

Problem A. Две кнопки

Input file: стандартный ввод
Output file: стандартный вывод
Time limit: 1 секунда
Memory limit: 256 мегабайт

Коля нашел странное устройство. На передней панели устройства расположен дисплей, отображающий некоторое целое положительное число, а также красная и синяя кнопки. При нажатии на красную кнопку число на дисплее увеличивается в два раза. При нажатии на синюю кнопку число на дисплее уменьшается на единицу. Если в какой-то момент число перестает быть положительным, устройство ломается. Дисплей может отображать сколь угодно большие числа.

Изначально на дисплее отображено число n . Вася нашел на задней панели устройства секретную кнопку, после нажатия на которую устройство перезагружается; после перезагрузки на дисплее снова отображается число n независимо от того, что на нем было до перезагрузки.

Вася хочет получить на дисплее число m , начиная с числа n . Какое минимальное количество нажатий ему придется совершить?

Input

Первая и единственная строка ввода содержит два различных целых числа n и m , разделенных пробелом ($1 \leq n, m \leq 10^4$).

Output

Выведите одно число — минимальное количество нажатий на кнопки, необходимое, чтобы получить число m из числа n .

Examples

стандартный ввод	стандартный вывод
4 6	2
10 1	9

Note

В первом примере нужно один раз нажать на синюю кнопку, и затем один раз на красную кнопку.

Во втором примере удваивать число невыгодно, поэтому надо девять раз нажать на синюю кнопку.

В этой задаче нет групп тестов - решения, работающие при заданных ограничениях, будут оцениваться в 100 баллов.

Problem B. Игра на подстроках

Input file: **stdin**
Output file: **stdout**
Time limit: 1 секунда
Memory limit: 256 мегабайт

Алиса и Боб играют в игру. Изначально выбраны строка s и её непустая подстрока p . За ход разрешается добавить символ в конец или начало p так, чтобы получившаяся строка оставалась подстрокой s . Тот, кто не может сделать ход, проигрывает. Алиса ходит первой. Алиса и Боб очень любят строку s_0 , и последние несколько месяцев в качестве строки s игроки используют именно её (с различными стартовыми подстроками p). Алисе кажется, что ей удалось понять, какие из стартовых подстрок p являются выигрышными для неё, а какие - проигрышными; Алиса хочет проверить себя и свое понимание этой игры. Помогите ей! Напишите программу, которая по заданной строке s_0 и номеру k выдаст k -ую в лексикографическом порядке подстроку p_k строки s_0 , выигрышную для Алисы.

Input

В первой строке два числа n, k . Во второй строке задана строка s_0 длины n из символов $a - z$.

Output

Одна строка — k -я выигрышная для Алисы подстрока строки s_0 . Гарантируется, что такая подстрока существует (Алиса точно знает число выигрышных подстрок).

Examples

stdin	stdout
2 1 fh	f
3 1 cba	ba

Note

В этой задаче будет две группы тестов, каждая из которых независимо оценивается в 50 баллов.

Ограничения в первой группе: $1 \leq n \leq 100000, k = 1$.

Ограничения во второй группе: $1 \leq n \leq 100, 1 \leq k$.

Две подстроки строки s считаются различными, если они различаются как строки. Например, в строке "aaa" всего три различные подстроки.

Строка s_1, s_2, \dots, s_N лексикографически меньше строки t_1, t_2, \dots, t_M , если существует такое натуральное число pos , $pos \leq \min(N, M)$, что $s_1 = t_1, s_2 = t_2, \dots, s_{pos-1} = t_{pos-1}$, но $pos \leq N$ и $s_{pos} < t_{pos}$, либо $pos = N + 1 \leq M$ (т.е. строка s является префиксом t). Например, строка *abacaba* лексикографически меньше строк *abbaaa*, *b* и *abacabadabacaba*, но не есть лексикографически меньше строк *abacab*, *abacaba* и *aa*.

Problem C. Дима, робот и перестановка

Input file: stdin
Output file: stdout
Time limit: 2 секунды
Memory limit: 256 мегабайт

Мало кто знает, но первоначальная версия романа-эпопеи "Война и мир" Льва Николаевича Толстого содержит ровно N томов. Диме пришлось потратить много сил и изъездить всю Москву, чтобы собрать все тома у себя. Вернувшись домой, Дима поставил книги на заранее заготовленную им полку и в изнеможении лег спать.

Проснувшись, Дима обнаружил, что тома расставлены не по порядку! Однако вставать Диме не хотелось, и он решил прибегнуть к помощи созданного им робота. Робот умеет выполнять последовательность инструкций, каждая из которых заключается в том, чтобы поменять местами две соседние (заданные инструкцией) книги на полке. Однако зарядки роботу хватает лишь на не более, чем K таких операций.

Дима хочет задать последовательность операций таким образом, чтобы итоговая перестановка томов оказалась лексикографически минимальной среди всех перестановок, которые могут реализоваться на полке (перестановка p_1, p_2, \dots, p_N лексикографически меньше перестановки q_1, q_2, \dots, q_N , если существует такое натуральное число pos , $pos \leq N$, что $p_1 = q_1, p_2 = q_2, \dots, p_{pos-1} = q_{pos-1}, p_{pos} < q_{pos}$).

Чтобы разработать последовательность инструкций, Диме нужно понять, к какой перестановке нужно стремиться. Помогите ему!

Input

Первая строка содержит два числа — N ($1 \leq N \leq 3 \cdot 10^5$) и K ($0 \leq K \leq 10^{18}$). Во второй строке задана последовательность номеров томов на полке — N попарно различных натуральных чисел, не превосходящих N .

Output

Выведите последовательность из N чисел — лексикографически минимальную перестановку томов, которая может получиться в результате действий робота.

Examples

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

Note

Решения, работающие при $1 \leq N \leq 10$, будут оцениваться в 30 баллов.

Решения, работающие при $1 \leq N \leq 1000$, будут оцениваться в 80 баллов.

Решения, работающие при $1 \leq N \leq 3 \cdot 10^5$, будут оцениваться в 100 баллов.

Problem D. Сплошные плюсы

Input file: `stdin`
Output file: `stdout`
Time limit: 1 секунда
Memory limit: 256 мегабайт

Вася сидит на невыносимо скучном уроке математики. Чтобы развлечься, он выписал на листок бумаги n цифр в одну строку. После этого Вася стал выписывать разные способы поставить в строке между некоторыми цифрами плюсы («+») таким образом, чтобы получалось корректное арифметическое выражение; формально, никакие два плюса в таком разбиении не должны стоять рядом (между любыми двумя соседними плюсами должна стоять хотя бы одна цифра), и никакой плюс не может стоять в начале или в конце строки. Например, в строке 100500, способы 100500 (не ставить плюсов совсем), 1+00+500 или 10050+0 являются корректными, а способы 100++500, +1+0+0+5+0+0 или 100500+ являются некорректными.

Урок был длинным, и Вася выписал все корректные способы расставить в своей строке из цифр ровно k плюсов. В этот момент его развлечение было замечено, и Вася получил задание до конца урока вычислить сумму значений всех получившихся арифметических выражений (при вычислении значения выражения ведущие нули в слагаемых следует игнорировать). Поскольку ответ может быть большим, Васе разрешили всего лишь получить остаток от деления ответа на $10^9 + 7$.

Input

В первой строке записано два целых числа n и k ($0 \leq k < n \leq 10^5$).

Во второй строке записана строка, состоящая из n цифр.

Output

Выведите ответ на задачу по модулю $10^9 + 7$.

Examples

stdin	stdout
3 1 108	27
3 2 108	9

Note

В первом примере результат равен $(1 + 08) + (10 + 8) = 27$.

Во втором примере результат равен $1 + 0 + 8 = 9$.

Решения, работающие при $n \leq 20$, будут оцениваться в 20 баллов.

Решения, работающие при $n \leq 200$, будут оцениваться в 40 баллов.

Решения, работающие при $n \leq 2000$, будут оцениваться в 60 баллов.

Решения, работающие при $n \leq 100000$, будут оцениваться в 100 баллов.

Problem E. Магазин

Input file: `stdin`
Output file: `stdout`
Time limit: 2.5 секунд
Memory limit: 256 мегабайт

Вася играет в одну очень известную и крайне популярную ММОПГ-игру. У игрового персонажа (героя) Васи имеется K характеристик, на текущий момент i -я из них равна a_i . Также в этой игре имеется общая рейтинговая таблица, в которой участники ранжируются согласно **произведению** всех характеристик героя, по убыванию величины.

Вася собирается "прокачать" своего персонажа, воспользовавшись услугами игрового магазина. Магазин предлагает N возможных способов улучшить характеристики героя, каждый из этих способов принадлежит одному из следующих трех типов:

1. сделать i -ю характеристику равной b ;
2. прибавить к i -ой характеристике b ;
3. увеличить i -ю характеристику в b раз.

К сожалению, каждым улучшением можно воспользоваться только один раз, а денег на карточке Васи хватит на покупку лишь не более чем M из этих N улучшений.

Помогите Васе достичь наибольшего рейтинга в игре. Для этого напишите программу, которая подскажет Васе, какие из улучшений надо приобрести и в каком порядке их использовать, чтобы рейтинговая статистика стала как можно больше. Если имеется несколько вариантов решения, выведите любой.

Input

Первая строка содержит три целых числа K, N, M ($1 \leq K \leq 10^5$, $0 \leq M \leq N \leq 10^5$), разделенные пробелом, — количество характеристик, общее количество доступных улучшений и максимальное число улучшений, которые Вася может использовать.

Во второй строке через пробел заданы K целых чисел a_1, a_2, \dots, a_K ($1 \leq a_i \leq 10^6$) — начальные значения характеристик.

Далее в N строках через пробел заданы 3 целых числа t_j, i_j, b_j ($1 \leq t_j \leq 3$, $1 \leq i_j \leq K$, $1 \leq b_j \leq 10^6$) — тип j -го улучшения (1 для присваивания, 2 для прибавления, 3 для умножения), номер характеристики, к которой его можно применить, и коэффициент b , с которым он действует.

Output

Первая строка должна содержать целое число L ($0 \leq L \leq M$) — количество использованных улучшений.

Вторая строка должна содержать L различных целых чисел v_1, v_2, \dots, v_L ($1 \leq v_i \leq N$), разделенных пробелом, — номера улучшений в том порядке, в котором их нужно совершать. Улучшения нумеруются начиная с 1 в порядке, в котором они указаны во входном файле.

Examples

stdin	stdout
2 4 3 13 20 1 1 14 1 2 30 2 1 6 3 2 2	3 2 3 4

Note

В данной задаче пять групп тестов.

В первой группе тестов $N, K \leq 10$, $1 \leq a_i, b_i \leq 8$.

Во второй группе тестов $N, K \leq 10^5$, $1 \leq a_i, b_i \leq 10^6$, все доступные улучшения имеют тип 1.

Во третьей группе тестов $N, K \leq 10^5$, $1 \leq a_i, b_i \leq 10^6$, все доступные улучшения имеют тип 2.

В четвертой группе тестов $N, K \leq 10^5$, $1 \leq a_i, b_i \leq 10^6$, все доступные улучшения имеют тип 3.

В пятой группе тестов дополнительных ограничений нет.

За прохождение первой группы тестов начисляется 10 баллов.

За прохождение второй группы тестов начисляется 15 баллов.

За прохождение третьей группы тестов начисляется 25 баллов.

За прохождение четвертой группы тестов начисляется 10 баллов.

За прохождение всех пяти групп начисляется 100 баллов.

Баллы за прохождение первых четырех групп начисляется независимо по каждой группе; за пятую группу баллы может набрать только корректно работающая на всех тестах жюри программа.

Problem F. Cycling City

Input file:	стандартный ввод
Output file:	стандартный вывод
Time limit:	2 секунды
Memory limit:	256 мегабайт

Вы организуете велосипедную гонку на улицах города. В городе присутствует n перекрестков, некоторые пары из которых соединены дорогами; по каждой из дорог можно перемещаться в любую сторону. Никакие две дороги не соединяют одну и ту же пару перекрестков, и никакая дорога не соединяет перекресток с самим собой.

Вы хотите, чтобы в гонке могли принять участие как профессиональные спортсмены, так и начинающие велосипедисты, и для этого вы проведете гонку в трех номинациях: легкая, средняя и сложная; каждый участник выберет себе сложность по силам. Для каждой номинации необходимо выбрать свой маршрут — цепочку перекрестков, последовательно соединенных дорогами. Маршруты должны удовлетворять следующим условиям:

- все три маршрута должны стартовать на одном и том же перекрестке, а также финишировать на одном и том же перекрестке (места старта и финиша не могут совпадать);
- во избежание столкновений ни через один перекресток и ни через одну дорогу не может проходить более одного маршрута (за исключением общего старта и финиша);
- ни один маршрут не должен дважды проходить через один и тот же перекресток.

Подготовка к соревнованиям скоро начнется, и вам нужно как можно быстрее определиться с маршрутами гонки. Длина маршрутов не играет роли, важно лишь, чтобы все перечисленные требования были удовлетворены.

Input

В первой строке записано два целых числа n и m ($1 \leq n, m \leq 2 \cdot 10^5$) — количество перекрестков и дорог соответственно.

В следующих m строках через пробел записано по два целых числа — номера перекрестков, соединенных очередной дорогой (перекрестки нумеруются, начиная с 1). Гарантируется, что каждая пара перекрестков соединена не более, чем одной дорогой, и никакая дорога не соединяет перекресток сам с собой.

Обратите внимание, что возможность добраться от любого перекрестка до любого другого по дорогам не гарантируется.

Output

Если проложить маршруты возможно, в первой строке выведите «YES». В следующих трех строках выведите описания маршрутов в формате « $l_i \ p_{i1} \dots p_{il_i}$ »; это означает, что i -ый маршрут начинается на перекрестке с номером p_{i1} , затем проходит по дороге к перекрестку p_{i2} , \dots , и, наконец, завершается на перекрестке с номером p_{il_i} . Маршруты должны удовлетворять всем приведенным в условии требованиям.

Если проложить маршруты требуемым образом невозможно, выведите «NO».

Examples

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

Note

Решения, работающие при $n \leq 70$, будут оцениваться в 60 баллов.

Решения, работающие при $n \leq 2 \cdot 10^5$, будут оцениваться в 100 баллов.