

## А. Технокубок Огня

1 секунда, 256 мегабайт

Все знают, что в скоро пройдет турнир, известный как «Турнир  $m$  программистов». В турнире примут участие  $m$  школ, из каждой школы примет участие один ученик.

Всего в этих школах учатся  $n$  школьников. Перед началом турнира имена всех школьников, а также названия их школ закладываются в Технокубок Огня, являющийся символом турнира, а затем он выбирает по одному ученику из каждой школы для участия в турнире. Известно, что Технокубок выберет сильнейшего участника из каждой школы.

Аркадий — хакер, он хочет, чтобы среди учеников, выбранных кубком, были  $k$  избранных. К сожалению, не все из них сильнейшие в своих школах, поэтому Аркадий может выдумать несколько новых школ и изменить некоторые названия школ, заложенные в Технокубок, на эти новые выдуманные названия. Нельзя использовать каждое из выдуманных названий более, чем один раз. В таком случае Технокубок выберет сильнейшего ученика в этих школах тоже.

Вам известна информация о силах учеников, а также то, в какой школе учится каждый из них. Вычислите, какое минимальное число школ должен выдумать Аркадий, чтобы  $k$  избранных учеников были среди выбранных кубком?

### Входные данные

Первая строка содержит три целых числа  $n$ ,  $m$  и  $k$  ( $1 \leq n \leq 100$ ,  $1 \leq m, k \leq n$ ) — общее число школьников, число школ и число избранных учеников.

Вторая строка содержит  $n$  **различных** целых чисел  $p_1, p_2, \dots, p_n$  ( $1 \leq p_i \leq n$ ), где  $p_i$  обозначает силу  $i$ -го ученика. Чем сильнее ученик, тем больше у него сила.

Третья строка содержит  $n$  целых чисел  $s_1, s_2, \dots, s_n$  ( $1 \leq s_i \leq m$ ), где  $s_i$  обозначает школу  $i$ -го ученика. В каждой школе учится хотя бы один ученик.

Четвертая строка содержит  $k$  **различных** целых чисел  $c_1, c_2, \dots, c_k$  ( $1 \leq c_i \leq n$ ) — номера избранных учеников.

### Выходные данные

Выведите одно целое число — минимальное количество школ, которые должен выдумать Аркадий, чтобы  $k$  избранных учеников были выбраны Технокубком.

входные данные
7 3 1 1 5 3 4 6 7 2 1 3 1 2 1 2 3 3
выходные данные
1

входные данные
8 4 4 1 2 3 4 5 6 7 8 4 3 2 1 4 3 2 1 3 4 5 6
выходные данные
2

В первом примере лишь один избранный ученик под номером 3. Его сила равна 3, но в той же школе 1 есть ученик под номером 5 с силой 6. Значит Технокубок не выберет избранного, если ничего не менять. Если же у избранного ученика выдумать новую школу (положим у нее номер 4), то Технокубок выберет учеников с номерами 2 (сильнейший в школе 3), 5 (сильнейший в школе 1), 6 (сильнейший в школе 2) и 3 (сильнейший в школе 4).

Во втором примере можно у ученика 3 изменить школу на выдуманную 5, а у ученика 4 изменить школу на выдуманную 6, тогда Технокубок выберет школьников 8, 7, 6, 5, 3 и 4.

## В. Системное тестирование

1 секунда, 256 мегабайт

Вася любит писать раунды на Codeforces. Когда раунд заканчивается и начинается системное тестирование, Вася внимательно следит за всеми посылками.

Всего есть  $n$  решений,  $i$ -е из них нужно протестировать на  $a_i$  тестах, тестирование на одном тесте занимает 1 секунду. Решения поступают на тестирование по очереди в порядке от 1 до  $n$ . Всего на серверах запущено  $k$  отдельных тестирующих процессов, которые работают одновременно. Каждый из них может обрабатывать одновременно только одно решение.

В любой момент времени  $t$ , когда какой-то из тестирующих процессов не обрабатывает ни одно решение, он берет очередное решение из очереди и тестирует его в порядке увеличения номеров тестов. Пусть номер этого решения —  $i$ , тогда оно будет тестироваться на первом тесте с момента времени  $t$  по момент времени  $t + 1$ , затем на втором тесте до момента времени  $t + 2$  и так далее. Полностью это решение будет протестировано в момент времени  $t + a_i$ , и этот процесс сразу начнет тестировать другое решение.

Рассмотрим некоторый момент времени, пусть сейчас полностью протестировано  $m$  решений. Тогда на странице с посылками отображается надпись «Системное тестирование:  $d\%$ », где  $d$  вычисляется по формуле

$$d = \text{round} \left( 100 \cdot \frac{m}{n} \right),$$

где  $\text{round}(x) = \lfloor x + 0.5 \rfloor$  есть функция округления числа до ближайшего целого.

Вася называет решение *интересным*, если существовал момент времени (возможно, не целочисленный), когда это решение исполнялось на некотором тесте  $q$ , и надпись сбоку гласила «Системное тестирование:  $q\%$ ». Найдите число интересных решений.

Обратите внимание, что если несколько процессов одновременно будут свободны и захотят взять первое решение из очереди (например, в начальный момент времени), то не имеет значения, в каком порядке они будут получать доступ к очереди решений.

### Входные данные

Первая строка содержит два целых числа  $n$  и  $k$  ( $1 \leq n \leq 1000$ ,  $1 \leq k \leq 100$ ) — число отправленных решений и количество тестирующих процессов, соответственно.

Вторая строка содержит  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 150$ ), где  $a_i$  равно числу тестов, на которых будет тестироваться  $i$ -е решение.

### Выходные данные

В единственной строке выведите число интересных решений.

<b>входные данные</b>
2 2 49 100
<b>выходные данные</b>
1

<b>ВХОДНЫЕ ДАННЫЕ</b>
4 2 32 100 33 1
<b>ВЫХОДНЫЕ ДАННЫЕ</b>
2

<b>ВХОДНЫЕ ДАННЫЕ</b>
14 5 48 19 6 9 50 20 3 42 38 43 36 21 44 6
<b>ВЫХОДНЫЕ ДАННЫЕ</b>
5

Рассмотрим первый пример. В момент времени 0 оба решения начинают тестироваться. В момент времени 49 первое решение полностью протестировано, поэтому в момент времени 49.5 второе решение тестировалось на тесте 50, а надпись сбоку гласила «Системное тестирование: 50%» (так как протестировано одно решение из двух). Значит, это интересное решение.

Рассмотрим второй пример. В момент времени 0 начинают тестироваться первое и второе решения. В момент времени 32 первое решение полностью протестировано, начинает тестироваться третье решение, надпись сбоку гласит «Системное тестирование: 25%». В момент времени  $32 + 24.5 = 56.5$  третье решение тестируется на 25-м тесте, а надпись сбоку все еще не изменилась, значит, это решение — интересное. После этого третье решение будет полностью протестировано в момент времени  $32 + 33 = 65$ , четверное решение будет полностью протестировано в момент времени  $65 + 1 = 66$ . Надпись сбоку будет гласить «Системное тестирование: 75%», и в момент времени 74.5 второе решение будет тестироваться на 75-м тесте. Значит, оно тоже интересное. Значит всего есть два интересных решения.

## С. Диана и Лиана

2 секунды, 256 мегабайт

В славном городе Короткоречном в первый выходной весны традиционно проходит праздник цветов. Во время этого праздника жители носят традиционные венки. В традиционном венке ровно  $k$  цветов.

Заготовки для венков на всех  $n$  жителей города Короткоречный нарезают из самой длинной лианы с цветами, которая выросла в городе в этом году. Лиана — это последовательность  $a_1, a_2, \dots, a_m$ , где  $a_i$  — целое число, обозначающее вид цветка на позиции  $i$ . В этом году короткореченцам повезло: лиана выросла такая длинная, что  $m \geq n \cdot k$ , а это значит, что каждому жителю города достанется по венку.

Очень скоро лиану вставят в специальную машину, которая нарежет её на заготовки для венков. Машина работает просто: отрезает от начала лианы первые  $k$  цветков, затем вторые  $k$  цветков и так далее. Каждый получившийся отрезок из  $k$  цветов называется заготовкой. Машина продолжает работу, пока оставшаяся части лианы содержит хотя бы  $k$  цветов.

Диана нашла схему для плетения самого красивого венка. Для плетения такого венка среди  $k$  цветов должны быть цветы видов  $b_1, b_2, \dots, b_s$ , а остальные могут быть любыми. Если некоторый вид встречается в этой последовательности несколько раз, то цветов этого вида должно быть хотя бы столько, сколько раз этот вид встречается в последовательности. Порядок, в котором эти цветы будут в заготовке, не важен.

Перед тем как лиану вставят в машину для нарезки, у Дианы есть возможность оторвать от неё часть цветов. Отрывать можно любые цветы, в том числе из середины лианы, при этом лиана не распадается на части. Если Диана оторвёт слишком много цветов, то кому-то из жителей не достанется венок. Можно ли оторвать от лианы часть цветов так, чтобы хотя бы одна из заготовок соответствовала схеме Дианы, но при этом машина по-прежнему могла нарезать хотя бы  $n$  заготовок?

## Входные данные

В первой строке находятся четыре целых числа  $m, k, n$  и  $s$  ( $1 \leq n, k, m \leq 5 \cdot 10^5, k \cdot n \leq m, 1 \leq s \leq k$ ): количество цветов на лиане, количество цветов в одном венке, количество жителей и длина последовательности видов цветов в схеме Дианы, соответственно.

Во второй строке находятся  $m$  целых чисел  $a_1, a_2, \dots, a_m$  ( $1 \leq a_i \leq 5 \cdot 10^5$ ) — виды цветов на лиане.

В третьей строке находятся  $s$  целых чисел  $b_1, b_2, \dots, b_s$  ( $1 \leq b_i \leq 5 \cdot 10^5$ ) — последовательность в схеме Дианы.

### Выходные данные

Если невозможно оторвать часть цветов от лианы так, чтобы после нарезки получилось хотя бы  $n$  заготовок для венков, и хотя бы одна из них соответствовала схеме Дианы, то выведите  $-1$ .

Иначе в первой строке выведите одно целое число  $d$  — количество цветов, которые нужно оторвать.

В следующей строке выведите  $d$  различных целых чисел — номера цветов, которые нужно оторвать.

Если есть несколько решений, выведите любое.

входные данные
7 3 2 2 1 2 3 3 2 1 2 2 2
выходные данные
1 4

входные данные
13 4 3 3 3 2 6 4 1 4 4 7 1 3 3 2 4 4 3 4

выходные данные
-1

входные данные
13 4 1 3 3 2 6 4 1 4 4 7 1 3 3 2 4 4 3 4
выходные данные
9 1 2 3 4 5 9 11 12 13

В первом примере если не отрывать ни одного цветка, то машина выдаст две заготовки, виды цветов в которых будут  $[1, 2, 3]$  и  $[3, 2, 1]$ . Видно, что это Диане не подходит. Если же убрать цветок с номером 4, то машина выдаст заготовки  $[1, 2, 3]$  и  $[2, 1, 2]$ . Вторая заготовка подойдет Диане.

Во втором примере нельзя так оторвать цветки, чтобы всем достался венок, а у Дианы была бы подходящая для самого красивого венка заготовка.

В третьем примере Диана — единственный житель города, поэтому она может, например, оторвать все цветы, кроме нужных.

## D. Сжать строку

3 секунды, 256 мегабайт

Имеется строка  $s$  из  $n$  символов. Каждый символ — маленькая латинская буква. Вам нужно сжать эту строку, потратив как можно меньше денег.

Для сжатия вы должны представить строку  $s$  в виде конкатенации некоторого количества непустых строк:  $s = t_1 t_2 \dots t_k$ . Далее,  $i$ -ю из этих строк нужно сжать одним из следующих двух способов:

- если  $|t_i| = 1$ , то есть текущая строка состоит всего из одного символа, то можно просто заплатить  $a$  монет;

- если  $t_i$  является подстрокой строки  $t_1 t_2 \dots t_{i-1}$ , то можно заплатить  $b$  монет.

Строка  $x$  является подстрокой строки  $y$ , если  $x$  может быть получена из  $y$  удалением нескольких (возможно, ни одного или всех) символов из начала и нескольких (возможно, ни одного или всех) символов из конца.

Итак, требуется по данной строке  $s$  определить наименьшее возможное количество монет, которого хватит для некоторого сжатия  $s$ .

### Входные данные

В первой строке находятся три целых положительных числа, разделённых пробелами:  $n$ ,  $a$  и  $b$  ( $1 \leq n, a, b \leq 5000$ ) — длина строки, стоимость сжатия строки из одного символа и стоимость сжатия подстроки, которая встречалась ранее, соответственно.

Во второй строке находится строка  $s$ , состоящая из  $n$  строчных латинских букв.

### Выходные данные

Выведите единственное число — наименьшее возможное количество монет, необходимое для сжатия.

<b>входные данные</b>
3 3 1 aba
<b>выходные данные</b>
7

<b>входные данные</b>
4 1 1 abcd
<b>выходные данные</b>
4

<b>входные данные</b>
4 10 1 aaaa
<b>выходные данные</b>
12

В первом тесте можно положить  $t_1 = \text{«a»}$ ,  $t_2 = \text{«b»}$ ,  $t_3 = \text{«a»}$  и заплатить  $3 + 3 + 1 = 7$  монет, поскольку  $t_3$  является подстрокой строки  $t_1 t_2$ .

Во втором тестовом примере нужно сжать каждую букву независимо.

В третьем тесте можно положить  $t_1 = t_2 = \text{«a»}$ ,  $t_3 = \text{«aa»}$  и заплатить  $10 + 1 + 1 = 12$  монет, поскольку  $t_2$  — подстрока  $t_1$ , а  $t_3$  — подстрока  $t_1 t_2$ .

## Е. Случай в казино

1 секунда, 256 мегабайт

Игрок пришёл в казино и нашёл игровой автомат, в котором всё зависит только от его действий. Игра устроена следующим образом.

На экране автомата написано число  $a$ . Положив монету в автомат, можно взять любые две соседние цифры числа на экране и либо прибавить к обеим 1, либо вычесть 1 из обеих. После этой операции все цифры по-прежнему должны принимать значения от 0 до 9, а также у числа не должно быть лидирующих нулей. Иными словами, нельзя вычитать 1 из 0, прибавлять 1 к 9, а также вычитать 1 из цифры 1, стоящей в старшем разряде числа. Как только на экране будет число  $b$ , игрок выиграет джекпот. Числа  $a$  и  $b$  имеют одинаковую длину.

Помогите игроку определить наименьшее число монет, которые нужно потратить, чтобы выиграть джекпот, и как именно нужно играть.

### Входные данные

Первая строка содержит одно целое число  $n$  ( $2 \leq n \leq 10^5$ ) — длину чисел  $a$  и  $b$ .

Следующие 2 строки содержат числа  $a$  и  $b$ , по одному на каждой строке ( $10^{n-1} \leq a, b < 10^n$ ).

### Выходные данные

Если выиграть джекпот невозможно, в единственной строке выведите  $-1$ .

В противном случае в первой строке выведите наименьшее количество монет  $c$ , которые нужно потратить игроку, чтобы выиграть.

Затем выведите  $\min(c, 10^5)$  строк. В  $i$ -й из них выведите два числа  $d_i$  и  $s_i$  ( $1 \leq d_i \leq n - 1$ ,  $s_i = \pm 1$ ), означающие, что на  $i$ -м ходу игрок прибавит  $s_i$  к цифрам, находящимся на местах  $d_i$  и  $(d_i + 1)$  слева (то есть,  $d_i = 1$  означает, что меняются два старших разряда, а  $d_i = n - 1$  — что два младших).

Обратите внимание, что ответ может быть очень большим, и в случае, если  $c > 10^5$ , Вам надо вывести лишь первые  $10^5$  ходов. Ответ считается правильным, если можно закончить выведенные ходы таким образом, чтобы выиграть джекпот за наименьшее число ходов. В частности, если ответов несколько, можно вывести любой из них.

Входные данные
3 223 322
Выходные данные
2 1 1 2 -1

Входные данные
2 20 42
Выходные данные
2 1 1 1 1

Входные данные
2 35 44
Выходные данные
-1

В первом примере можно сначала сделать операцию  $+1$  на первых двух цифрах, то есть превратить число **223** в **333**, а затем сделать операцию  $-1$  на последних двух цифрах, то есть превратить **333** в **322**.

Можно проделать эти две операции в обратном порядке, и это тоже будет правильным ответом.

Можно показать, что в последнем примере превратить 35 в 44 нельзя.

## Г. Дерево власти

2 секунды, 256 мегабайт

Дано подвешенное дерево из  $n$  вершин, корнем дерева является вершина 1. Каждая вершина имеет некоторую неотрицательную цену. Листом дерева называется вершина, не являющаяся корнем, степень которой равна 1.

Аркадий и Василий играют в странную игру на этом дереве. Игра состоит из трех этапов. На первом этапе Аркадий покупает некоторое непустое подмножество вершин дерева. На втором этапе Василий расставляет в листьях дерева некоторые целые числа. На третьем этапе Аркадий может несколько (возможно, ни одного) раз выполнить следующую операцию: выбрать некоторую купленную на первом этапе вершину  $v$  и некоторое целое число  $x$ , а затем прибавить ко всем числам, расставленным в листьях поддерева вершины  $v$ , число  $x$ . Число  $x$  может быть положительным, отрицательным или нулем.

Лист  $a$  находится в поддереве вершины  $b$ , если простой путь от  $a$  до корня проходит через  $b$ .

Задача Аркадия — сделать так, чтобы числа во всех листьях оказались равными нулю. Какую минимальную сумму  $s$  он должен заплатить на первом этапе, чтобы гарантировать себе победу независимо от того, как Василий расставит числа на втором этапе? Кроме этого, найдите все такие вершины, что существует оптимальный (то есть со стоимостью  $s$ ) набор вершин, включающий заданную, купив который Аркадий может гарантировать себе победу.

### Входные данные

Первая строка содержит одно целое число  $n$  ( $2 \leq n \leq 200\,000$ ) — число вершин в дереве.

Вторая строка содержит  $n$  целых чисел  $c_1, c_2, \dots, c_n$  ( $0 \leq c_i \leq 10^9$ ), где  $c_i$  — цена  $i$ -й вершины.

В следующих  $n - 1$  строках содержатся по два целых числа  $a$  и  $b$  ( $1 \leq a, b \leq n$ ), описывающих ребра дерева.

### Выходные данные

В первой строке выведите два целых числа: минимальную стоимость  $s$ , которую должен заплатить Аркадий, чтобы гарантировать победу, и количество вершин  $k$ , которые содержатся хотя бы в одном оптимальном множестве.

Во второй строке выведите  $k$  различных целых чисел в **возрастающем порядке** — номера вершин, которые содержатся хотя бы в одном оптимальном множестве.

Входные данные
5 5 1 3 2 1 1 2 2 3 2 4 1 5
Выходные данные
4 3 2 4 5

Входные данные
3 1 1 1 1 2 1 3
Выходные данные
2 3 1 2 3

Во втором примере любое множество из двух вершин является оптимальным. Поэтому каждая вершина лежит хотя бы в одном оптимальном множестве.

## Г. Тот самый Мюнхгаузен

1 секунда, 256 мегабайт

Дано натуральное число  $a$ . Барон Мюнхгаузен утверждает, что знает такое натуральное число  $n$ , что при умножении числа  $n$  на  $a$  его сумма цифр уменьшается в  $a$  раз. Иными словами,  $S(an) = S(n)/a$ , где  $S(x)$  — сумма цифр числа  $x$ . Может ли барон говорить правду?

### Входные данные

Единственная строка содержит одно целое число  $a$  ( $2 \leq a \leq 10^3$ ).

### Выходные данные

Если барон лжёт, и такого числа  $n$  не существует, выведите  $-1$ .

В противном случае выведите любое натуральное число  $n$ , обладающее нужным свойством. Длина выводимого числа не должна превышать  $5 \cdot 10^5$ . Можно показать, что при данных ограничениях ответ либо не существует, либо существует ответ, длина которого не превышает  $5 \cdot 10^5$ .

<b>входные данные</b>
2
<b>выходные данные</b>
6

<b>входные данные</b>
3
<b>выходные данные</b>
6669

<b>входные данные</b>
10
<b>выходные данные</b>
-1

## Н. Секретные письма

2 секунды, 512 мегабайт

Винни-Пух и Пятачок договорились писать друг другу секретные письма про самые важные события за день. В день происходит  $n$  событий: в момент времени  $t_i$  с участником  $p_i$  ( $p_i$  равно либо  $\bar{W}$ , либо  $\bar{P}$ , что обозначает Винни-Пуха и Пятачка соответственно) что-то происходит, и он должен немедленно написать об этом другому, пока всё не забыл. Написанное сообщение надо сразу отправить одним из двух способов:

- Попросить Сову доставить письмо напрямую получателю. Каждый раз за свои услуги Сова попросит  $d$  желудей.
- Оставить письмо на хранение в норе у Кролика. Кролик дорожит свободным местом в своей норе, поэтому хранить письмо в период времени длины  $T$  будет стоить  $c \cdot T$  желудей. Забрать письмо от Кролика получатель сможет, когда в следующий раз сам захочет оставить у него письмо, либо в момент времени  $t_{n+1}$ , когда все придут к Кролику на чай. Забирать письма в какие-либо другие моменты времени нельзя (иначе Кролик что-то заподозрит). У Кролика можно одновременно хранить сколько угодно писем, оплачивая хранение каждого по отдельности.

Помогите друзьям определить наименьшее количество желудей, которое потребуется для успешного обмена всеми письмами.

### Входные данные

В первой строке записано три целых числа  $n, c, d$  ( $1 \leq n \leq 10^5$ ,  $1 \leq c \leq 10^2$ ,  $1 \leq d \leq 10^8$ ) — количество сообщений, стоимость хранения письма у Кролика, и стоимость доставки сообщения Совой.

Следующие  $n$  строк описывают события. В  $i$ -й из этих строк записано целое число  $t_i$  и символ  $p_i$  ( $0 \leq t_i \leq 10^6$ ,  $p_i$  либо  $\bar{W}$ , либо  $\bar{P}$ ) — время, в которое происходит  $i$ -е событие, и с кем это событие происходит.

В последней строке записано одно целое число  $t_{n+1}$  ( $0 \leq t_{n+1} \leq 10^6$ ) — время, в которое все соберутся к Кролику на чай и заберут оставшиеся у Кролика письма.

Гарантируется, что  $t_i < t_{i+1}$  для всех  $i$  от 1 до  $n$ .

### Выходные данные

Выведите одно целое число — минимальную общую стоимость доставки.

входные данные
5 1 4 0 P 1 W 3 P 5 P 8 P 10
выходные данные
16

входные данные
10 10 94 17 W 20 W 28 W 48 W 51 P 52 W 56 W 62 P 75 P 78 P 87

выходные данные
916

Одно из оптимальных решений в первом примере:

- В момент 0 Пятачок оставляет письмо Кролику.
- В момент 1 Винни оставляет Кролику своё письмо и забирает письмо Пятачка. Поскольку это письмо пролежало у Кролика с момента 0 по 1, это стоит 1 желудь.
- В момент 3 Пятачок отправляет письмо с Совой, это стоит 4 желудя.
- В момент 5 Пятачок оставляет письмо у Кролика и забирает письмо Винни, за которое надо заплатить 4 желудя.
- В момент 8 Пятачок оставляет Кролику еще одно письмо.
- В момент 10 Винни приходит к Кролику на чай и забирает два письма Пятачка, за которые надо заплатить 5 и 2 желудя соответственно.

Таким образом, общая стоимость равна  $1 + 4 + 4 + 5 + 2 = 16$  желудей.