

Задача А. Разрезание графа

Имя входного файла: `cutting.in`
Имя выходного файла: `cutting.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан неориентированный граф. Над ним производят операции следующих двух типов:

- `cut` — разрезать граф, то есть удалить из него ребро;
- `ask` — проверить, лежат ли две вершины графа в одной компоненте связности.

Известно, что после выполнения всех операций типа `cut` рёбер в графе не осталось. Найдите результат выполнения каждой из операций типа `ask`.

Формат входного файла

Первая строка входного файла содержит три целых числа, разделённые пробелами — количество вершин графа n , количество рёбер m и количество операций k ($1 \leq n \leq 50\,000$, $0 \leq m \leq 100\,000$, $m \leq k \leq 150\,000$). Следующие m строк задают рёбра графа; i -ая из этих строк содержит два числа u_i и v_i ($1 \leq u_i, v_i \leq n$), разделённые пробелами — номера концов i -го ребра. Вершины нумеруются с единицы; граф не содержит петель и кратных рёбер. Далее следуют k строк, описывающих операции. Операция типа `cut` задаётся строкой “`cut u v` ” ($1 \leq u, v \leq n$), которая означает, что из графа удаляют ребро между вершинами u и v . Операция типа `ask` задаётся строкой “`ask u v` ” ($1 \leq u, v \leq n$), которая означает, что необходимо узнать, лежат ли в данный момент вершины u и v в одной компоненте связности. Гарантируется, что каждое ребро графа встретится в операциях типа `cut` ровно один раз.

Формат выходного файла

Для каждой операции `ask` во входном файле выведите на отдельной строке слово “YES”, если две указанные вершины лежат в одной компоненте связности, и “NO” в противном случае. Порядок ответов должен соответствовать порядку операций `ask` во входном файле.

Примеры

<code>cutting.in</code>	<code>cutting.out</code>
3 3 7	YES
1 2	YES
2 3	NO
3 1	NO
ask 3 3	
cut 1 2	
ask 1 2	
cut 1 3	
ask 2 1	
cut 2 3	
ask 3 1	

Задача В. Хорошие дни

Имя входного файла: `feelgood.in`
Имя выходного файла: `feelgood.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Билл разрабатывает новую математическую теорию, описывающую человеческие эмоции. Его последние исследования посвящены изучению того, насколько хорошие и плохие дни влияют на воспоминания людей о различных периодах жизни.

Недавно Билл придумал методику, которая описывает, насколько хорошим или плохим был день человеческой жизни с помощью сопоставления дню некоторого неотрицательно-го целого числа. Билл называет это число *эмоциональной значимостью* этого дня. Чем больше это число, тем лучше этот день. Билл полагает, что значимость некоторого периода человеческой жизни равна сумме эмоциональных значимостей каждого из дней периода, помноженной на минимум эмоциональных значимостей дней этого периода. Эта методика отражает то, что период, который в среднем может быть весьма неплох, бывает испорчен одним плохим днем.

Теперь Билл хочет проанализировать свою собственную жизнь и найти в ней период максимальной значимости. Помогите ему это сделать.

Формат входного файла

Первая строка входного файла содержит число n — количество дней в жизни Билла, которые он хочет исследовать ($1 \leq n \leq 100\,000$). Оставшаяся часть файла содержит n целых чисел a_1, a_2, \dots, a_n , все в пределах от 0 до 10^6 — эмоциональные значимости дней. Числа во входном файле разделяются пробелами и переводами строки.

Формат выходного файла

В первой строке выходного файла выведите максимальную значимость периода жизни Билла. Во второй строке выведите два числа l и r , означающие, что значимость периода с l -го по r -й день (включительно) в жизни Билла была максимально возможной.

Примеры

<code>feelgood.in</code>	<code>feelgood.out</code>
6	60
3 1 6 4 5 2	3 5

Задача С. Цветные волшебники

Имя входного файла: `magic.in`
Имя выходного файла: `magic.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Сказочная страна представляет собой множество городов, соединенных дорогами с двухсторонним движением. Причем из любого города страны можно добраться в любой

другой город либо непосредственно, либо через другие города. Известно, что в сказочной стране не существует дорог, соединяющих город сам с собой и между любыми двумя разными городами, существует не более одной дороги.

В сказочной стране живут желтый и синий волшебники. Желтый волшебник, пройдя по дороге, перекрашивает ее в желтый цвет, синий — в синий. Как известно, при наложении желтой краски на синюю, либо синей краски на желтую, краски смешиваются и превращаются в краску зеленого цвета, который является самым нелюбимым цветом обоих волшебников.

В этом году в столице страны (городе f) проводится конференция волшебников. Поэтому желтый и синий волшебники хотят узнать, какое минимальное количество дорог им придется перекрасить в зеленый цвет, чтобы добраться в столицу. Изначально все дороги не покрашены.

Начальное положение желтого и синего волшебников заранее не известно. Поэтому необходимо решить данную задачу для k возможных случаев их начальных расположений.

Формат входного файла

Первая строка входного файла содержит целые числа: n ($1 \leq n \leq 100\,000$) и m ($1 \leq m \leq 500\,000$) — количество городов и дорог в волшебной стране соответственно. Третья строка содержит одно целое число f ($1 \leq f \leq n$) — номер города, являющегося столицей сказочной страны. В следующих m строках, находится описание дорог страны. В этих m строк записано по два целых числа a_i и b_i , означающих, что существует дорога, соединяющая города a_i и b_i . Следующая строка содержит целое число k ($1 \leq k \leq 100\,000$) — количество возможных начальных расположений волшебников. Далее следуют k строк, каждая из которых содержит два целых числа — номера городов, в которых изначально находится желтый и синий волшебники соответственно.

Формат выходного файла

Для каждого из k случаев, ваша программа должна вывести в выходной минимальное количество дорог, которое придется покрасить в зеленый цвет волшебникам для того, чтобы добраться в столицу.

Примеры

magic.in	magic.out
6 6	1
1	2
1 2	
2 3	
3 4	
4 2	
4 5	
3 6	
2	
5 6	
6 6	

Задача D. Представление числа

Имя входного файла:	number.in
Имя выходного файла:	number.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Учительница математики попросила школьников составить арифметическое выражение, так чтобы его значение было равно данному числу N , и записать его в тетради. В выражении могут быть использованы натуральные числа, не превосходящие K , операции сложения и умножения, а также скобки. Петя очень не любит писать, и хочет придумать выражение, содержащее как можно меньше символов. Напишите программу, которая поможет ему в этом.

Формат входного файла

В первой строке входного файла записаны два натуральных числа: N ($1 \leq N \leq 10000$) — значение выражения и K ($1 \leq K \leq 10000$) — наибольшее число, которое разрешается использовать в выражении.

Формат выходного файла

В единственной строке выходного файла выведите выражение с данным значением, записывающееся наименьшим возможным количеством символов. Если решений несколько, выведите любое из них.

Примеры

number.in	number.out
7 3	3+3+1
15 20	15
96 1	(1+1+1+1)*(1+1+1+1)*(1+1+1)*(1+1)

Note

При подсчете длины выражения учитываются все символы: цифры, знаки операций, скобки.