

## Задача А. Прибытие Таноса

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

Когда Халк прибыл на Землю, он сообщил Доктору Стрэнджу время прибытия на землю Таноса. Однако, Доктор не уверен, что после такого падения Халк правильно запомнил эту дату. Однако он уверен, что набор цифр, который сказал Халк, точно верный. Теперь ему нужно понять, в какие даты может прибыть Танос.

Помогите ему. По дате, которую сообщил Халк, посчитайте все возможные корректные даты, которые можно получить из нее перестановкой цифр. Дата называется корректной, если ее год положительный, месяц не больше 12, а номер дня не больше количества дней в этом месяце. В високосном году в феврале на один день больше — 29 дней вместо 28. Год называется високосным, если его номер делится на 400, или если его номер делится на 4, но не делится на 100.

### Формат входных данных

В первой строке входных данных задана дата в формате ГГГГ ММ ДД. Гарантируется, что это корректная дата с учетом високосных годов.

### Формат выходных данных

В первой строке выведите количество дат, которые можно получить перестановкой цифр из исходной даты. Далее, по одной в строке, выведите сами эти даты в таком же формате, в порядке от самой ранней до самой поздней.

### Пример

| стандартный ввод | стандартный вывод  |
|------------------|--|
| 0001 01 01       | 16<br>0001 01 01<br>0001 01 10<br>0001 10 01<br>0001 10 10<br>0010 01 01<br>0010 01 10<br>0010 10 01<br>0010 10 10<br>0100 01 01<br>0100 01 10<br>0100 10 01<br>0100 10 10<br>1000 01 01<br>1000 01 10<br>1000 10 01<br>1000 10 10 |

## Задача В. Зал брони

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

Тони Старк обустривает новый зал брони, который представляет из себя прямую, в некоторых точках которой находятся отсеки для железных костюмов.

Тони хочет выбрать точку, над которой нужно сделать люк для вылета. Костюмы самостоятельно прилетают к Старку, но он не любит ждать, поэтому люк должен быть расположен в такой целой точке, чтобы суммарное расстояние между ней и всеми костюмами было наименьшим. При этом если таких точек существует несколько, Старк хочет выбрать ту, в координата которой наименьшая.

Напишите программу для Джарвиса, которая сможет определить оптимальную точку для размещения люка.

### Формат входных данных

В первой строке задано одно натуральное число  $n$  — число точек, на которых находятся железные костюмы ( $1 \leq n \leq 10^5$ ).

Во второй строке содержатся  $n$  целых чисел  $a_i$  — количество костюмов в  $i$ -м отсеке ( $1 \leq a_i \leq 100$ ).

Во второй строке содержатся  $n$  целых чисел  $x_i$  — координата  $i$ -го отсека ( $1 \leq x_i \leq 10^9$ ). В одной координате может быть более одного отсека для костюмов. Гарантируется, что для любого  $i$  выполняется  $x_{i-1} \leq x_i$ .

### Формат выходных данных

Выведите одно целое число — оптимальную точку для размещения люка.

### Примеры

| стандартный ввод            | стандартный вывод |
|-----------------------------|-------------------|
| 5<br>1 1 1 1 1<br>1 2 3 4 5 | 3                 |
| 4<br>1 4 1 1<br>1 2 3 3     | 2                 |

## Задача С. Ресторан

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

Тор, Ракета и Грут ходят по торговому центру и выбирают ресторан, в котором они поужинают. У каждого из них свои предпочтения, Тору нравятся рестораны с большим выбором стейков, Ракете нравятся бургеры, а Грут предпочитает пиццу. Так или иначе, каждый из них выставил все рестораны (с номерами от 1 до  $n$ ) в порядке своих приоритетов.

Как же им выбрать ресторан? Будет неловко, если они пойдут в какой-то ресторан, а кафе напротив будет более предпочтительно и у Ракеты, и у Грута. А кафе может оказаться хуже другой забегаловки по мнению Тора и Ракеты... Они решили выбрать ресторан, который не будет уступать никакому другому ресторану.

Более формально, нужно найти такой ресторан, что любой другой ресторан будет менее предпочтителен у не менее чем двух Мстителей. Помогите им найти такой ресторан, или скажите что его не существует.

### Формат входных данных

В первой строке содержится  $n$  — число ресторанов ( $1 \leq n \leq 100\,000$ ).

Во второй строке содержится перестановка чисел от 1 до  $n$  — список ресторанов в порядке предпочтения у Тора, от более предпочтительного к менее предпочтительному.

В третьей строке в аналогичном формате записаны предпочтения Ракеты, а в четвертой строке — предпочтения Грута.

### Формат выходных данных

Выведите номер самого лучшего ресторана, или  $-1$ , если его не существует.

### Примеры

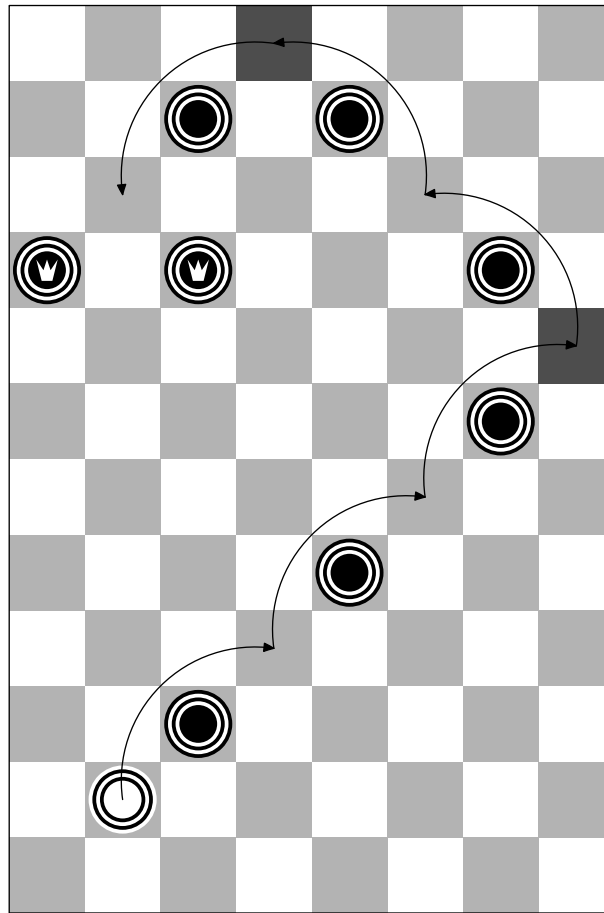
| стандартный ввод                   | стандартный вывод |
|------------------------------------|-------------------|
| 3<br>1 2 3<br>2 3 1<br>3 1 2       | -1                |
| 4<br>2 1 3 4<br>3 1 4 2<br>4 1 2 3 | 1                 |

## Задача D. Достойный финал

|                         |                   |
|-------------------------|-------------------|
| Имя входного файла:     | стандартный ввод  |
| Имя выходного файла:    | стандартный вывод |
| Ограничение по времени: | 4 секунды         |
| Ограничение по памяти:  | 512 мегабайт      |

Халк и Танос, решив, что на сегодня довольно драк, решили сыграть в шашки, чтобы выяснить, кто круче. Однако играют они по особым правилам.

- Поле имеет размеры  $h \times w$ , его строки пронумерованы снизу вверх числами от 1 до  $h$ , а столбцы — слева направо числами от 1 до  $w$ .
- Клетки, у которых номера столбца и строки имеют одинаковую четность, покрашены в черный цвет, а остальные — в белый.
- В некоторых черных клетках находятся белые или черные шашки. Ни в какой клетке не находится более одной шашки.
- Игроки ходят по очереди, за ход можно либо ничего не сделать, либо походить шашкой своего цвета: Халк — белой, Танос — черной.
- Если игрок ходит шашкой, он совершает ею последовательность *взятий*. А именно, пусть шашка игрока находится в клетке  $A$ , в соседней по углу клетке  $B$  находится шашка противника, а следующая в том же направлении клетка  $C$  пустая. Тогда игрок может *взять* шашкой в клетке  $A$  шашку в клетке  $B$ : при этом шашка игрока перемещается с  $A$  на  $C$ , а «перепрыгнутая» шашка противника снимается с поля, то есть клетка  $B$  оказывается пустой. Если после взятия походившая шашка способна взять ещё какую-нибудь шашку противника, то игрок может продолжать совершать *взятия* своей шашкой, пока желает.
- В игре есть дополнительное правило: нельзя за один ход более двух раз менять направление, в котором шашка движется, совершая взятия. Таким образом, в приведенной на рисунке ниже ситуации белая шашка не могла за свой ход взять ни одну из двух шашек, в которых нарисована корона: чтобы взять левую из них, ей необходимо было бы выпрыгнуть из поля, а чтобы взять правую — в третий раз сменить направление прыжков (первые два изменения направления произошли в выделенных клетках).



После нескольких ходов у Халка осталась всего одна пешка, однако сдаваться он не собирается. Помогите ему за один ход взять как можно больше пешек Таноса!

### Формат входных данных

В первой строке находится три целых числа  $h, w, n$  — высота, ширина поля и количество черных пешек ( $1 \leq h, w, n \leq 250\,000$ ).

Во второй строке находится два целых числа  $p, q$  — номера строки и столбца, на пересечении которых находится белая пешка ( $1 \leq p \leq h, 1 \leq q \leq w$ ).

В каждой из следующих  $n$  строк находится по два целых числа  $r_i, s_i$  — номера строки и столбца, на пересечении которых находится  $i$ -я черная пешка ( $1 \leq r_i \leq h, 1 \leq s_i \leq w$ ).

Гарантируется, что ни у какой черной пешки пара координат не совпадает с парой координат другой черной или белой пешки. Гарантируется, что все пешки находятся в клетках черного цвета.

### Формат выходных данных

Выведите одно целое число — наибольшее количество черных пешек, которое за один ход может взять белой пешкой Халк, следуя указанным выше правилам.

### Пример

| стандартный ввод  | стандартный вывод |
|---|-------------------|
| 12 8 8<br>2 2<br>9 1<br>9 3<br>11 3<br>3 3<br>5 5<br>11 5<br>9 7<br>7 7 | 6                 |

## Задача Е. Карточная игра

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

Это интерактивная задача.

В свободное от битв время, Тони Старк любит играть с Вижном в карты. Однако, обычные карточные игры им быстро надоели, и они решили придумать свою собственную.

У них есть колода из  $n$  карт с написанными на них числами от 1 до  $n$ , которые лежат на столе рубашкой вверх. Перед началом игры Вижн тщательно тасует эту колоду, а после этого в тайне от Тони переворачивает произвольные  $k$  карт рубашкой вниз. После этого задача Тони — не смотря на колоду, разделить ее на две непустых так, чтобы в них было поровну карт, лежащих рубашкой вниз. Для этого Тони (участник) может делать следующие действия:

- *Reverse* — перевернуть карту, лежащую наверху колоды;
- *Move* — отложить карту, лежащую наверху колоды, в отдельную колоду;
- *Up  $i$*  — переместить карту, находящуюся на  $i$ -м месте, считая сверху, наверх колоды. После этого верхние  $i - 1$  карт смещаются вниз;
- *Finish* — сообщение о том, что больше никаких действий с колодой производиться не будет. Это сообщение предполагает, что среди оставшихся в колоде карт и отложенных количество карт, лежащих рубашкой вниз, одинаково.

В ответ на каждое действие Вижн (жюри) сообщает Тони количество карт в отложенной колоде, лежащих рубашкой вниз.

Тони понимает, что Вижн намного умнее него благодаря камню Бесконечности, однако, все равно хочет победить. Поэтому он попросил вас помочь ему победить в этой игре.

### Протокол взаимодействия

Во время взаимодействия вашей программы с программой жюри происходит следующее: сначала на вход вашей программе дается два числа  $n$  и  $k$  ( $1 \leq n \leq 100$ ,  $0 \leq k \leq n$ ;  $n, k$  — четные), а затем несколько раз повторяются следующие действия:

- Ваша программа сообщает программе жюри одно из четырех действий, описанных в условии;
- Если это одно из первых трех действий, программа жюри сообщает количество карт, лежащих рубашкой вниз в отложенной колоде после выполнения этого действия;
- Если же это действие *Finish*, программа жюри проверяет, что количество карт, лежащих рубашкой вниз, в обоих колодах совпадает и завершает взаимодействие с вашей программой (соответственно, после этого действия и ваша программа должна завершиться).

Вашей программе разрешается выполнить не более 1000 действий.

### Замечание

После каждого действия вашей программы выводите символ перевода строки. Если вы используете «`writeln`» в Паскале, «`cout << ... << endl`» в C++, «`System.out.println`» в Java или «`print`» в Python, сброс потока вывода у вас происходит автоматически, дополнительно делать «`flush`» не обязательно. Если вы используете другой способ вывода, рекомендуется делать «`flush`», но все равно обязательно требуется выводить символ перевода строки.

Ниже приведены наиболее типичные причины получения тех или иных сообщений об ошибке.

Если ваша программа соблюдает протокол, но в итоге сдается, хотя изначально можно было выиграть, вы получите вердикт «Wrong Answer».

Если ваша программа выводит некорректно отформатированные сообщения программе жюри, то вы получите результат «Presentation Error» либо «Wrong Answer».

Если ваша программа нарушила протокол и ждет ввода в то же время, когда его ждет и программа жюри, то вы получите результат «Idleness Limit Exceeded». Обратите внимание, что к такому же результату может привести и то, что вы не переводите строку после каждого выведенного сообщения или выводите не тем способом, который описан в начале раздела, и не делаете «flush».

## Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 6 4              | Move              |
| 1                | Reverse           |
| 1                | Move              |
| 1                | Up 4              |
| 1                | Reverse           |
| 1                | Move              |
| 2                | Finish            |

## Замечание

В тестовом примере представлено возможное развитие событий:

- В колоде 6 карт, из них 4 рубашкой вниз;
- Откладываем верхнюю карту в отдельную колоду, она оказывается рубашкой вниз, в ответе жюри сообщает, что теперь в отложенной колоде одна карта лежит рубашкой вниз (соответственно, в колоде осталось 3 карты рубашкой вниз);
- Переворачиваем новую верхнюю карту в колоде;
- Теперь перекладываем ее в отдельную колоду, но она оказывается рубашкой вверх, возвращаемое значение не меняется (в колоде осталось 2 карты рубашкой вниз, так как до переворачивания верхняя карта лежала рубашкой вниз);
- Переносим четвертую (последнюю в текущей колоде) карту наверх;
- Переворачиваем ее;
- Откладываем ее в сторону, она оказывается рубашкой вниз. Теперь слева осталось 2 карты рубашкой вниз, и справа тоже (как нам только что сообщило жюри). Поэтому, можно заканчивать игру, мы победили.

## Задача F. Просчет событий

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

Ожидая Таноса на Титане, Доктор Стрэндж не терял время зря — он сел просчитывать вероятность победы в войне с помощью Глаза Агамотто, содержащего камень времени, пятый камень Бесконечности. Для этого он выделил  $n$  действий, которые могут сделать Мстители и  $q$  действий, про которые надо узнать, можно ли их сделать. Каждое действие Стрэндж обозначил числом, уникально описывающим его — действия, которые Мстители могут сделать, он обозначил числами  $a_1, a_2, \dots, a_n$ , а действия, про которые надо узнать возможность их выполнения —  $b_1, b_2, \dots, b_q$ .

Маг знает, что временной континуум устроен так, что если можно сделать действие, обозначенное числом  $x$  и действие, обозначенное числом  $y$ , то можно сделать и действия, обозначенные числами  $x \vee y$  и  $x \wedge y$  (где  $\vee$  и  $\wedge$  — побитовые операции «или» и «и» соответственно). Поэтому теперь про каждое из событий  $b_1, b_2, \dots, b_q$  осталось понять, можно ли их получить описанным выше способом. Помогите Стрэнджу справиться с этим заданием, ведь времени до прибытия Таноса на Титан осталось совсем немного. Обратите внимание, что одно действие можно совершать любое количество раз.

### Формат входных данных

В первой строке входного файла содержится число  $n$  — количество действий, которые могут выполнить Мстители ( $1 \leq n \leq 100\,000$ ).

В следующей строке содержится  $n$  чисел  $a_i$  — числа, описывающие эти действия ( $0 \leq a_i \leq 10^9$ ). Гарантируется, что все числа попарно различны.

В третьей строке содержится число  $q$  — количество действий, про которые надо узнать их возможность выполнения ( $1 \leq q \leq 100\,000$ ).

В последней строке содержится  $q$  чисел  $b_j$  — числа, описывающие эти действия ( $0 \leq b_j \leq 10^9$ ).

### Формат выходных данных

В  $i$ -й строке выходного файла выведите «YES», если действие, описанное числом  $b_i$ , можно выполнить и «NO» в противном случае.

### Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 3                | YES               |
| 1 3 4            | NO                |
| 6                | YES               |
| 1 2 3 4 5 6      | YES               |
|                  | YES               |
|                  | NO                |

### Замечание

Числа 1, 3 и 4 можно получить не задействуя операций  $\vee$  и  $\wedge$ , а  $5 = 4 \vee 1$ .