

## Задача А. Три и один

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

Кот Бенедикт любит наблюдать из окна за происходящим в парке. На одном из газонов парка устроили инсталляцию из камней. Мелкие камни образуют равномерную сетку из квадратов со стороной длины 1, а в трех узлах сетки расположены валуны. Эти валуны давно облюбовали три уличных кота — на них можно греться в прохладный, но солнечный осенний день.

Иногда коты меняются местами: первый отправляется на место второго, второй на место третьего, а третий — на место первого. Коты перемещаются исключительно по мелким камням: газон влажный, и они не хотят пачкать лапки. Именно это коты и задумали сделать, но... В один из узлов сетки пришёл четвёртый кот, имеющий намерение занять один из тёплых валунов.

Коты планировали одновременно покинуть те валуны, на которых они располагаются, и отправиться по кратчайшему пути (по дорожкам из камней) к тем валунам, на которых они хотят расположиться. Их интересует, смогут ли они сделать это, или же кого-то из них сможет опередить четвёртый кот.

Считайте, что все четыре кота начинают двигаться одновременно, двигаются с одинаковой скоростью и только по дорожкам из мелких камней. Также считайте, что при движении они не мешают друг другу, даже если будут двигаться по одной дорожке. Вы можете быть уверены, что если один из трёх котов и четвёртый одновременно достигнут валуна, четвёртый кот окажется менее проворным и не сможет занять валун.

Выведите в качестве ответа количество валунов, к которым четвёртый кот может добраться **строго быстрее** других котов. Также выведите номера этих валунов.

### Формат входных данных

В первой строке содержится номер горизонтальной дорожки, на которой располагается первый валун.

Во второй строке содержится номер вертикальной дорожки, на которой располагается первый валун.

В третьей строке содержится номер горизонтальной дорожки, на которой располагается второй валун.

В четвёртой строке содержится номер вертикальной дорожки, на которой располагается второй валун.

В пятой строке содержится номер горизонтальной дорожки, на которой располагается третий валун.

В шестой строке содержится номер вертикальной дорожки, на которой располагается третий валун.

В седьмой строке содержится номер горизонтальной дорожки, на которой располагается четвёртый кот.

В восьмой строке содержится номер вертикальной дорожки, на которой располагается четвёртый кот.

Все номера дорожек — положительные целые числа, не превосходящие 10000.

Гарантируется, что все валуны расположены в разных узлах сетки. Также гарантируется, что четвёртый кот находится в узле сетки, отличном от положения какого-либо валуна.

### Формат выходных данных

В первой строке выведите количество валунов, к которым четвёртый кот может добраться **строго быстрее** других котов.

Далее выведите номера этих валунов **в порядке возрастания**. Разделяйте номера пробелами или переводами строки.

## Система оценки

В этой задаче баллы начисляются независимо за каждый тест. Каждый из 20 тестов оценивается в 5 баллов. Тесты из условия не оцениваются.

## Примеры

стандартный ввод	стандартный вывод
5 8 2 7 6 4 3 3	1 3
5 8 2 7 6 4 4 7	3 1 2 3
5 8 2 7 6 4 5 6	2 1 3

## Замечание

Поясним приведённые примеры.

В первом примере четвёртый кот может опередить только второго кота и занять третий валун раньше него.

Первый кот отправится ко второму валуну и сможет достичь его, пройдя расстояние 4 единицы. Четвёртому же коту до второго валуна нужно пройти 5 единиц.

Второй кот отправится к третьему валуну, и его путь будет иметь длину 7 единиц. А вот путь четвёртого кота до третьего валуна будет иметь длину 4 единицы.

Наконец, третий кот отправится к первому валуну, и для этого ему потребуется преодолеть расстояние в 5 единиц. Четвёртому же коту понадобится пройти 7 единиц.

Во втором примере валуны находятся на тех же позициях, что и в первом примере, поэтому пути котиков к валунам будут иметь ту же длину.

А вот четвёртый кот занял другое место.

Длина пути четвёртого кота до первого валуна составит 2 единицы, поэтому он сможет опередить третьего кота, которому требуется пройти 5 единиц длины.

Длина пути четвёртого кота до второго валуна также составит 2 единицы, поэтому он сможет опередить и первого кота, которому требуется пройти 4 единицы длины.

Наконец, длина пути четвёртого кота до третьего валуна составит 5 единиц, поэтому он сможет опередить и второго кота, которому требуется пройти 7 единиц длины.

Таким образом, четвёртый кот может достичь любого из валунов раньше того кота, который к нему направится.

В третьем примере валуны по-прежнему находятся на тех же позициях, что и в первом, и во втором примерах.

Четвёртый кот может опередить третьего на пути к первому валуну: длина пути четвёртого кота составит 2 единицы, а длина пути третьего кота — 5 единиц.

Также четвёртый кот может опередить второго кота на пути к третьему валуну: длина пути четвёртого кота составит 3 единицы, а второму нужно проделать путь длиной 7 единиц.

А вот ко второму валуну четвёртый кот придёт одновременно с первым: и тот, и другой преодолеют путь длиной 4 единицы. Однако по условию задачи валун займет первый кот.

## Задача В. Абсолютно относительно

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

Из окна Бенедикту видно площадку в парке, на которой участники кружка робототехники проводят тестовые испытания своих разработок. Сейчас маленькая платформа на колесиках должна была проехать по заданному маршруту, который был сформирован заранее.

Участник кружка, который составлял маршрут, задавал направление движения платформы относительно карты местности: команда  $N$  означала движение на одну единицу расстояния вверх по карте,  $S$  – вниз,  $W$  – влево,  $E$  – вправо.

Ориентация карты при этом не менялась. Будем называть эту карту неподвижной, а маршрут, проложенный по неподвижной карте – абсолютным.

Однако карта, которую использует платформа, связана с платформой. Это означает, что если платформа двигается, не меняя направления движения, она двигается в направлении  $N$ . Команда  $S$  предписывает платформе выполнить разворот и переместиться на одну единицу расстояния в направлении, противоположном направлению, в котором платформа была ориентирована в предыдущий момент времени. Аналогично, команда  $W$  предписывает платформе повернуть налево относительно её текущей ориентации и переместиться на одну единицу расстояния в новом направлении, а команда  $E$  предписывает платформе повернуть относительно текущей ориентации направо и также переместиться на одну единицу расстояния в этом направлении.

Будем называть эту карту связанной с платформой, а маршрут, проложенный по связанной с платформой карте, – относительным.

В начальный момент времени платформа была ориентирована по направлению  $N$  на неподвижной карте (иными словами, карта, связанная с платформой, совпадала с неподвижной картой).

Ваша задача – проложить относительный маршрут таким образом, чтобы при движении платформы он в точности совпал с абсолютным. Иными словами, нужно проложить маршрут по карте, связанной с платформой, так, чтобы после каждого шага платформа оказывалась в соответствующей точке маршрута на неподвижной карте, составленного участником кружка.

### Формат входных данных

В первой строке содержится непустая последовательность символов  $N, S, W, E$  (без пробелов) – абсолютный маршрут.

Количество символов в последовательности не превосходит  $10^5$ .

### Формат выходных данных

Выведите последовательность символов  $N, S, W, E$  (той же длины, что и исходная) – относительный маршрут, совпадающий при движении платформы с абсолютным.

### Система оценки

В каждой из двух подзадач этой задачи 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 второй подзадачи требуется, чтобы программа верно решала первую подзадачу. Более подробно разбиение на подзадачи показано в таблице ниже.

Подзадача	Баллы за тест (баллы за подзадачу)	Ограничения	Необходимые подзадачи	Информация о проверке
1	до 3 (до 40)	в абсолютном маршруте используются только два символа направления	нет	полная
2	4 (до 60)	в абсолютном маршруте используются любые символы направлений	1	полная

## Пример

стандартный ввод	стандартный вывод
NWSEESNW	NWWWNESW

## Замечание

Поясним приведённый пример.

Будем считать, что на неподвижной карте ось абсцисс направлена вправо, а ось ординат — вверх. Также будем считать, что изначально платформа располагается в точке  $(0, 0)$ .

Первая команда в абсолютном маршруте  $N$ , платформа должна переместиться в точку  $(0, 1)$ . Поскольку платформа изначально ориентирована по направлению  $N$  неподвижной карты, то первой командой в относительном маршруте также будет команда  $N$ .

Вторая команда  $W$  предписывает платформе повернуть налево и переместиться в точку  $(-1, 1)$ . В относительном маршруте для этого потребуется такая же команда  $W$ .

Третья команда  $S$  предписывает платформе переместиться в точку  $(-1, 0)$ . Поскольку после выполнения второй команды платформа ориентирована по направлению  $W$ , то на карте, связанной с платформой, это вновь будет поворот налево, т.е. команда  $W$ .

Четвёртая команда  $E$  предписывает платформе переместиться в точку  $(0, 0)$ . Для карты, связанной с платформой, это вновь будет поворот налево и команда  $W$ .

Пятая команда  $E$  предписывает платформе переместиться в точку  $(0, 1)$ . Для карты, связанной с платформой, это движение в текущем направлении и команда  $N$ .

Шестая команда  $S$  предписывает платформе переместиться в точку  $(-1, 1)$ . Для карты, связанной с платформой, это поворот направо относительно текущего направления и команда  $E$ .

Седьмая команда  $N$  предписывает платформе переместиться в точку  $(0, 1)$ . Для карты, связанной с платформой, это разворот в противоположном направлении и команда  $S$ .

Наконец, восьмая команда  $W$  предписывает платформе переместиться в точку  $(0, 0)$ . Для карты, связанной с платформой, это поворот налево и (также) команда  $W$ .

## Задача С. Вывоз мусора

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

Вчера кот Бенедикт целый день мог наблюдать из окна, как работники парка собирают опавшую листву и ветки и складывают их в специальные мешки. Мешки были сложены вдоль узкой аллеи, и сегодня приехали их забирать. Вдоль узкой прямой аллеи едет маленький погрузчик и собирает мешки в ковш. В некоторых местах с аллеи есть выезды на широкую дорогу. Когда погрузчик набирает в ковш  $k$  мешков, он отправляется к ближайшему выезду и сбрасывает эти мешки в кузов большого грузовика.

В случае, если погрузчик находится на равном расстоянии от двух выездов, он выбирает выезд с большей координатой.

Разумеется, погрузчик отвезет последнюю партию мешков к большому грузовику, даже если мешков будет меньше  $k$ .

Вам известны координаты всех  $n$  мешков с мусором и всех  $m$  выездов на широкую дорогу. Погрузчик собирает мешки строго в порядке их появления в исходных данных.

Ваша задача — определить расстояние, которое придётся проехать погрузчику, чтобы поместить все мешки с мусором в кузов грузовика.

Изначально погрузчик находится в точке с координатой 0. Расстояние, которое погрузчик проезжает по выезду с аллеи на дорогу (или обратно) равно 1.

### Формат входных данных

В первой строке содержится целое число  $k$  ( $1 \leq k \leq 10^5$ ) — количество мешков, которые могут поместиться в ковш погрузчика.

Во второй строке содержится целое число  $m$  ( $1 \leq m \leq 10^5$ ) — количество выездов с аллеи.

В каждой из следующих  $m$  строк содержится по одному целому числу  $c_j$  ( $0 \leq c_j \leq 10^9$ ,  $j = 1, 2, \dots, m$ ) — координате выезда с аллеи.

Гарантируется, что  $c_1 < c_2 < \dots < c_m$ .

В следующей строке содержится целое число  $n$  ( $1 \leq n \leq 10^5$ ) — количество мешков с мусором.

В каждой из следующих  $n$  строк содержится по одному целому числу  $b_i$  ( $0 \leq b_i \leq 10^9$ ,  $i = 1, 2, \dots, n$ ) — координате очередного мешка с мусором.

Гарантируется, что  $b_1 \leq b_2 \leq \dots \leq b_n$ .

### Формат выходных данных

Выведите единственное целое число — расстояние, которое понадобится проехать погрузчику, чтобы поместить все мешки с мусором в кузов грузовика.

Обратите внимание, что ответ может превышать возможное значение 32-битной целочисленной переменной, поэтому необходимо использовать 64-битные целочисленные типы данных (тип *int64* в языке *Pascal*, тип *long long* в *C++*, тип *long* в *Java* и *C#*).

### Система оценки

В первых двух 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.

В третьей 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.

Для второй и третьей подзадач требуется, чтобы программа верно решала предшествующие подзадачи. 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 30)	$n, m \leq 1000$	нет	полная
2	2 (до 30)	$n, m \leq 10^5, k \geq 10,$	1	полная
3	0 (40)	любые допустимые значения	1,2	первая ошибка

## Пример

стандартный ввод	стандартный вывод
2	77
4	
0	
12	
18	
25	
15	
3	
4	
4	
4	
4	
6	
6	
6	
13	
16	
16	
18	
19	
21	
21	

## Замечание

Поясним приведённый пример.

В ковш погрузчика помещается  $k = 2$  мешка с мусором.

Выездов на широкую дорогу 4, они имеют координаты 0, 12, 18 и 25.

Мешков с мусором всего  $n = 15$ , имеют координаты 3, 4, 4, 4, 4, 6, 6, 6, 13, 16, 16, 18, 19, 21, 21.

Сначала погрузчик (стартуя из точки 0) отправляется в точку с координатой 3 и забирает там единственный мешок. Затем он переезжает в точку 4 и забирает один из 4 мешков, находящихся в этой точке. Ближайший выезд находится в точке 0, поэтому погрузчик отвозит мешки туда. Сразу после выгрузки мешков проделанный им путь равен 9 (4 до более дальнего мешка, столько же обратно и 1 по выезду).

Затем погрузчик заберёт ещё два мешка из этой же точки 4 (1 по выезду, 4 до мешков, 4 обратно, ещё 1 по выезду). Суммарный путь становится равным 19 (в момент выгрузки).

После этого погрузчик отправится за последним мешком в точке 4, проедет в точку 6 и заберёт один мешок из этой точки. Расстояния от точки 6 до выезда с координатой 0 и до выезда с координатой 12 одинаковые; по условию задачи погрузчик отправится к выезду с большей координатой. Суммарный путь составит  $19 + 1 + 12 + 1 = 33$  к моменту выгрузки.

Затем погрузчик вернётся в точку 6 и отвезёт ещё два мешка. Суммарный путь в момент их выгрузки будет  $33 + 1 + 6 + 6 + 1 = 47$ .

Следующая точка, к которой поедет погрузчик, имеет координату 13. В этой точке он заберёт один мешок и отправится в точку 16, где заберёт ещё один мешок. Ближайший выезд находится в точке 18, поэтому к моменту выгрузки этих мешков суммарный путь составит  $47 + 1 + 6 + 1 = 55$ .

Теперь погрузчику придётся вернуться в точку 16, чтобы забрать там оставшийся мешок, а затем отправиться в точку 18, чтобы забрать ещё один мешок и отвезти его к грузовику. Суммарный путь в момент выгрузки этих мешков будет  $55 + 1 + 2 + 2 + 1 = 61$ .

После этого погрузчик поедет сначала в точку 19, а затем в точку 21 и заберёт в ковш по одному мешку в каждой точке. Ближайшим выездом к точке 21 является выезд в точке 18, так что суммарный путь к моменту выгрузки этих мешков составит  $61 + 1 + 3 + 3 + 1 = 69$ .

Наконец, погрузчик отправится в точку 21 за последним мешком и отвезёт его через выезд в точке 18. Суммарный путь, который проделает погрузчик к моменту выгрузки этого мешка, составит  $69 + 1 + 3 + 3 + 1 = 77$ . Это число и будет ответом.

## Задача D. Занавеска

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

Недавно кот Бенедикт с интересом наблюдал, как его хозяйка вешает новую занавеску. Занавеска держится на  $n$  крючках вдоль всей гардины.

Очевидно, что если начать вешать занавеску на крючки слева направо подряд, то нагрузка будет неравномерной и гардина может свалиться.

Поэтому хозяйка кота Бенедикта придерживается следующей стратегии:

- **Сначала** она одновременно вешает занавеску на крючки 1 и  $n$ .
- Далее, пока есть хотя бы один неиспользованный крючок:
  - хозяйка кота Бенедикта ищет непрерывный отрезок из неиспользованных крючков.
    - \* Среди таких отрезков она выбирает отрезок **наибольшей** длины.
    - \* Среди отрезков наибольшей длины она выбирает **самый левый**.
  - Пусть хозяйка кота Бенедикта выбрала отрезок с  $l$ -го по  $r$ -й крючок, всего на нём  $s = r - l + 1$  крючков. Хозяйка кота Бенедикта старается подвесить занавеску в **центр отрезка**.
    - \* Если количество крючков  $S$  нечетное, то хозяйка кота Бенедикта вешает занавеску на крючок  $\frac{l+r}{2}$ .
    - \* Иначе хозяйка кота Бенедикта одновременно вешает занавеску на крючки  $\frac{l+r-1}{2}$  и  $\frac{l+r+1}{2}$ .

После того, как хозяйка кота Бенедикта повесила занавеску, её заинтересовало, а **на каком шаге** были использованы крючки под номерами  $a_1, a_2, \dots, a_q$ .

Ваша задача — определить для каждого интересующего хозяйку кота Бенедикта крючка, на каком шаге этот крючок был использован.

### Формат входных данных

В первой строке через пробел даны два целых числа  $n$  и  $q$  ( $1 \leq n \leq 10^{18}; 1 \leq q \leq 10^3$ ) — количество крючков на гардине и количество интересующих хозяйку кота Бенедикта крючков.

Во второй строке через пробел записаны  $q$  различных целых чисел  $a_1, a_2, \dots, a_q$  ( $1 \leq a_i \leq n$ ) — интересующие хозяйку кота Бенедикта крючки.

### Формат выходных данных

В единственной строке через пробел выведите  $q$  чисел  $t_i$  — на каком шаге был использован крючок  $a_i$ .

### Система оценки

В первых трех 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.

В следующих трёх подзадачах 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.

Также для некоторых подзадач требуется, чтобы программа верно решала предшествующие подзадачи. 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)	$n \leq 10, q = n$	нет	полная
2	3 (до 15)	$n \leq 10^3,$	1	полная
3	3 (до 15)	$n \leq 3 \cdot 10^5,$	1, 2	полная
4	0 (20)	Ответы $t_i \leq 3 \cdot 10^5$	1, 2, 3	первая ошибка
5	0 (15)	$n = 2^k$ для $k > 0$	нет	первая ошибка
6	0 (25)	любые допустимые значения	1, 2, 3, 4, 5	первая ошибка

## Примеры

стандартный ввод	стандартный вывод
10 10 1 10 2 9 3 8 4 7 5 6	1 1 5 8 3 4 6 7 2 2
9876543210 3 3456789120 5678912340 7891234560	5798205414 6764530002 3863296238

## Замечание

**Первый** тестовый пример.

На гардине  $n = 10$  крючков.

Крючки будут использованы в следующем порядке:

- На первом шаге вешаем на крайние крючки 1 и 10;
- На втором шаге есть один отрезок крючков  $[2; 9]$ , вешаем на центральные крючки  $5 = \frac{2+9-1}{2}$  и  $6 = \frac{2+9+1}{2}$ ;
- На третьем шаге есть два отрезка  $[2; 4]$ ,  $[7; 9]$ :
  - оба имеют длину 3, выбираем самый левый  $[2; 4]$ ;
  - вешаем на центральный крючок  $3 = \frac{2+4}{2}$ ;
- На четвертом шаге есть три отрезка  $[2; 2]$ ,  $[4; 4]$ ,  $[7; 9]$ :
  - выбираем отрезок  $[7; 9]$ , так как он имеет наибольшую длину 3 (длина остальных 1);
  - вешаем на центральный крючок  $8 = \frac{7+9}{2}$ ;
- Остались крючки 2, 4, 7 и 9 — вешаем на самый левый 2;
- Остались крючки 4, 7 и 9 — вешаем на самый левый 4;
- Остались крючки 7 и 9 — вешаем на самый левый 7;
- Остался единственный крючок 9 — вешаем на него и заканчиваем процесс.