

Задача А. Танец-палиндром

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Группа из n танцоров репетирует выступление на закрытии олимпиады. Танцоры выстроены в ряд, они выучили свои движения и не могут меняться местами. Для некоторых из них уже куплен белый танцевальный костюм, для некоторых уже куплен чёрный костюм, а с костюмами остальных руководитель ансамбля ещё не определился.

Как раз в день, когда он собрался завершить художественный замысел и заказать пошив оставшихся костюмов, ему сообщили, что участники олимпиады будут в восторге, если цвета костюмов танцоров будут составлять палиндром, если смотреть на них в том порядке, в котором танцоры выстроены на сцене. Напомним, что палиндромом называется последовательность, одинаково читающаяся как слева направо, так и справа налево. Руководителю ансамбля идея очень понравилась, и теперь он хочет заказать оставшиеся костюмы таким образом, чтобы цвет костюма первого слева танцора совпадал с цветом костюма первого справа танцора, цвет костюма второго слева танцора совпадал с цветом костюма второго справа танцора и так далее.

Руководитель знает, сколько бурлей стоит один чёрный костюм и сколько бурлей стоит один белый костюм. Вам требуется определить, возможно ли заказать оставшиеся костюмы таким образом, чтобы полученная последовательность цветов костюмов танцоров была палиндромом, и, если это возможно, то какое минимальное суммарное количество бурлей потребуется потратить? Напомним, что танцоры не могут меняться местами, а также в силу бюрократических формальностей не разрешается заказывать новые костюмы для танцоров, чьи цвета на сцене уже определены, даже если это позволит получить ответ меньшей итоговой стоимости.

Формат входных данных

В первой строке содержатся три целых числа n , a и b ($1 \leq n \leq 20$, $1 \leq a, b \leq 100$) — количество танцоров, стоимость одного белого костюма и стоимость одного чёрного костюма.

В следующей строке содержатся n чисел c_i , i -е из которых означает цвет костюма i -го слева танцора. Число 0 соответствует белому костюму, 1 — чёрному, а 2 означает, что для этого танцора цвет костюма только предстоит выбрать. В следующей строке содержатся n чисел s_i , i -е из которых означает цвет костюма i -го слева танцора.

Формат выходных данных

Если невозможно добиться, чтобы цвета танцоров образовывали палиндром, не меняя танцоров местами, и не меняя уже заказанные костюмы, то выведите -1 . Иначе выведите минимальное количество бурлей, которое придётся потратить для получения желаемого художественного эффекта.

Примеры

стандартный ввод	стандартный вывод
5 100 1 0 1 2 1 2	101
3 10 12 1 2 0	-1
3 12 1 0 1 0	0

Замечание

В первом примере наиболее дешёвым способом получить палиндромную раскраску будет заказать чёрный костюм для третьего слева танцора и белый костюм для самого правого танцора.

Во втором примере костюм самого левого танцора уже отличается от костюма самого правого танцора, поэтому добиться желаемой раскраски не получится.

В третьем примере все костюмы уже заказаны и цвета танцоров составляют палиндром.

Система оценки

В данной задаче 103 теста. Тесты из условия не оцениваются, остальные тесты стоят по 1 баллу каждый.

Решения, верно работающие для тестов $n \leq 2$ будут набирать не менее 40 баллов.

Задача В. Погоня в метро

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Это интерактивная задача. Параллельно с выполнением вашего решения жюри запускает проверяющую программу с которой вы общаетесь сообщениями через стандартный ввод и вывод. Подробнее о протоколе взаимодействия написано ниже. Также в конце условия вы можете посмотреть корректные примеры взаимодействия с проверяющей программой на разных языках программирования.

В прекрасной Метрополии будущего необходимость в машинистах, управляющих поездами метро, отпала. Благодаря развитию технологий, их заменил искусственный интеллект (ИИ). К сожалению, в один прекрасный день опасения писателей-фантастов сбылись — ИИ взбунтовался, и теперь где-то в метро ездит неуправляемый поезд. Страшно представить, чем это грозит городской транспортной системе! Ваша задача состоит в том, чтобы найти поезд в сложной системе метро и остановить неуправляемый ИИ.

В целях безопасности все остальные поезда были отправлены в депо, а все ветки, кроме той, на которой находится неконтролируемый поезд, были перекрыты, поэтому на данный момент метро Метрополии представляет из себя одну ветку (обычную прямую ветку без самопересечений) из n станций, последовательно пронумерованных от 1 до n , ровно на одной из которых находится поезд. Для поимки неуправляемого поезда вам требуется определить номер этой станции, после чего на путях будут установлены искусственные заграждения и поезд будет пойман.

Для определения нужной станции диспетчер Сара одолжила вам устройство, позволяющее вам выбрать произвольные числа l и r ($l \leq r$), после чего оно проверит, верно ли, что поезд находится на станции с номером между l и r . К сожалению, для перезарядки устройства требуется k минут (и вы используете его как только перезарядка завершается), поэтому между двумя применениями поезд может перебраться из той станции, где он сейчас находится, в любую станцию с номером отличающимся не более чем на k . Формально, если при некотором применении устройства поезд находился на станции x , то при следующем применении он может находиться на любой станции y , такой что $\max(1, x - k) \leq y \leq \min(n, x + k)$. При этом поезд не знает, что вы пытаетесь его поймать и совершает все перемещения согласно некоторому заранее составленному им плану.

В процессе изучения устройства вы выяснили, что оно было сделано очень давно, и сможет выдержать не более чем q использований, после чего оно сломается, а ваша задача будет считаться проваленной.

Сможете ли вы найти станцию, на которой находится поезд, за не более чем q использований устройства?

Протокол взаимодействия

При запуске решения на вход подаются три целых числа n ($1 \leq n \leq 10^{18}$), k ($0 \leq k \leq 10$) и q ($q \leq 15\,000$).

Чтобы использовать устройство, вы должны вывести через пробел два числа l и r

($1 \leq l \leq r \leq n$). В ответ на это вы получите либо строку «Yes», если между станциями с номерами l и r находится поезд, либо строку «No» иначе. Если $l = r$ и вы получили ответ «Yes», это значит, что вы точно определили станцию, на которой находится поезд, и ваша программа после этого должна немедленно завершиться.

Если ваша программа за q запросов не смогла точно определить станцию, на которой находится поезд, программа должна также немедленно завершиться, в противном случае вердикт тестирующей системы **может быть любым**.

После каждого запроса необходимо вывести перевод строки и сбросить буфер вывода — для этого используйте команды `flush(output)` на языке Паскаль или Delphi, `fflush(stdout)` или `cout.flush()` в C/C++, `sys.stdout.flush()` на языке Python, `System.out.flush()` на языке Java.

В точности соблюдайте формат взаимодействия с системой.

Пример

стандартный ввод	стандартный вывод
10 2 15000	3 4
Yes	3 3
No	2 2
Yes	

Система оценки

Тесты к этой задаче состоят из пяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп, от которых она зависит.

Группа	Баллы	Дополнительные ограничения			Комментарий	Необходимые группы
		n	k	q		
0	0	$n = 10$	$k = 2$	$q = 15\,000$	Тест из условия.	—
1	20	$n \leq 10^{18}$	$k = 0$	$q = 15\,000$		—
2	20	$n \leq 10^3$	$k \leq 10$	$q = 15\,000$		0
3	30	$n \leq 10^6$	$k \leq 10$	$q = 15\,000$		0, 2
4	30	$n \leq 10^{18}$	$k \leq 10$	$q = 4500$		0, 1, 2, 3

Замечание

В первом тесте поезд изначально находился на станции 3, после первого использования устройства переместился на станцию 2, а после второго — остался на месте.

Ниже вы найдёте примеры программ на языках **C++**, **Java**, **Python 2** и **Python 3**. Эти программы набирают 0 баллов, но иллюстрируют корректное взаимодействие с проверяющей программой и сброс буфера вывода.

C++

```
#include <iostream>

using namespace std;

long long n, k, q;

bool ask(long long l, int r) {
    // using endl forces output buffer flush
    cout << l << ' ' << r << endl;
    string ans;
    cin >> ans;
    return ans == "Yes";
}

main() {
    cin >> n >> k >> q;
    while (q--) {
        bool res = ask(1, 1);
        if (res) {
            return 0;
        }
    }
}
```

Java

```
import java.util.Scanner;

public class Main {

    static Scanner in;

    static boolean ask(long l, long r) {
        System.out.print(l);
        System.out.print(" ");
        System.out.print(r);
        System.out.println();
        System.out.flush();
        String ans = in.next();
        return ans.equals("Yes");
    }

    public static void main(String[] args) {
        in = new Scanner(System.in);
        long n = in.nextLong();
        long k = in.nextLong();
        long q = in.nextLong();
        while (q > 0) {
            q--;
            boolean res = ask(1, 1);
            if (res) {
                return;
            }
        }
    }
}
```

Python 2

```
import sys

def ask(l, r):
    print str(l) + ' ' + str(r)
    sys.stdout.flush()
    ans = raw_input()
    return ans == "Yes"

n, k, q = map(int, raw_input().split())
while q > 0:
    q -= 1
    res = ask(1, 1)
    if res:
        exit(0)
```

Python 3

```
import sys

def ask(l, r):
    print(l, r)
    sys.stdout.flush()
    ans = input()
    return ans == "Yes"

n, k, q = map(int, input().split())
while q > 0:
    q -= 1
    res = ask(1, 1)
    if res:
        exit()
```

Задача С. Безопасность сети

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

К компьютерной сети Метрополии подключены n серверов, каждому из которых назначен некоторый ключ шифрования, выбранный в диапазоне от 0 до $2^k - 1$ включительно. Обозначим через c_i ключ шифрования, назначенный i -му серверу. Также некоторые m пар серверов напрямую соединены каналом передачи данных. Из-за особенностей используемых алгоритмов шифрования канал передачи данных безопасен, только если два соединённых этим каналом сервера используют **различные** ключи шифрования. Изначально ключи шифрования выбраны таким образом, что все каналы передачи данных безопасны.

Вам стало известно, что по интернету активно распространяется новый вирус, который при заражении им какого-либо сервера Метрополии изменяет его ключ шифрования. А именно, в теле вируса хранится некоторое неизвестное вам число x , выбранное в том же диапазоне, и если вирусом заразится сервер i , то его ключ шифрования изменится с c_i на $c_i \oplus x$, где \oplus означает операцию побитового исключающего «ИЛИ» (XOR).

К сожалению, вы не знаете ни число x , ни какие именно серверы Метрополии будут заражены опасным вирусом, поэтому решили посчитать количество вариантов, при которых безопасность каналов нарушена не будет. Формально, вам требуется найти количество пар (A, x) , где A — некоторое подмножество серверов (возможно пустое), а x — возможное значение внутреннего параметра вируса в диапазоне от 0 до $2^k - 1$ включительно, таких что если вирус с параметром x заразит компьютеры из подмножества A , а остальные компьютеры заражены не будут, то все каналы передачи данных останутся безопасными. Поскольку эта величина может быть достаточно большой, найдите остаток от её деления на $10^9 + 7$.

Формат входных данных

В первой строке входных данных записаны три целых числа n , m и k ($1 \leq n \leq 500\,000$, $0 \leq m \leq \min(\frac{n(n-1)}{2}, 500\,000)$, $0 \leq k \leq 60$) — количество серверов, количество пар серверов, непосредственно соединённых каналом передачи данных, и параметр k , определяющий диапазон возможных значений ключа шифрования.

Следующая строка содержит n целых чисел c_i ($0 \leq c_i \leq 2^k - 1$), i -е из которых соответствует ключу шифрования, используемому на i -м сервере.

Следующие m строк содержат по два целых числа u_i и v_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$), означающих, что эти два сервера соединены каналом передачи данных. Гарантируется, что любая пара серверов встречается в этом списке не более одного раза.

Формат выходных данных

В единственной строке выходного файла выведите одно число — количество удачных исходов заражения какого-то подмножества компьютеров вирусом по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
4 4 2 0 1 0 1 1 2 2 3 3 4 4 1	50
4 5 3 7 1 7 2 1 2 2 3 3 4 4 1 2 4	96

Замечание

Определим операцию побитового ИЛИ (XOR). Пусть даны два целых неотрицательных числа x и y , рассмотрим их двоичные записи (возможно с ведущими нулями): $x_k \dots x_2 x_1 x_0$ и $y_k \dots y_2 y_1 y_0$. Здесь x_i это i -й бит числа x , а y_i это i -й бит числа y . Пусть $r = x \oplus y$ — результат операции XOR над числами x и y . Тогда двоичной записью r будет $r_k \dots r_2 r_1 r_0$, где:

$$r_i = \begin{cases} 1, & \text{если } x_i \neq y_i \\ 0, & \text{если } x_i = y_i \end{cases}$$

Система оценки

Тесты к этой задаче состоят из четырех групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп.

Группа	Баллы	Дополнительные ограничения			Комментарий
		n	m	k	
0	0	–	–	–	Тесты из условия.
1	10	$n \leq 10$	$m \leq 45$	$k \leq 10$	
2	20	$n \leq 5000$	$m \leq 5000$	$k \leq 14$	
3	30	$n \leq 5000$	$m \leq 5000$	–	
4	40	–	–	–	

Задача D. Вам дано дерево

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

Деревом называется связный неориентированный граф, в котором между любыми двумя вершинами существует ровно один простой путь. Будем говорить, что множество простых путей является k -допустимым, если каждая вершина дерева принадлежит не более чем одному из этих путей (включая концевые вершины) и каждый путь состоит ровно из k вершин.

Вам дано дерево на n вершинах. Для каждого k от 1 до n включительно найдите максимально возможную мощность k -допустимого множества простых путей.

Формат входных данных

Первая строка содержит одно целое число n ($2 \leq n \leq 500\,000$) — количество вершин в дереве.

Следующие $n - 1$ строк задают дерево, каждая из них содержит два целых числа v и u ($1 \leq v, u \leq n$) — концы соответствующего ребра.

Гарантируется, что заданный граф является деревом.

Формат выходных данных

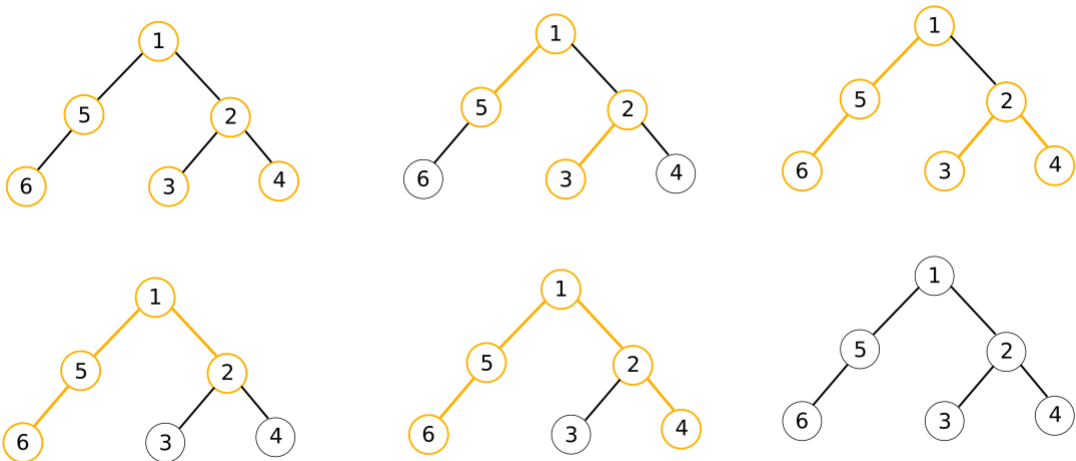
Выведите n чисел, i -е из которых равно максимально возможной мощности i -допустимого множества путей.

Примеры

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

Замечание

Один из способов достичь наибольшего количества путей во втором примере изображён на следующей картинке:



Система оценки

Тесты к этой задаче состоят из 14 групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов всех **предыдущих** групп.

Группа	Баллы	Дополнительные ограничения	Комментарий
		n	
0	0	—	Тесты из условия
1	7	$n \leq 50$	—
2	7	$n \leq 500$	—
3	7	$n \leq 3000$	—
4	7	$n \leq 10\,000$	—
5	8	$n \leq 25\,000$	—
6	6	$n \leq 50\,000$	—
7	9	$n \leq 75\,000$	—
8	6	$n \leq 100\,000$	—
9	7	$n \leq 150\,000$	—
10	7	$n \leq 200\,000$	—
11	7	$n \leq 250\,000$	—
12	8	$n \leq 300\,000$	—
13	7	$n \leq 400\,000$	—
14	7	$n \leq 500\,000$	—