соседа справа направленным вправо.
- Некоторый антинейтрино, который направлен влево (кроме самого левого) делает своего соседа слева направленным влево.

Важная характеристика ряда из антинейтрино — его *вариативность*. Эта величина равна количеству различных возможных состояний, в которые может прийти ряд под воздействием излучения, т. е. количество различных конфигураций ряда после применения некоторого (возможно, нулевого) количества описанных выше операций. Две конфигурации считаются различными, если некоторый антинейтрино в этих конфигурациях направлен в разные стороны.

Ученые проводят эксперименты на антинейтринной установке. Эксперименты заключаются в следующем:

- Они выбирают некоторый подотрезок из антинейтрино и изменяют направление каждого из них на противоположное.
- Они временно убирают некоторые антинейтрино слева и справа ряда, оставляя некоторый подотрезок. Затем ученые измеряют вариативность оставшегося ряда. Наконец, они возвращают ряд в то состояние, каким он был до начала эксперимента.

К сожалению, эксперименты на антинейтринной установке проводить дорого, а измерения оставляют желать лучшего — вариативность ряда из антинейтрино измеряется с большими погрешностями. По этой причине было решено написать программу, которая могла бы симулировать эксперименты на реальной установке. А поскольку ученым надо провести много экспериментов над большим количеством антинейтрино, то программа должна быть быстрой и эффективной.

Ввиду того, что значение вариативности может быть довольно большим, ученых интересует не его точное значение, а лишь остаток от деления на $10^9 + 7$.

Реализация требуемой программы была поручена Вам. Справитесь?

Формат входных данных

В первой строке входных данных находится два целых числа n и q ($1 \leq n, q \leq 3 \cdot 10^5$) — количество антинейтрино в установке и количество экспериментов, которые будут проведены учеными.

Во второй строке входных данных находится строка s длины n , состоящая из символов «<» и «>» — описание начального состояния установки. Если i -й символ строки равен «<», то i -й антинейтрино в ряду направлен влево, иначе i -й антинейтрино в ряду направлен вправо.

В каждой из следующих q строк находится три целых числа t_i , l_i и r_i ($1 \leq t_i \leq 2$, $1 \leq l_i \leq r_i \leq n$) — описание i -го эксперимента. Величина t_i указывает на тип эксперимента. Если $t_i = 1$, то тогда ученые изменяют направление всех антинейтрино на отрезке с l_i по r_i на противоположное. Если $t_i = 2$, то тогда ученые находят вариативность отрезка антинейтрино с l_i по r_i .

Формат выходных данных

Для каждого эксперимента второго типа выведите по одному целому числу в отдельной строке — остаток от деления вариативности ряда на число $10^9 + 7$. Ответы следует выводить в том же порядке, в котором эксперименты заданы во входных данных.

Система оценки

| № | Дополнительные ограничения | Баллы за подзадачу | Необходимые подзадачи |
|---|---|--------------------|-----------------------|
| 1 | $n \leq 20, q = 1, t_i = 1, l_i = 1, r_i = n$ | 8 | |
| 2 | $n \leq 200, q = 1, t_i = 1, l_i = 1, r_i = n$ | 6 | 1 |
| 3 | $n \leq 2000, q = 1, t_i = 1, l_i = 1, r_i = n$ | 14 | 1, 2 |
| 4 | $q = 1, t_i = 1, l_i = 1, r_i = n$ | 15 | 1 - 3 |
| 5 | $t_i = 1$ | 16 | 1 - 4 |
| 6 | если $t_i = 2$, то $l_i = r_i$ | 11 | 1 - 5 |
| 7 | $n, q \leq 8 \cdot 10^4$ | 13 | 1 - 3 |
| 8 | Нет дополнительных ограничений | 17 | 1 - 7 |

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 5 6 >>><< | 6 |
| 2 1 5 | 1 |
| 1 1 5 | 9 |
| 2 1 5 | |
| 1 1 5 | |
| 1 3 4 | |
| 2 2 5 | |

Замечание

Рассмотрим пример из условия.

В первом событии вариативность будет равна шести, т. к. под воздействием излучения может получиться шесть возможных состояний: «<<<<<<», «><<<<<», «>><<<<», «>>><<<», «>>>><<» и «>>>>>>».

После второго события все антинейтрино изменят свое направление, и конфигурация будет выглядеть как «<<<>>>».

В третьем событии описанные операции никак не смогут изменить направление антинейтрино, поэтому вариативность равна 1.

После четвертого события конфигурация станет выглядеть как «>>><<<».

После пятого события конфигурация станет выглядеть как «>><><<».

В шестом событии необходимо вычислить вариативность ряда «><<><». Она равна девяти, поскольку возможно получить конфигурации «>>>>>>», «>>><<<», «>><<<<», «><>>>>», «><><<<», «><<<<<», «<<<>>>», «<<<<<<» и «<<<<<<».