Задача A. Зарядка для кота

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

Хозяйка кота Бенедикта считает, что он мало двигается, и поэтому придумывает для кота разные подвижные игры. Одна из игр — погоня за светящейся точкой от лазерной указки.

Хозяйка включает лазерную указку и перемещает светящуюся точку по полу в течение n единиц времени.

Будем считать, что на пол нанесена координатная сетка. За одну единицу времени светящаяся точка перемещается от одной точки с целыми координатами к другой точке с целыми координатами. Каждую единицу времени кот оценивает, будет ли он прыгать к светящейся точке или нет.

Кот выполняет оценку следующим образом. Пусть он находится в точке с координатами (x_C, y_C), а светящаяся точка — в точке с координатами (x_L, y_L).

- Если $s \leqslant (x_C - x_L)^2 + (y_C - y_L)^2 \leqslant b$, то кот прыгнет в точку координатной сетки, где находится светящаяся точка.
- Если $0 \leqslant (x_C - x_L)^2 + (y_C - y_L)^2 \leqslant p < s$, то кот поймает светящуюся точку лапой. Перемещаться при этом он не будет.
- Во всех остальных случаях кот сочтёт расстояние либо слишком маленьким, либо слишком большим. В этом случае кот также не будет перемещаться (и будет выражать недовольство, постукивая хвостом).

По ходу игры кот подсчитывает, сколько он совершил прыжков и сколько раз поймал светящуюся точку лапой. Каждый раз, когда количество прыжков или количество пойманных светящихся точек меняется, кот вычисляет разность между этими величинами. Его интересует максимальное (по абсолютной величине) значение, которого достигала эта разность по ходу игры.

Изначально кот сидит в точке с координатами $(0, 0)$. Ваша задача — определить максимальную по абсолютной величине разность между количеством прыжков и количеством светящихся точек, пойманных лапой, которая наблюдалась по ходу игры. Также необходимо указать, в какой точке координатной сетки кот окажется по завершении игры.

Формат входных данных

В первой строке содержится целое число s ($1 \leqslant s \leqslant 10^9$) — нижняя граница оценки для прыжка кота.

Во второй строке содержится целое число b ($s \leqslant b \leqslant 10^9$) — верхняя граница оценки для прыжка кота.

В третьей строке содержится целое число p ($0 \leqslant p < s$) — верхняя граница оценки для поимки светящейся точки лапой.

В четвёртой строке содержится целое число n ($1 \leqslant n \leqslant 2 \cdot 10^5$) — количество единиц времени, в течение которого хозяйка кота перемещала лазерную указку.

В каждой из следующих n строк содержится по два целых числа x_j и y_j ($-10^4 \leqslant x_j, y_j \leqslant 10^4$, $j = 1, 2, \ldots, n$).

Формат выходных данных

Выведите максимально возможную по абсолютной величине разность между количеством прыжков, которое совершит кот, и количеством светящихся точек, пойманных лапой, а также координаты точки, в которой кот окажется (сначала координату x, потом y).

Разделяйте выводимые числа пробелами или переводами строк.

Обратите внимание, что выводить следует именно разность, а не её абсолютное значение.

Система оценки

Во всех подзадачах применяется потестовая система оценки. В графе «Баллы» указано количество баллов за тест и в скобках максимальное количество баллов, которое можно набрать за подзадачу. Участнику сообщаются номера тестов подзадачи, которые не были пройдены.

Для всех подзадач, кроме первой, требуется, чтобы программа верно решала одну или несколько из предшествующих подзадач. Более подробно разбиение на подзадачи показано в таблице ниже.

Подзадача	Баллы за тест (баллы за подзадачу)	Ограничения	Необходимые подзадачи	Информация о проверке
1	2 (до 10)	$p = 0$, $s = 1$	нет	полная
2	2 (до 10)	$p = 0$, $s \geqslant 2$	1	полная
3	2 (до 20)	$p = s - 1 \geqslant 1$	1	полная
4	3 (до 60)	любые возможные значения	1, 2, 3	полная

Примеры

стандартный ввод	стандартный вывод
1 50 0 9 2 3 1 7 2 8 -4 -3 -5 -1 0 -2 5 3 5 3 3 -5	4 5 3
30 100 20 10 2 3 1 7 2 8 -4 -3 -5 -1 -4 -3 -7 3 -5 -1 -1 4 2 0	-3 -1 4

Замечание

Поясним приведённые примеры.

Будем обозначать количество прыжков $jumps$ и количество пойманных светящихся точек $points$. В начале игры обе эти величины равны 0.

В первом примере кот действует следующим образом.

- Находясь в точке $(0, 0)$, он оценивает расстояние до точки $(2, 3)$: $1 \leqslant (0-2)^2+(0-3)^2 = 13 \leqslant 50$ и принимает решение прыгнуть. Таким образом, $jumps = 1$, кот перемещается в $(2, 3)$. Поскольку $points = 0$, то разность $jumps - points = 1$, что совпадает с количеством прыжков.

- Оценка для точки $(1, 7)$: $(2-1)^2 + (3-7)^2 = 17$, так что $jumps = 2$, кот перемещается в $(1, 7)$.
- Оценка для точки $(2, 8)$: $(1-2)^2 + (8-7)^2 = 2$, $jumps = 3$, кот перемещается в $(2, 8)$.
- Оценка для точки $(-4, -3)$: $(2+4)^2 + (8+3)^2 = 157$, и это больше $b = 50$, так что кот прыгать не будет.
- Оценка для точки $(-5, -1)$: $(2+5)^2 + (8+1)^2 = 130$, это также больше 50, и кот остаётся на месте.
- Оценка для точки $(0, -2)$: $(2-0)^2 + (8+2)^2 = 104$, кот опять остаётся на месте.
- Оценка для точки $(5, 3)$: $(2-5)^2 + (8-3)^2 = 34$, $jumps = 4$, кот перемещается в точку $(5, 3)$. Поскольку кот пока ещё ни разу не ловил светящуюся точку лапой, то и разность $jumps - points = 4$.
- Оценка для точки $(5, 3)$: $(5-5)^2 + (3-3)^2 = 0$, кот остаётся в этой точке и ловит светящуюся точку лапой, по-прежнему $jumps = 4$, но впервые изменяется $points = 1$, так что $jumps - points = 3$.
- Оценка для точки $(3, -5)$: $(5-3)^2 + (3+5)^2 = 68$, кот опять остаётся на месте.

Как можно видеть, максимально возможная разность между количеством прыжков и количеством пойманных точек будет равна 4. В конце игры кот окажется в точке $(5, 3)$.

Во втором примере кот действует следующим образом.

- Находясь в точке $(0, 0)$, он оценивает расстояние до точки $(2, 3)$: $(0-2)^2 + (0-3)^2 = 13$ и принимает решение поймать светящуюся точку лапой.. Таким образом, $points = 1$, кот остаётся в $(0, 0)$. Разность $jumps - points = -1$.
- Оценка для точки $(1, 7)$: $(0-1)^2 + (0-7)^2 = 50$, кот будет прыгать. Так что $jumps = 1$, кот перемещается в $(1, 7)$, разность $jumps - points = 0$.
- Оценка для точки $(2, 8)$: $(1-2)^2 + (8-7)^2 = 2$, $points = 2$, кот остаётся в $(1, 7)$, разность $jumps - points = -1$.
- Оценка для точки $(-4, -3)$: $(1+4)^2 + (7+3)^2 = 125$, и это больше $b = 100$, так что кот прыгать не будет, ловить точку лапой не будет тем более.
- Оценка для точки $(-5, -1)$: $(1+5)^2 + (7+1)^2 = 100$, это как раз равно 100, поэтому кот прыгнет в точку $(-5, -1)$, $jumps = 2$, разность $jumps - points = 0$.
- Оценка для точки $(-4, -3)$: $(-5+4)^2 + (-1+3)^2 = 5$, кот остаётся на месте и ловит лапой точку. Теперь $points = 3$, $jump - points = -1$.
- Оценка для точки $(-7, 3)$: $(-5+7)^2 + (-1-3)^2 = 20$, это верхняя граница для расстояния, на котором кот ещё ловит точку лапой. Поэтому $points = 4$, $jump - points = -2$, кот остаётся в точке $(-5, -1)$.
- Оценка для точки $(-5, -1)$: $(-5+5)^2 + (-1+1)^2 = 0$, кот останется в точке, в которой находился, и поймает точку лапой, $points = 5$, $jumps - points = -3$.
- Оценка для точки $(-1, 4)$: $(-5+1)^2 + (-1-4)^2 = 34$, кот прыгнет в эту точку, $jumps = 3$, $jumps - points = -2$.
- Оценка для точки $(2, 0)$: $(-1-2)^2 + (4-0)^2 = 25$, кот не станет ни прыгать, ни дотягиваться лапой до точки.

Как можно видеть, в течение игры максимальная по абсолютной величине разность между количеством прыжков и количеством пойманных светящихся точек составляла -3 (3 по абсолютной величине).

Задача B. Прогнозы

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

Кот Бенедикт уже несколько дней недоумевает: вместо фильмов про животных хозяева целыми днями смотрят странные (с точки зрения кота) игры людей. Эти люди бегают по большому полю за одним-единственным мячом, пытаясь отобрать его друг у друга.

Бенедикт уже не раз слышал, что есть 4 команды, которые для определённости будем называть A, B, C и D. Каждая из этих команд должна сыграть с каждой по одному разу. Таким образом, всего должно быть сыграно 6 матчей. Часть матчей уже состоялась, а часть — скоро состоится.

Результатом матча может быть ничья, за которую каждой из команд начисляют по одному очку, или выигрыш одной из команд и, соответственно, проигрыш другой. Выигравшей команде начисляют три очка, проигравшей — ноль очков.

Очки суммируются, и команда, набравшая максимальное количество очков, занимает первое место. Если максимальное количество очков набирает несколько команд, то все такие команды считаются занявшими первое место.

Бенедикт знает, что хозяева переживают за команду A и очень хотят, чтобы она заняла первое место. Ваша задача — зная результаты уже состоявшихся матчей, определить, может ли команда A занять первое место, и если да, то с какими результатами должны завершиться ещё не состоявшиеся матчи.

Формат входных данных

В первой строке содержится целое число n ($4 \leqslant n \leqslant 5$) —количество уже состоявшихся матчей.

Далее следует n строк, в каждой из которых содержится по три символа t_1, r, t_2 ($t_1, t_2 \in \{A, B, C, D\}$, $t_1 \neq t_2$, $r \in \{W, R\}$), t_1 и t_2 – наименования команд, а r — результат их игры: W означает, что команда t_1 победила команду t_2, а R — что команды t_1 и t_2 сыграли вничью (посмотрите пример ниже для лучшего понимания формата).

Гарантируется, что все описанные матчи попарно различны.

Формат выходных данных

В первой строке выведите YES, если возможны такие результаты оставшихся матчей, при которых команда A может занять первое место, и NO — если ни при каких результатах оставшихся матчей команда A не займёт первое место.

Если в первой строке выведено YES, далее следуют $6-n$ строк, в которых записаны результаты оставшихся матчей, при которых команда A займёт первое место, в таком же формате, как во входных данных.

Если существует несколько вариантов правильного ответа, выведите любой из них.

Система оценки

Во всех подзадачах применяется потестовая система оценки. В графе «Баллы» указано количество баллов за тест и в скобках максимальное количество баллов, которое можно набрать за подзадачу. Участнику сообщаются номера тестов подзадачи, которые не были пройдены.

Для всех подзадач, кроме первой, требуется, чтобы программа верно решала одну или несколько из предшествующих подзадач. Более подробно разбиение на подзадачи показано в таблице ниже.

Подзадача	Баллы за тест (баллы за подзадачу)	Ограничения	Необходимые подзадачи	Информация о проверке
1	2 (до 10)	$n = 5$, в оставшемся матче команда A участвует	нет	полная
2	2 (до 20)	$n = 5$, в оставшемся матче команда A не участвует	1	полная
3	2 (до 10)	$n = 4$, в обоих оставшихся матчах команда A участвует	1	полная
4	2 (до 30)	$n = 4$, в одном из оставшихся матчей команда A участвует, в другом — нет	1, 2, 3	полная
5	2 (до 30)	$n = 4$, в обоих оставшихся матчах команда A не участвует	1, 2, 3, 4	полная

Примеры

стандартный ввод	стандартный вывод
5 A W B C W A B W C C R D D R B	YES A R D
4 A R B C W B D W A C R D	YES A W C B W D
4 A R B B W C D W A C W D	NO

Замечание

Поясним приведённые примеры.

В первом примере единственная ещё не состоявшаяся игра — это игра между командами A и D.

Подсчитаем количество очков, которые набрали команды.

Команда A выиграла у B (3 очка) и проиграла C (0 очков), и на настоящий момент имеет 3 очка.

Команда B проиграла команде A (0 очков), выиграла у команды C (3 очка), а также сыграла вничью с командой D (1 очко). Итого у команды B 4 очка.

Команда C выиграла у команды A (3 очка), проиграла команде B (0 очков), сыграла вничью с командой D (1 очко), так что суммарно набрала 4 очка.

Наконец, команда D дважды сыграла вничью: с командой B и с командой C, поэтому имеет 2 очка.

Как понятно, команде A достаточно не проиграть команде D. В случае ничьей она наберёт 4 очка и займёт одно из первых мест (что допустимо по условиям задачи), в случае победы у неё будет 6 очков, что сделает её абсолютным победителем.

Таким образом, корректными будут следующие ответы:

$$A \quad R \quad D$$

$$D \quad R \quad A$$

$A \quad W \quad D$

В качестве ответа в этом примере выведен второй из перечисленых.

Во втором примере сыграно четыре игры. Как несложно определить, что между собой ещё не играли пара A и C и пара B и D.

После этих четырёх игр команды A и B имеют по одному очку, а команды C и D по четыре очка. За одну игру команда не может набрать более трёх очков, поэтому если хотя бы одна из команд C или D не проиграет, она займёт первое место. Проигрыш этих команд обеспечивается выигрышем команд A и B: в этом случае именно они поделят первое место в группе.

В третьем примере также сыграно четыре игры, и между собой ещё не играли пара A и C и пара B и D (как и во втором примере).

Однако ситуация с набранными очками здесь другая. Команда A набрала лишь одно очко, в то время как команда B имеет четыре очка, а команды C и D — по три очка. Даже если команда A выиграет свою игру и наберёт четыре очка, она не сможет занять первое место. В игре между командами B и C любой исход плох для A. Действительно, при победе B набирает 7 очков и становится единоличным лидером. В случае ничьей у команды B будет 5 очков, и обойти её может команда D, выиграв у A, но никак не команда A. Наконец, если выиграет команда C, то она наберёт 6 очков и займёт первое место.

Таким образом, команда A не сможет занять первое место ни при каких результатах оставшихся игр.

Задача C. CAT

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

Кот Бенедикт недавно узнал, как выглядит слово *кот* на английском языке (*cat*, если вдруг вы не знаете).

Бенедикт дремлет, и ему снится сон. В этом сне он видит экран, на котором последовательно одна за другой появляются буквы, быстро сменяя друг друга. Каждая буква — это либо C, либо A, либо T.

Букв много, но ему нужно дотронуться лапкой сначала до какой-то из букв C, потом до какой-то из букв A, а затем — до какой-то из букв T. Когда ему это удаётся, буквы пропадают, а на экране показывают забавный короткий ролик про котов.

Ваша задача — определить, сколькими способами Бенедикт может сформировать слово CAT. Поскольку это число может быть достаточно большим, выведите в качестве ответа остаток от деления этого числа на 10000019 $\left(10^7 + 19\right)$.

Формат входных данных

В первой строке содержится непустая последовательность, состоящая из символов C, A, T (заглавных букв латинского алфавита), в том порядке, в котором они появляются на экране. Последовательность имеет длину не более 10^5 символов.

Гарантируется, что последовательность содержит хотя бы один символ C, хотя бы один символ A и хотя бы один символ T.

Формат выходных данных

Выведите единственное целое число — остаток от деления на 10000019 $\left(10^7 + 19\right)$ количества способов, которыми кот может сформировать слово CAT.

Система оценки

В подзадачах с первой по четвёртую применяется потестовая система оценки. В графе «Баллы» указано количество баллов за тест и в скобках максимальное количество баллов, которое можно набрать за подзадачу. Участнику сообщаются номера тестов подзадачи, которые не были пройдены.

В пятой подзадаче баллы начисляются только в случае прохождения всех тестов этой подзадачи. Участнику сообщается либо номер первого непройденного теста и результат проверки на этом тесте, либо что все тесты подзадачи пройдены.

Почти для всех подзадач требуется, чтобы программа верно решала одну или несколько из предшествующих подзадач. Более подробно разбиение на подзадачи показано в таблице ниже.

Ради краткости будем обозначать входную строку s, а её длину $|s|$.

Подзадача	Баллы за тест (баллы за подзадачу)	Ограничения	Необходимые подзадачи	Информация о проверке
1	2 (до 10)	$\|s\| \leqslant 6$	нет	полная
2	2 (до 10)	$\|s\| \leqslant 1000$, строка является соединением трёх строк: одной строки только из символов C, другой — только из символов A и третьей — только из символов T, т.е. имеет вид $C \ldots CA \ldots AT \ldots T$	нет	полная
3	2 (до 10)	в строке содержится только один символ A, $\|s\| \leqslant 10^5$	нет	полная
4	2 (до 20)	$\|s\| \leqslant 10^2$	1, 2, 3	полная
5	0 (50)	$\|s\| \leqslant 10^5$	1, 2, 3, 4	первая ошибка

Примеры

стандартный ввод	стандартный вывод
CACATT	6
ACTCT	0
ATCCATATTA	10

Замечание

Поясним приведённые примеры.

Перенумеруем буквы в первом примере, начиная с 1. Тогда слово CAT можно сформировать следующими способами: 125, 126, 145, 146, 345, 346. Иных способов последовательно выбрать сначала C, потом A, потом T, не существует.

Во втором примере нет ни одного способа последовательно получить слово CAT.

Перенумеруем буквы в третьем примере (также начиная с 1). Способы, которыми можно получить слово CAT, будут следующими: 356, 456, 358, 458, 359, 459, 378, 478, 379, 479. В отличие от первого примера не все буквы участвуют в образовании слова.

Также приведём пояснение про вычисления с остатком от деления.

Пусть есть числа a и b, которые нужно сложить и сообщить остаток от деления их суммы на число d. Можно сделать это непосредственно: сначала вычислить $s = a + b$, а затем посчитать $s \mod d$, где `mod` обозначена операция взятия остатка от деления.

Однако можно поступить и по-другому: сначала вычислить $q = a \mod d$ и $p = b \mod d$, а потом уже посчитать $(p + q) \mod d$. Действительно, если a представимо в виде $k_a \cdot d + q$, а b — в виде $k_b \cdot d + p$, то их сумма запишется как $s = (k_a + k_b) \cdot d + (q + p)$. При вычислении остатка от деления s на d определять его значение будет только сумма $(q + p)$.

Хотя по отдельности $q < d$ и $p < d$, их сумма может быть больше d, поэтому результат вычисляется именно как $(p + q) \mod d$.

Такой подход позволяет поддерживать результат вычислений не превосходящим значения $2 \cdot (d - 1)$.

Аналогично можно показать, что остаток от деления произведения чисел a и b на число d будет определяться произведением их остатков — $(p \cdot q) \mod d$.

Задача D. Камышовый кот

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

Коту Бенедикту очень нравится число k и он хочет стать k-мышовым котом.

Как известно, кот считается k-мышовым, если он в течение некоторого времени ловит не менее k мышей. Известны последовательные моменты времени $(0 \leqslant) t_1 \leqslant t_2 \leqslant \ldots \leqslant t_n$, в которые кот Бенедикт ловил мышей. Для каждого момента t_i также известно количество мышей m_i, которых кот поймал в этот момент.

Кот Бенедикт хотел бы продемонстрировать, что он способен поймать k мышей за короткое время. Поэтому он хочет выбрать наиболее короткий отрезок времени из диапазона от 0 до t_n (включительно), в течение которого он поймал не менее k мышей.

Ваша задача — указать границы такого временного отрезка минимальной длины, которые позволят считать кота Бенедикта k-мышовым. Если существует несколько вариантов ответа, выведите любой из них.

Формат входных данных

В первой строке содержатся целые числа n и k ($1 \leqslant n \leqslant 3 \cdot 10^5$, $1 \leqslant k \leqslant 10^{18}$) — количество моментов времени, в которые кот Бенедикт ловил мышей, а также число k.

Во второй строке содержатся целые числа $t_1, t_2, \ldots, t_n$ $\left(0 \leqslant t_1 \leqslant t_2 \leqslant \ldots \leqslant t_n \leqslant 10^{18}\right)$ — моменты времени, в которые кот Бенедикт ловил мышей.

В третьей строке содержатся целые числа $m_1, m_2, \ldots, m_n$ $\left(1 \leqslant m_j \leqslant 10^9,\ j = 1, 2, \ldots, n\right)$, m_j — количество мышей, которых кот Бенедикт поймал в момент t_j.

Формат выходных данных

Выведите два целых числа — начальный и конечный моменты времени, ограничивающие временной отрезок минимальной длины, в течение которого кот Бенедикт поймал не менее k мышей. Разделяйте числа пробелом или переводом строки.

Если существует несколько вариантов ответа, выведите любой из них.

Если кот Бенедикт не сможет стать k-мышовым котом, выведите в качестве ответа два числа: -1 -1.

Система оценки

В первой и второй подзадачах применяется потестовая система оценки. В графе «Баллы» указано количество баллов за тест и в скобках максимальное количество баллов, которое можно набрать за подзадачу. Участнику сообщаются номера тестов подзадачи, которые не были пройдены.

Баллы за подзадачи с третьей по пятую начисляются только в случае прохождения всех тестов этой подзадачи. Участнику сообщается либо номер первого непройденного теста и результат проверки на этом тесте, либо что все тесты подзадачи пройдены.

Для всех подзадач, кроме первой, требуется, чтобы программа верно решала одну или несколько из предшествующих подзадач.

Более подробно разбиение на подзадачи показано в таблице ниже.

Подзадача	Баллы за тест (баллы за подзадачу)	Ограничения	Необходимые подзадачи	Информация о проверке
1	2 (до 20)	$n \leqslant 1000$, $k \leqslant 1000$, $0 \leqslant t_1 < t_2 \ldots < t_n \leqslant 10^6$, $m_j = 1\,(j = 1, 2, \ldots, n)$	нет	полная
2	2 (до 20)	$n \leqslant 1000$, $k \leqslant 10^9$, $0 \leqslant t_1 < t_2 \ldots < t_n \leqslant 10^6$, $m_j \leqslant 10^6\,(j = 1, 2, \ldots, n)$	1	полная
3	0 (20)	$n \leqslant 3 \cdot 10^5$, $k \leqslant 10^9$, $0 \leqslant t_1 < t_2 \ldots < t_n \leqslant 10^9$, $m_j = 1\,(j = 1, 2, \ldots, n)$	1	первая ошибка
4	0 (20)	$n \leqslant 3 \cdot 10^5$, $k \leqslant 10^9$, $0 \leqslant t_1 < t_2 \ldots < t_n \leqslant 10^9$, $m_j \leqslant 5000\,(j = 1, 2, \ldots, n)$	1, 2, 3	первая ошибка
5	0 (20)	любые возможные значения	1, 2, 3, 4	первая ошибка

Примеры

стандартный ввод	стандартный вывод
7 3 0 6 8 19 22 23 25 1 1 1 1 1 1 1	22 25
11 7 5 8 8 14 23 37 42 42 55 64 71 2 1 2 1 4 3 2 2 5 6 2	37 42
5 8 1 5 15 20 28 2 1 1 2 1	-1 -1

Замечание

Поясним приведённые примеры.

В первом примере кот ловит по одной мыши в каждый момент времени.

- На то, чтобы поймать первых трёх мышей, кот потратит 8 единиц времени (первую мышь он поймает в момент 0, а третью — в момент 8).
- Если кот будет учитывать мышей со второй по четвёртую, то на их поимку уйдёт 13 единиц времени.
- Если кот будет учитывать мышей с третьей по пятую, то их он поймает за 14 единиц времени.
- Если кот будет учитывать мышей с четвёртой по шестую, то на их поимку будет потрачено 4 единицы времени.
- Наконец, если кот будет учитывать мышей с пятой по седьмую, то их он поймает за 3 единицы времени.

Таким образом, кот может утверждать, что может стать k-мышовым котом за 3 единицы времени.

Во втором примере минимальное время, которое потребуется коту — 5 единиц времени: он поймает трёх мышей в момент 37, а затем еще дважды по две мыши в момент 42.

В третьем примере кот не сможет стать k-мышовым

Задача E. Фонари

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

С точки зрения кота Бенедикта, каждое время суток интересно по-своему. Когда наступает вечер, в парке включают фонари, и Бенедикту из окна видна цепочка из n светящихся шаров.

Однако лампочки в фонарях иногда перегорают, и работникам парка приходится их заменять. Работники парка полагают, что сразу же заменять перегоревшую лампочку необязательно. Они дожидаются первого момента времени, когда перегоревших лампочек станет не менее m, и только тогда отправляются менять их на новые.

Работники парка, отправляясь менять лампочки, действуют следующим образом.

Они берут лампочки со склада в нужном для замены всех перегоревших лампочек количестве (обратите внимание, что одновременно могут перегорать несколько лампочек, поэтому количество взятых лампочек может превосходить m) и двигаются в направлении от первого к последнему фонарю.

Все лампочки занумерованы номерами от 1 до s ($n \leqslant s$). Работники устанавливают лампочки в соответствии с их номерами:

- Лампочка с большим номером не может быть использована для замены, пока не использованы все лампочки с меньшими номерами;
- Если две лампочки взяты в рамках одного «набора для замены», то лампочка с меньшим номером будет установлена в фонарь с меньшим номером (независимо от того, в каком порядке фонари перегорали).

Когда работники в очередной раз придут на склад и обнаружат, что имеющегося количества лампочек недостаточно для замены всех перегоревших, они не станут менять лампочки вовсе и проинформируют начальство о необходимости приобретения лампочек. Приобретение лампочек — процесс достаточно долгий, поэтому после информирования начальства работники парка в полном составе отправятся в отпуск.

Вам известна длительность работы в днях лампочки в каждом фонаре и каждой лампочки на складе.

Считайте, что все лампочки в фонарях установлены сегодня, и это первый день работы лампочек в фонарях. Из соображений общности установку лампочек будем считать первой состоявшейся заменой. Лампочка, находящаяся на складе, не расходует свой ресурс, пока не будет установлена в фонаре.

Работники парка интересуются номером дня k, в который они проинформируют начальство о необходимости закупки лампочек (и, соответственно, у них появится возможность отправиться в отпуск).

Кот Бенедикт хотел бы узнать:

- какое максимальное количество дней пройдёт между двумя последовательными заменами лампочек в каком-либо одном фонаре;
- какое максимальное **суммарное** количество дней какой-либо один фонарь будет оставаться не работающим.

Оба этих значения кот Бенедикт хочет узнать на отрезке от сегодняшнего дня до дня k.

Ваша задача — определить эти три числа.

Формат входных данных

В первой строке содержатся целые числа n, s, m $\left(1 \leqslant n \leqslant s \leqslant 2 \cdot 10^5; 1 \leqslant m \leqslant n\right)$ — количество фонарей, количество лампочек на складе, количество перегоревших лампочек, которое с точки зрения работников парка является поводом отправиться их менять.

Во второй строке содержится n целых чисел $d_1, d_2, \ldots, d_n$ ($1 \leqslant d_i \leqslant 10^6$, $i = 1, 2, \ldots, n$), d_i — длительность работы лампочки, установленной в фонаре $\#i$.

В третьей строке содержится s целых чисел $t_1, t_2, \ldots, t_s$ ($1 \leqslant t_j \leqslant 10^6$, $j = 1, 2, \ldots, s$), t_j — длительность работы лампочки, имеющей складской номер $\#j$.

Формат выходных данных

Выведите три числа.

Первое число — номер дня, в который работникам парка придётся писать заявку (и они смогут отправиться в долгожданный отпуск).

Второе число — максимальное количество дней, которое пройдёт между двумя последовательными заменами лампочек в каком-либо одном фонаре.

Третье число — максимальное **суммарное** количество дней, которое какой-либо один фонарь будет оставаться неработающим.

Разделяйте числа пробелами или переводами строк.

Система оценки

Во подзадачах с первой по третью применяется потестовая система оценки. В графе «Баллы» указано количество баллов за тест и в скобках максимальное количество баллов, которое можно набрать за подзадачу. Участнику сообщаются номера тестов подзадачи, которые не были пройдены.

Баллы за четвёртую подзадачу начисляются только в случае прохождения всех тестов этой подзадачи. Участнику сообщается либо номер первого непройденного теста и результат проверки на этом тесте, либо что все тесты подзадачи пройдены.

В первой подзадаче лампочки, исходно установленные в фонарях, двух видов: с длительностью работы a и с длительностью работы b. Все лампочки на складе имеют одну и ту же длительность работы c.

Во второй подзадаче все лампочки на складе имеют одну и ту же длительность работы c.

Для всех подзадач, кроме первой, требуется, чтобы программа верно решала одну или несколько из предшествующих подзадач.

Более подробно разбиение на подзадачи показано в таблице ниже.

Подзадача	Баллы за тест (баллы за подзадачу)	Ограничения	Необходимые подзадачи	Информация о проверке
1	1 (до 10)	$s \leqslant 1000$, $d_j \leqslant 1000$, $t_i \leqslant 1000$ $d_j \in \{a, b\}$, $t_i = c$ для всех j, i	нет	полная
2	2 (до 30)	$s \leqslant 1000$, $d_j \leqslant 1000$, $t_i \leqslant 1000$ $t_i = c$ для всех j, i	1	полная
3	2 (до 30)	$s \leqslant 1000$, $d_j \leqslant 1000$, $t_i \leqslant 1000$ для всех j, i	1, 2	полная
4	(30)	любые возможные значения	1, 2, 3	первая ошибка

Примеры

стандартный ввод	стандартный вывод
3 3 2 7 4 7 8 4 6	13 7 5
7 19 3 5 2 14 8 11 8 18 7 4 6 3 3 4 10 21 19 19 20 21 12 13 3 5 4 4 5	36 22 7

Замечание

Во втором примере (**только в условии**) часть данных из третьей строки перенесена на четвёртую строку, поскольку не помещалась в колонку таблицы. В тестирующей системе никаких переносов строк нет.

Поясним приведённые примеры.

В первом примере работники отправятся менять лампочки в день 7. В день 4 перегорит одна лампочка, но для них это не будет поводом делать замену (они планируют делать замены, если перегоревших лампочек не меньше 2). В день 7 перегорят ещё две лампочки, суммарное количество перегоревших станет равным 3, и работники пойдут на склад.

На складе они обнаружат достаточное количество лампочек — 3 штуки. Так что в день 7 в первом фонаре будет установлена лампочка, которая проработает 8 дней, во втором — лампочка, которая проработает 4 дней, а в третьем — лампочка, которая проработает 6 дней.

Это означает, что в день 11 лампочка, установленная во втором фонаре, перегорит. Через два дня, в день 13 перегорит лампочка в третьем фонаре. Перегоревших лампочек станет 2, и работники отправятся на склад. Поскольку лампочек на складе не окажется, в день 13 они составят заявку.

В период времени до дня 13 замены лампочек производились в «нулевой» день, а затем в седьмой день — так что максимальное время прошедшее между двумя последовательными заменами составит 7 дней.

Максимальное суммарное время, которое не работал какой-либо один фонарь, составляет 5 дней для второго фонаря. Как легко видеть, первый и третий фонари до дня 13 всегда были в работающем состоянии. А вот второй фонарь не работал сначала 3 дня — с 4 по 7 день, а затем ещё 2 дня — с 11 по 13 день. В условии предлагается ограничиться днём, в который была сформирована заявка на приобретение лампочек, поэтому о дальнейшей «истории» этого фонаря никаких предположений не делается.

Второй пример демонстрирует различные ситуации замен лампочек. Заметим, что значение 22 (максимальное количество дней, прошедших между последовательными заменами лампочек) достигается для седьмого фонаря.

Максимальное суммарное количество дней, которое не работал какой-либо фонарь, составляет 7 и достигается для второго и пятого фонарей.