

Задача А. Буквы на заказ

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

Плотник Иван Семенович Бобров делает на заказ любую мебель. Но в городе он славится тем, что изготавливает из дерева на заказ прописные буквы латинского алфавита. Из таких букв очень модно делать вывески для магазинов, кафе, торговых центров.

В целом Иван Семенович любит изготавливать буквы, но не все буквы делать одинаково приятно. Дело в том, что есть буквы, в которых нужно выдалбливать отверстия, а это нужно делать очень аккуратно, чтобы изделие не треснуло. Например, в буквах «С» и «Е» нет отверстий, у букв «А» и «D» есть одно отверстие, а у буквы «В» — два отверстия.

Сегодня Иван Семенович получил очередной заказ на изготовление прописной буквы латинского алфавита. Ему интересно, сколько отверстий придется делать для изготовления этой буквы?

Формат входных данных

В единственной строке дана единственная заглавная буква латинского алфавита.

Формат выходных данных

В единственной строке выведите единственное целое число — количество отверстий, которое придется сделать для изготовления этой буквы.

Примеры

стандартный ввод	стандартный вывод
A	1
B	2
C	0

Замечание

Список всех строчных букв латинского алфавита выглядит следующим образом:

A, B, C, D, E, F, G, H, I, J, K, L, M,
N, O, P, Q, R, S, T, U, V, W, X, Y, Z.

Задача В. Бариста

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

Игорь хочет научиться варить самый вкусный в мире кофе, как настоящий бариста. Сейчас он изучает процесс приготовления капучино. Основными ингредиентами капучино являются эспрессо и молоко. Игорь считает, что у него получилось сделать правильное капучино, если отношение молока к эспрессо лежит в пределах от $3 : 1$ до $5 : 1$ включительно. Например, если в напитке 45 миллилитров молока и 10 миллилитров эспрессо, то Игорь считает, что у него получилось капучино, потому что выполнено условие $3 : 1 \leq 45 : 10 \leq 5 : 1$.

Если отношение молока к эспрессо строго меньше $3 : 1$, то Игорь считает, что у него получилось макиато. Например, если в напитке 29 миллилитров молока и 10 миллилитров эспрессо, то Игорь считает, что у него получилось макиато, потому что выполнено условие $29 : 10 < 3 : 1$.

Если же отношение молока к эспрессо строго больше $5 : 1$, то Игорь считает, что у него получилось латте. Например, если в напитке 53 миллилитров молока и 10 миллилитров эспрессо, то Игорь считает, что у него получилось латте, потому что выполнено условие $5 : 1 < 53 : 10$.

Сегодня утром Игорь приготовил себе напиток, в котором содержится a миллилитров молока и b миллилитров эспрессо. Определите, какой напиток получился у Игоря.

Формат входных данных

В единственной строке через пробел даны два целых числа a и b ($1 \leq a, b \leq 10^5$) — количество молока и количество эспрессо в напитке Игоря.

Формат выходных данных

В единственной строке выведите название напитка, который получился у Игоря, исходя из критериев данных в условии.

Если у Игоря получилось макиато, выведите слово «macchiato», без кавычек.

Если у Игоря получилось капучино, выведите слово «cappuccino», без кавычек.

Если у Игоря получилось латте, выведите слово «latte», без кавычек.

Примеры

стандартный ввод	стандартный вывод
29 10	macchiato
45 10	cappuccino
53 10	latte

Замечание

Тестовые примеры соответствуют всем возможным напиткам и разобраны в условии задачи.

Задача С. Посудомойка

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

Я женщина, а не посудомойка!

Алина учится на отлично, участвует в олимпиадах, занимается спортом. К сожалению, это никак не спасает её от «*Дочь, помой посуду, пожалуйста!*»

Вчера у мамы Алины был день рождения, на котором гости подарили набор из n тарелок. Каждая тарелка имеет свой уникальный красивый рисунок и отличается от остальных. Для простоты будем считать, что тарелки занумерованы числами $1, 2, \dots, n$.

Сегодня утром Алина увидела, что после вчерашней встречи никто не помыл посуду, поэтому n грязных подаренных тарелок разложены в k стопок на кухне. Известно, что сегодня к маме снова придут гости, а Алине придётся помогать их обслуживать. В течение дня у мамы будет q просьб двух видов:

1. «*Дай, пожалуйста, тарелку i* » — в дальнейшем будем обозначать такую просьбу 1 i .

На эту просьбу Алине надо будет поставить на праздничный стол тарелку с номером i . В этот момент тарелка i должна быть **чистой**.

2. «*Убери, пожалуйста, тарелку i* » — в дальнейшем будем обозначать такую просьбу 2 i .

На эту просьбу Алине надо будет убрать со стола тарелку с номером i в одну из k стопок. Считаем, что в этот момент тарелка i была использована и стала **грязной**. Тарелку можно убрать только **наверх** стопки.

Понятно, что для того, чтобы выполнить просьбу типа 1, тарелку i необходимо в какой-то момент **помыть**. Опишем процесс мытья посуды подробнее:

- Алина может взять тарелку сверху из какой-то стопки и помыть её.
- Брать тарелку из середины стопки нельзя — по неосторожности можно случайно уронить стопку и разбить посуду (мама будет очень недовольна).
- Нельзя перекладывать тарелку из стопки в стопку — если мама увидит это, то вставит неуместный комментарий про неэффективность Алины в таком простом деле, как мытьё посуды.
- Помытую тарелку Алина складывает в шкаф с чистой посудой. Шкаф может вместить одновременно все n чистых тарелок.

Алина очень хочет как можно больше времени посвятить саморазвитию. Она просит помочь **минимизировать количество помытых в процессе тарелок**. Каждый случай мытья тарелки учитывается в ответе отдельно.

Напишите программу, которая по заданным просьбам мамы подскажет Алине оптимальный алгоритм действий.

Формат входных данных

В первой строке записаны 3 целых числа n, k, q ($1 \leq n, k, q \leq 10^5$) — количество тарелок, количество стопок с посудой, количество просьб мамы соответственно.

В следующих k строках идёт описание стопок. В строке $r + 1$ описывается стопка с номером r . Первое число в строке s_r — количество тарелок в стопке r . Далее идут s_r чисел — номера тарелок,

лежащих в стопке (в порядке снизу вверх, то есть последнее число — номер тарелки, лежащей сверху стопки).

Гарантируется, что:

- $s_1 + s_2 + \dots + s_k = n$ — все тарелки лежат в стопках грязной посуды.
- все тарелки во всех стопках различны — каждая тарелка лежит ровно в одной стопке ровно в одном экземпляре.

Далее следуют q просьб мамы, в следующем формате: $t i$ ($1 \leq t \leq 2, 1 \leq i \leq n$).

- t — тип просьбы (1 — поставить тарелку на стол, 2 — убрать тарелку со стола);
- i — номер тарелки, с которой надо провести действие.

Гарантируется, что просьбы мамы **не противоречивы**, то есть не бывает такого, что мама просит поставить на стол тарелку, которая уже там стоит или убрать со стола тарелку, которой на нём нет.

Формат выходных данных

В первой строке выведите a — минимальное количество помытых в процессе тарелок. Каждый случай мытья тарелки учитывается в ответе отдельно.

В следующих $a + q$ строках выведите действия Алины. Каждое действие в следующем формате:

- 1 — поставить на стол чистую тарелку. В этот момент первая невыполненная просьба мамы должна быть вида $(1, i)$, а тарелка i должна находиться в шкафу с чистой посудой.
- 2 j ($1 \leq j \leq k$) — взять со стола грязную тарелку i и положить её наверх стопки с номером j . В этот момент первая невыполненная просьба мамы должна быть вида $(2, i)$.
- 3 j ($1 \leq j \leq k$) — взять из стопки с номером j верхнюю тарелку (в этот момент стопка j должна содержать в себе хотя бы одну тарелку), помыть её и убрать в шкаф с чистой посудой.

В последовательности действий должно быть ровно a действий 3-го типа.

Если существует несколько оптимальных последовательностей действий Алины — выведите любую из них.

Пример

стандартный ввод	стандартный вывод
3 3 4	3
1 2	3 2
2 1 3	3 2
0	1
1 1	1
1 3	2 3
2 1	3 1
1 2	1

Замечание

Опишем действия в примере:

1. Хотим поставить тарелку номер 1 на стол, чтобы выполнить первую просьбу мамы, но не можем этого сделать, потому что тарелка 1 не лежит на вершине стопки. Поэтому сначала моем и убираем в шкаф тарелку номер 3 из 2-й стопки.
2. Тарелка номер 1 лежит на вершине 2-й стопки, моем её и убираем в шкаф.

3. Из шкафа выставляем чистую тарелку номер 1 на стол.
4. Затем хотим выставить на стол тарелку номер 3. Она уже лежит в шкафу, поэтому можем сразу её выставить.
5. Надо убрать тарелку номер 1 со стола. Положим её в стопку номер 3. Аналогично можно было положить в пустую стопку номер 2.
(Неоптимально убирать тарелку в стопку номер 1 — в дальнейшем нам понадобится тарелка номер 2 из той стопки, но тарелка номер 1 «перекроет» к ней доступ).
6. Далее надо выставить на стол тарелку номер 2, но она грязная. Поэтому помоем её и уберём в шкаф.
7. Теперь выставляем чистую 2-ю тарелку на стол.

Задача D. Королевства и союзы

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

Демидург Слава создал одномерный мир из n королевств на прямой, причем i -е королевство находится на позиции i в мире.

Все люди верят в Славу, но воспринимают его по-разному, поэтому религии различных королевств могут различаться. В дальнейшем каждой уникальной религии будет соответствовать уникальное неотрицательное целое число. На момент создания мира восприятие людьми Славы было одинаковым, поэтому все n королевств имели религию 0.

Королевства могут объединяться в союзы. Слава не хочет, чтобы у королевств в союзе возникали разногласия, поэтому у всех в союзе должно быть одинаковое вероисповедание.

С другой стороны, если королевства так близки друг другу по духу, то что им мешает объединиться в одно королевство? Славе не хотелось бы такого, так как достаточно сильное королевство может начать войны с королевствами других религий.

В то же время, если расстояние между союзниками будет слишком большим, то становится сложно поддерживать контакт.

Осознав всё это, Слава решил ввести следующие правила относительно союзов:

1. союз состоит минимум из 2 королевств;
2. все королевства в союзе имеют одну и ту же религию;
3. соседние в союзе королевства находятся строго на расстоянии k , ни больше ни меньше. Расстояние между королевствами i и j равно $|i - j|$.

Рассмотрим пример: пусть $n = 6$, $k = 2$ и вероисповедания королевств равны $\{1, 2, 1, 3, 1, 3\}$. В таком случае возможны только следующие союзы:

1. $\{1, 3\}$
2. $\{1, 3, 5\}$
3. $\{3, 5\}$
4. $\{4, 6\}$

Обратите внимание, что союз $\{1, 5\}$ невозможен, так как расстояние между королевствами равно 4, что не равно $k = 2$.

Иногда королевства переосмысливают свои ценности или восприятие мира и меняют религию. Это может приводить к изменению ситуации с возможными союзами.

К примеру, если в описанном выше примере королевство 6 изменит свою веру с 3 на 2, то пропадёт возможность союза $\{4, 6\}$, так как теперь у этих королевств будут различные религии.

Если же после этого королевство 4 тоже изменит религию на 2, то станут возможными сразу три союза:

1. $\{2, 4\}$
2. $\{2, 4, 6\}$
3. $\{4, 6\}$

Слава — очень занятой демиург, поэтому просит вас — ангела-стажёра — помочь ему и после каждого изменения каким-либо королевством своей религии сообщать, сколько различных союзов возможно в данный момент в описанном мире.

Формат входных данных

В первой строке записано три числа n , k и q ($1 \leq k \leq n \leq 2 \cdot 10^5$, $1 \leq q \leq 10^5$) — количество королевств, разрешенное для союза расстояние и количество изменений вероисповедания.

Далее в q строках вводятся по два числа m_i , r_i ($1 \leq m_i \leq n$, $0 \leq r_i \leq 10^9$) — номер королевства и его новое вероисповедание. Возможны ситуации, когда новое вероисповедание совпадает со старым (это уже вопросы политики, не рассматриваемые в рамках данной задачи).

Начальная религия для каждого из n королевств равна 0.

Формат выходных данных

Выведите q целых чисел a_i — количество возможных союзов в мире Славы после принятия королевством m_i религии r_i .

Пример

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

Замечание

Тестовый пример соответствует описанному в условии примеру — первые 6 запросов каждое королевство изменяет свою религию с 0, после чего происходят 2 описанных в примере изменения.

Рассмотрим вероисповедания королевств и возможные союзы после каждого изменения религии:

1. $\{1, 0, 0, 0, 0, 0\}$:

- $\{3, 5\}$
- $\{2, 4\}$
- $\{2, 4, 6\}$
- $\{4, 6\}$

2. $\{1, 0, 1, 0, 0, 0\}$:

- $\{1, 3\}$
- $\{2, 4\}$
- $\{2, 4, 6\}$
- $\{4, 6\}$

3. $\{1, 2, 1, 0, 0, 0\}$:

- $\{1, 3\}$
- $\{4, 6\}$

4. $\{1, 2, 1, 0, 0, 3\}$:

- $\{1, 3\}$

5. $\{1, 2, 1, 0, 1, 3\}$:

- $\{1, 3\}$

- {1, 3, 5}
- {3, 5}

6. {1, 2, 1, 3, 1, 3}:

- {1, 3}
- {1, 3, 5}
- {3, 5}
- {4, 6}

7. {1, 2, 1, 3, 1, 2}:

- {1, 3}
- {1, 3, 5}
- {3, 5}

8. {1, 2, 1, 2, 1, 2}:

- {1, 3}
- {1, 3, 5}
- {2, 4}
- {2, 4, 6}
- {3, 5}
- {4, 6}

Задача Е. Стикеры

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

Организаторы олимпиады «Технологичные Когнитивы» обещали стикеры тем, кто позовет друзей участвовать в соревновании. Каждый участник при регистрации мог указать ФИО друга, который уговорил его принять участие в олимпиаде (а мог и не указать никого).

Если двое или более друзей, указавших вас как «пригласившего», зарегистрировались строго позже вас и решили хотя бы одну задачу, вы получаете заветные стикеры.

Пока жюри подводит итоги самой олимпиады, вопрос «Ну что там по стикерам?» звучит из каждого утюга, поэтому помогите жюри и найдите список всех участников, получивших набор стикеров.

Формат входных данных

В первой строке вводится число N ($1 \leq N \leq 3 \cdot 10^5$) — число участников. Для удобства участники пронумерованы целыми числами от 1 до N .

Во второй строке записаны N чисел T_i ($1 \leq T_i \leq 10^9$) — время регистрации i -го участника.

В третьей строке записаны N чисел P_i ($0 \leq P_i \leq 12$) — количество задач, решенных участником i .

В четвертой строке записаны N чисел F_i ($0 \leq F_i \leq N$) — номер друга, которого i -й участник отметил как «пригласившего». Если $F_i = 0$, то i -й участник никого не указал в таком статусе.

Формат выходных данных

В первой строке выведите одно целое число K ($0 \leq K \leq N$) — количество участников, которые должны получить стикеры.

Во второй строке через пробел выведите K целых чисел S_i ($1 \leq S_1 \leq S_2 \leq \dots \leq S_K \leq N$) — номера получивших стикеров участников в возрастающем порядке.

Если $K = 0$, то на второй строке не надо выводить ничего.

Примеры

стандартный ввод	стандартный вывод
9 2 3 1 4 5 9 8 7 6 12 11 10 9 8 7 6 5 4 1 1 1 2 2 9 9 9 9	2 2 9
5 1 2 10 11 5 6 7 0 9 10 5 5 5 5 0	0

Замечание

Рассмотрим детально первый тестовый пример.

- Первый участник указал себя в качестве друга, соответственно он зарегистрировался НЕ позже себя — не учитывается при распределении.
- Второй участник указал первого как друга, а также зарегистрировался после участника 1, значит, один приглашенный у первого есть.
- Третий участник указал первого как друга, но зарегистрировался раньше, чем он, поэтому по правилам участник 3 так же не учитывается при распределении.
- Четвертый и пятый участники указали второго как друга и зарегистрировались после него. Засчитываем участнику 2 двух приглашенных друзей.

- Шестой, седьмой и восьмой участники указали девятого участника пригласившим и зарегистрировались после него. Участнику 9 засчитали трех приглашенных друзей.
- Сам девятый участник указал себя в качестве друга, соответственно он зарегистрировался НЕ позже себя — не учитывается при распределении.

Итого, первому участнику «засчитали» одного друга, второму — двух, девятому — трёх. Все засчитанные участники решили хотя бы одну задачу, поэтому дают право участникам 2 и 9 получить стикеры.

Рассмотрим второй тестовый пример.

- Первый и второй участники зарегистрировались до пятого, значит ему «в зачет» они не идут.
- Третий участник зарегистрировался позже пятого, но не решил ни одной задачи, поэтому его голос учитываться не будет.
- Лишь четвертый участник указал пятого, зарегистрировался позже него и решил хотя бы одну задачу. Поэтому только четвертый участник и будет засчитан пятому при распределении стикеров.
- Сам пятый участник не указал вообще никого в анкете — вероятно он узнал об олимпиаде не от своего друга.

По итогу пятый участник не набрал необходимых двух приглашенных друзей, поэтому ответ на данный тестовый пример равен 0.

Задача F. Прыгай вперед!

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

Федот решил пройти уровень в своей любимой игре «Прыгай вперед!». Уровень представляет из себя n последовательных клеток, которые пронумерованы числами от 1 до n . За посещение клетки с номером i необходимо заплатить a_i тугриков. У каждой клетки есть выталкивающая сила, которая изначально равна одному (это означает, что из клетки с номером i можно прыгнуть только в клетку с номером $i + 1$). Цель состоит в том, чтобы пройти из клетки номер 1 в клетку номер n , потратив как можно меньше тугриков.

Понятно, что в таких условиях уровень можно пройти единственным образом — последовательно посетить все клетки от первой до последней. А стоимость такого прохождения будет равна сумме стоимостей посещения всех клеток уровня. Но все не так просто, ведь в игре есть $n - 1$ дополнительных карточек, пронумерованных числами от 1 до $n - 1$. На карточке номер i записано число b_i ($1 \leq b_i \leq n - i$), которое означает, что если применить i -ю карточку, то выталкивающая сила клетки с номером i станет равна b_i и после этого из клетки с номером i можно будет прыгнуть только в клетку с номером $i + b_i$. Примененная один раз карточка действует **на все последующие прохождения** уровня.

Федот решил применить q ($1 \leq q \leq n - 1$) карточек. Он будет применять их в определенном порядке и после каждого такого действия заново вычислять минимальное количество тугриков, которое нужно потратить, чтобы пройти из клетки номер 1 в клетку номер n . Можно доказать, что как бы мы не применяли карточки, всегда найдется путь из клетки номер 1 в клетку номер n .

И тут Федот понял, что после применения очередной карточки пересчитать минимальную стоимость прохождения не так и просто. Поэтому эту задачу он поручил вам.

Формат входных данных

В первой строке входных данных дано целое число n ($2 \leq n \leq 10^5$) — количество клеток.

Во второй строке через пробел даны n целых чисел a_i ($1 \leq a_1, \dots, a_n \leq 10^9$) — стоимости посещения клеток.

В третьей строке через пробел даны $n - 1$ целых чисел b_i ($1 \leq b_i \leq n - i$) — числа, записанные на карточках.

В четвертой строке дано целое число $1 \leq q \leq n - 1$ — количество карточек, которые применит Федот.

В пятой строке через пробел даны q уникальных целых чисел c_i ($1 \leq c_1, \dots, c_q \leq n - 1$; $c_i \neq c_j$ для $1 \leq i \neq j \leq q$) — номера карточек, которые последовательно применит Федот.

Формат выходных данных

Требуется вывести q строк. В k -й строке ($1 \leq k \leq q$) должна быть выведена минимальная стоимость прохождения уровня после применения карточек с номерами c_1, \dots, c_k .

Пример

стандартный ввод	стандартный вывод
4	1011
1 10 100 1000	1101
2 2 1	1101
3	
2 1 3	

Замечание

Рассмотрим детально тестовый пример:

- Изначально путь пролегает через 4 клетки 1 - 2 - 3 - 4 со стоимостями 1, 10, 100, 1000 соответственно — суммарная стоимость получается равной 1111. Обратите внимание, что данную стоимость выводить **не нужно**.
- Далее Федот применяет карточку 2, после чего с клетки 2 прыжок переносит сразу на клетку $2 + 2 = 4$ (так как на карточке было написано число 2). В получившейся конфигурации путь пролегает через клетки 1 - 2 - 4, а его стоимость соответственно равна 1011.
- Следующим действием применяется карточка 1, что продлевает прыжок с клетки 1 сразу на клетку $1 + 2 = 3$ (так как на карточке аналогично было написано число 2). В новой схеме уровня путь пролегает уже через клетки 1 - 3 - 4, а его стоимость равна 1101.
- Последней карточкой применяется 3-я. Но так как она была равна 1, то прыжок с клетки 3 не изменился и до сих пор ведет на клетку $3 + 1 = 4$. Стоимость пути тоже не изменилась и составляет 1101.

Задача G. Полив... «Ой!»

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

Алиса пишет компьютерную игру «Полив». В игре на клумбе растёт n цветков в ряд, каждый из которых может быть полит или не полит. Таким образом, клумбу можно представить в виде массива F из нулей и единиц длины n , где $F_i = 1$ означает, что цветок i не полит, а $F_i = 0$ означает, что цветок i полит.

За одну операцию игрок может выбрать позицию i такую, что $F_i = 1$ и полить цветок i (то есть F_i станет равно 0).

Пока Алиса писала код игры, она часто отвлекалась на общение со своим любимым преподавателем по программированию. Поэтому для неё большой неожиданностью стало то, что в какой-то момент времени (она не поняла, в какой именно) полив цветка стал «странным». Теперь, если полить цветок i , то несколько политых цветков слева и справа от цветка i станут не политыми.

Более формально эту «странность» можно описать так: когда игрок выбирает позицию i для которой $F_i = 1$, в программе Алисы выбираются какие-нибудь границы l и r ($1 \leq l \leq i \leq r \leq n$) при которых для любого $j \in [l, r]$ выполняется $F_j = 1 \Leftrightarrow j = i$ (то есть на отрезке $[l, r]$ все цветы политы, кроме цветка на позиции i). И после полива цветка i этот цветок станет политым, а все остальные цветы на отрезке $[l, r]$ станут не политыми.

«Ой» — вскрикнула Алиса, как только заметила «странность». Конечно же, вместо того, чтобы исправить «странность», Алиса начала случайно выбирать цветы, поливать их и смотреть, как изменяется клумба. Спустя какое-то время Алиса заскучала, она остановилась на клумбе в состоянии A . Теперь ей стало интересно, можно ли из данного состояния получить состояние B . И если да, то как это может произойти.

Формат входных данных

В первой строке записано одно целое число n ($1 \leq n \leq 100$) — количество цветов на клумбе.

Во второй строке записано состояние клумбы, которое есть на данный момент у Алисы — строка A длины n , состоящая из нулей и единиц.

В третьей строке записано состояние клумбы, которое хочет получить Алиса — строка B длины n , состоящая из нулей и единиц.

Формат выходных данных

Если из состояния A невозможно получить состояние B за не более чем $4 \cdot n + 1$ полив, то в единственной строке выведите «NO» (без кавычек).

Если из состояния A можно получить состояние B , то в первой строке выведите «YES» (без кавычек).

Во второй строке выведите M ($M \leq 4 \cdot n + 1$) — необходимое количество поливов, чтобы из состояния A получить состояние B .

Обратите внимание, что минимизировать количество поливов **не требуется**.

В каждой из следующих M строк выведите по 3 числа: $i \ l \ r$:

- $[l, r]$ ($1 \leq l \leq r \leq n$) — отрезок, который должен быть выбран программой Алисы;
- i ($l \leq i \leq r$) — номер цветка, который надо полить на выбранном отрезке. Цветок i должен быть единственным не политым на отрезке $[l, r]$ ($F_j = 1 \Leftrightarrow j = i$ для $l \leq j \leq r$).

Если существует несколько возможных последовательностей поливов, преобразующих A в B , то выведите любую из них.

Пример

стандартный ввод	стандартный вывод
10	YES
1001010010	3
1110111110	4 2 5
	9 9 10
	10 7 10

Замечание

Рассмотрим, как будет меняться строка после каждой операции в примере:

1. 1001010010 → 1110110010
2. 1110110010 → 1110110001
3. 1110110001 → 1110111110

Задача Н. Дорога в школу.

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

Это интерактивная задача.

В центре города Глубокоозёрный находится круглое и глубокое озеро. Вокруг озера проложена кольцевая дорога, которая представляет из себя окружность. Вдоль дороги расположены n домов таким образом, что расстояние между соседними домами составляет ровно один километр. Все дома пронумерованы последовательно числами от 1 до n , то есть дома i и $i + 1$ являются соседними при $1 \leq i < n$, а также дома n и 1 являются соседними. В одном из домов располагается единственная в городе школа, а еще в одном из домов живет мальчик Пахом. Каждый будний день Пахом ходит из дома в школу вдоль дороги по кратчайшему пути (если такой путь не единственный, то Пахом выбирает любой из кратчайших путей), пусть длина этого пути равна len километров. Можно заметить, что всего есть два возможных пути из дома Пахома в школу, так как улица только одна и представляет из себя окружность.

Иногда в городе меняют асфальт на дороге и в таком случае дорогу перекрывают на участке между двумя соседними домами. Причем перекрытым может быть только один участок, чтобы из любого дома можно было дойти до любого другого дома по дороге. Участки дороги пронумерованы таким образом, что участок между домами i и $i + 1$ имеет номер i ($1 \leq i < n$), а участок между домами n и 1 имеет номер n .

Можно заметить, что если один участок дороги перекрыт, то остается единственный путь из дома Пахома в школу, вне зависимости от того какой именно участок перекрыт. В таком случае Пахом идет этим единственным путем. Этот путь может оказаться длиннее чем len километров, поэтому Пахом не очень любит, когда в его городе меняют асфальт.

Ваша задача состоит в том, чтобы определить длину len . Для этого вы можете сделать не более трех запросов первого типа:

? i ($1 \leq i \leq n$) — узнать длину пути от дома Пахома до школы в случае, если перекрыт участок дороги номер i .

После этого следует вывести запрос второго типа, который сообщает ответ на задачу:

! len

После этого запроса необходимо завершить программу.

Формат входных данных

Первая строка входных данных содержит целое число n ($3 \leq n \leq 10^5$) — количество домов в городе.

Формат выходных данных

Чтобы вывести ответ на задачу, выведите его в формате ! len , где len — длина кратчайшего пути из дома Пахома в школу при условии, что ни один из участков дороги не перекрыт.

Протокол взаимодействия

Чтобы задать запрос первого типа, выведите ? i ($1 \leq i \leq n$), где i — номер участка дороги, который вы хотите перекрыть. После вывода запроса необходимо **вывести перевод строки** и сделать операцию «flush».

После каждого запроса типа ? считайте одно целое число val — длину пути из дома Пахома в школу при условии, что перекрыт участок дороги с номером i .

Обратите внимание, что вы можете сделать не более 3 запросов типа ?.

Чтобы сообщить, что ответ найден, требуется вывести ! len , где len — длина кратчайшего пути из дома Пахома в школу при условии, что ни один из участков дороги не перекрыт. После этого требуется сделать «flush» и завершить программу.

Запрос на вывод ответа не входит в ограничение на 3 запроса.

Для сброса буфера вывода (то есть для операции «flush») сразу после вывода нужно сделать:

- C++: `endl` одновременно и делает перевод строки, и «flush»;
Другие варианты: `cout.flush()`, `cout << flush`.
- Java: `System.out.println()` одновременно и делает перевод строки, и «flush».
Если вы используете не `System.out`, то используйте команду `flush` вашего потока вывода.
- Python: `print` одновременно и делает перевод строки, и «flush»;
Напрямую можно сделать «flush» с помощью `sys.stdout.flush()` (требуется `import sys`).
- Pascal: Выполните `flush(output)` после вывода через `writeln`.

Пример

стандартный ввод	стандартный вывод
3	? 1
2	? 2
1	? 3
1	! 1

Замечание

В тестовом примере показано, как программа взаимодействует с проверяющей системой.

Сначала программа считывает значение n , которое равно трем.

Далее программа задает запрос первого типа ? 1 (перекрыли участок дороги 1 между домами 1 и 2), а затем считывает число 2 — ответ на этот запрос.

Далее программа задает запрос первого типа ? 2 (перекрыли участок дороги 2 между домами 2 и 3), а затем считывает число 1 — ответ на этот запрос.

Далее программа задает запрос первого типа ? 3 (перекрыли участок дороги 3 между домами 3 и 1), а затем считывает число 1 — ответ на этот запрос.

И в конце программа выводит ответ на задачу в виде ! 1 и завершает свою работу.

Почему ответ в данном случае равен 1? Давайте посмотрим, что мы узнали из запросов. Если перекрыть участок дороги 1, то путь будет иметь длину 2, а если перекрыть участок дороги 2 или 3, то путь будет иметь длину 1. Из этого можно сделать вывод, что путь из дома Пахома в школу состоит из участка 1 и имеет длину 1.

Задача I. Башни из спичек

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

Жил-был Вовочка. Однажды дома он нашел спички и от нечего делать решил построить из них Башню, очень особенную. Правда так и не придумал, что именно особенного будет в его Башне.

На следующий день случилось чудо. Папа мальчика работал на Балабановской спичечной фабрике и решил взять его с собой на работу. В кабинете, где он сидел, было полно спичек разных длин. В углу Вовочка заметил книжку с надписью «ГОСТ». Книжка была толстая, нашлись в ней и требования к башне из спичек:

1. Башней называется построение из равных между собой прямоугольников — у всех прямоугольников должны быть одинаковая ширина и одинаковая высота;
2. Этажом башни называется один прямоугольник;
3. Спичка, являющаяся потолком одного этажа, одновременно является полом следующего этажа — к примеру, башня из двух этажей состоит из 7 спичек (3 в ширину и 4 в высоту);
4. Ширина и высота этажа должны совпадать с длиной ровно одной спички, используемой в качестве стены / пола / потолка;
5. Высотой башни называется произведение количества этажей на высоту одного этажа.
6. Корректная башня содержит хотя бы 1 этаж.

Маленький мальчик сразу понял, что его Башня будет самой высокой! Правда тут же у Вовочки возник вопрос — а какую максимальную высоту может иметь построенная им Башня? Мальчик еще не знает, что такое «произведение», поэтому просит вас помочь ему в этом деле.

Помогите Вовочке вычислить максимальную высоту Башни, которая может быть построена из имеющихся у мальчика спичек.

Формат входных данных

В первой строке вводится одно число N ($1 \leq N \leq 200000$) — количество спичек. Во второй строке записаны N целых чисел A_i ($1 \leq A_i \leq 200000$) через пробел — длина i -й спички.

Формат выходных данных

Выведите через пробел 3 целых числа — количество этажей, высоту одного этажа и его ширину. Если существует несколько башен с максимальной высотой, выведите любую. Если ни одну башню составить не получается, выведите 3 раза через пробел -1 .

Примеры

стандартный ввод	стандартный вывод
4 5 5 5 5	1 5 5
7 4 4 8 4 8 4 8	1 8 4
5 1 2 3 4 3	-1 -1 -1
10 5 5 5 5 5 5 7 7 7 5	2 5 7
7 1 1 2 3 3 3 3	1 3 1

Замечание

Пояснения к тестам из условия.

1. В тесте 1 можно построить башню, состоящую из одного квадрата 5×5 .
2. Во втором тесте подходят 2 башни. Первая состоит из 2 прямоугольников 4 в высоту, 8 в ширину. Вторая — из одного прямоугольника 8 в высоту, 4 в ширину. Высота обеих башен — 8, а значит любая из них будет являться правильным ответом.
3. В третьем тесте не получается составить ни одного прямоугольника, поэтому ответ -1 .
4. В четвертом тесте оптимальной башней является построение из двух прямоугольников со сторонами 5×7 или двух квадратов 5×5 .
5. В пятом тесте можно построить только следующие башни:
 - один этаж высотой 3 и шириной 1 — оптимальная;
 - один этаж высотой 1 и шириной 3 — неоптимальная;
 - один этаж высотой 3 и шириной 3 — оптимальная.