

Задача 5. Максимальное произведение

Задан массив натуральных чисел $[a_1, a_2, \dots, a_n]$. Весом массива назовём сумму его элементов.

Необходимо разрезать заданный массив на два непустых массива $[a_1, a_2, \dots, a_i]$ и $[a_{i+1}, a_{i+2}, \dots, a_n]$ так, чтобы произведение их весов было как можно больше.

Требуется написать программу, которая по заданному массиву определяет, после какого элемента его необходимо разрезать, чтобы произведение весов получившихся массивов было максимальным.

Формат входных данных

В первой строке входных данных находится целое число n — количество элементов в массиве ($2 \leq n \leq 2 \cdot 10^5$). В следующей строке находятся n целых чисел a_1, a_2, \dots, a_n — элементы массива ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите одно число — номер элемента, после которого необходимо разрезать заданный массив. Если оптимальных вариантов ответа несколько, можно вывести любой из них.

Система оценивания

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Дополнительные ограничения	Необх. подзадачи	Информация о проверке
1	10	$1 \leq n \leq 5000$	Сумма всех a_i не превосходит 10^9		полная
2	10	$1 \leq n \leq 5000$	Все a_i равны		полная
3	20	$1 \leq n \leq 5000$	$a_i \leq 10^9$	1, 2	полная
4	20	$1 \leq n \leq 200000$	Сумма всех a_i не превосходит 10^9	1	полная
5	20	$1 \leq n \leq 200000$	Все a_i равны	2	полная
6	20	$1 \leq n \leq 200000$	$a_i \leq 10^9$	1, 2, 3, 4, 5	полная

Пример

стандартный ввод	стандартный вывод
3 1 2 3	2

Пояснение к примеру

Если сделать разрез после первого элемента, произведение весов равно $1 \cdot (2 + 3) = 5$, а если после второго, то $(1 + 2) \cdot 3 = 9$.

Задача 6. Планировка участка

Учёные планируют участок для испытательного полигона. Участок должен иметь форму прямоугольника $a \times b$, а полигон должен иметь форму прямоугольника $c \times d$. С точными значениями чисел a , b , c и d ученые пока не определились, однако известно следующее:

- Длины сторон a , b , c , d должны быть натуральными числами и выражаться в километрах.
- Для безопасности эксперимента длина и ширина участка должны отличаться от значения x , то есть должны выполняться неравенства $a \neq x$, $b \neq x$.
- Участок будет огражден забором, а полигон должен полностью помещаться внутри участка, то есть должны выполняться следующие условия: $a > c$, $b > d$.
- Площадь участка, не занятого полигоном, должна быть ровно n квадратных километров, то есть должно выполняться следующее условие: $a \cdot b - c \cdot d = n$.

Учёные хотят понять, сколько у них способов выбрать подходящие значения a , b , c и d .

Требуется написать программу, которая по заданным n и x определяет количество способов выбрать числа a , b , c и d так, чтобы все описанные условия выполнялись.

Формат входных данных

В первой строке ввода содержится число n — площадь свободного участка без полигона ($1 \leq n \leq 3000$).

Во второй строке ввода содержится число x — запрещенная длина стороны участка ($0 \leq x \leq 3000$). Значение $x = 0$ означает, что ограничений на длины сторон нет (так как длины сторон должны быть натуральными числами, и, следовательно, больше 0).

Формат выходных данных

В единственной строке выведите количество способов выбрать числа a , b , c и d так, что все описанные условия выполняются.

Система оценивания

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи	Информация о проверке
1	11	$1 \leq n \leq 50, x = 0$		первая ошибка
2	10	$1 \leq n \leq 50$	1	первая ошибка
3	20	$1 \leq n \leq 500, x = 0$	1	баллы
4	22	$1 \leq n \leq 500$	1, 2, 3	баллы
5	17	$1 \leq n \leq 3000, x = 0$	1, 3	баллы
6	20	$1 \leq n \leq 3000$	1, 2, 3, 4, 5	баллы

Примеры

стандартный ввод	стандартный вывод
3 0	1
5 0	5
5 3	2

Пояснение к примеру

В первом тестовом примере подходят только $a = 2, b = 2, c = 1, d = 1$.

Во втором тестовом примере подходят следующие ответы:

- $a = 2, b = 3, c = 1, d = 1$;
- $a = 2, b = 4, c = 1, d = 3$;
- $a = 3, b = 2, c = 1, d = 1$;
- $a = 3, b = 3, c = 2, d = 2$;
- $a = 4, b = 2, c = 3, d = 1$.

В третьем тестовом примере подходят следующие ответы:

- $a = 2, b = 4, c = 1, d = 3$;
- $a = 4, b = 2, c = 3, d = 1$.

В остальных ответах из предыдущего теста либо a , либо b равняется 3.

Задача 7. Банкомат

Ведётся разработка универсального банкомата, который сможет работать с любой денежной системой. Пусть в денежной системе некоторой страны используются n типов купюр, номиналы которых равны a_1, a_2, \dots, a_n . При этом все номиналы различны и перечислены в порядке возрастания: если $i \geq 2$, то $a_{i-1} < a_i$, а также $a_1 = 1$.

Банкомат использует следующий жадный алгоритм для выдачи купюр. Пусть клиент запросил у банкомата сумму s . Изначально есть пустой набор выдаваемых купюр. На каждом шаге алгоритм добавляет в набор купюру максимально возможного номинала так, чтобы сумма номиналов купюр в наборе не превышала s . Когда сумма номиналов купюр в наборе стала равна s , алгоритм останавливается. Отметим, что, поскольку существует купюра с номиналом $a_1 = 1$, алгоритм всегда заканчивает работу за конечное число шагов.

Чтобы оценить эффективность данного алгоритма, требуется выяснить, какое максимальное число купюр может потребоваться выдать за один раз, если максимальная сумма, которую можно запросить, равна b . Поскольку максимальная сумма может зависеть от категории обслуживания клиента, необходимо ответить на q запросов для сумм b_1, b_2, \dots, b_q .

Требуется написать программу, которая по списку номиналов купюр денежной системы и максимальной сумме, которую можно запросить, определяет сумму, которую необходимо запросить в банкомате, чтобы получить максимальное число купюр, и искомое число купюр.

Формат входных данных

Первая строка содержит целое число n — количество номиналов купюр ($1 \leq n \leq 200\,000$).

Вторая строка содержит n различных целых чисел a_i ($1 = a_1 < a_2 < \dots < a_n \leq 10^{18}$).

Третья строка содержит целое число q — количество запросов ($1 \leq q \leq 200\,000$).

Следующие q строк содержат по одному целому числу b_i ($1 \leq b_i \leq 10^{18}$).

Формат выходных данных

Для каждого запроса выведите два числа — сумму, которую необходимо запросить в банкомате, чтобы получить максимальное число купюр, и искомое число купюр.

Если существует несколько сумм, не превышающих максимального значения, для которых будет выдано максимальное число купюр, требуется вывести любую из них.

Система оценивания

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Доп. ограничения	Необходимые подзадачи	Информация о проверке
1	13	$n \leq 500, q \leq 5, a_i \leq 500, b_i \leq 500$		первая ошибка
2	18	$n = 60, q \leq 5, a_i = 2^{i-1}$		первая ошибка
3	20	$q \leq 5, b_i \leq 2 \cdot 10^5$	1	первая ошибка
4	21	$q \leq 5$	1, 2, 3	первая ошибка
5	28		1 – 4	первая ошибка

Пример

стандартный ввод	стандартный вывод
4	2 2
1 5 10 50	8 4
3	49 9
2	
8	
50	

Пояснение к примеру

В примере:

- 2 будет выдано как $1 + 1$,
- 8 будет выдано как $5 + 1 + 1 + 1$,
- 49 будет выдано как $10 + 10 + 10 + 10 + 5 + 1 + 1 + 1 + 1$.

Задача 8. Плакаты

Друзья готовятся встретить национальную команду, возвращающуюся с международной олимпиады по информатике. Для этого они подготовили множество красочных плакатов. Осталось только продумать детали поздравления.

Для того, чтобы приветствовать команду, n друзей встанут в круг. Пронумеруем их от 1 до n в порядке их расположения по кругу. Таким образом, для всех i от 1 до $n - 1$ друзья с номерами i и $i + 1$ стоят рядом, также рядом стоят друзья с номерами n и 1. У каждого из друзей есть плакат. Каждый плакат характеризуется своей *красочностью* — целым неотрицательным числом. Плакат у друга с номером i имеет красочность a_i .

Когда поздравление начнётся, некоторые из друзей поднимут свои плакаты и будут показывать их команде. Для того, чтобы члены команды не растерялись и смогли рассмотреть все плакаты, не должно быть четырёх или более стоящих подряд друзей с поднятым плакатом.

Друзья планируют изменять плакаты в процессе встречи. Всего будет внесено q изменений в плакаты. После i -го из них плакат друга с номером p_i будет иметь красочность v_i . После каждого из изменений друзья хотят определить, какую максимальную суммарную красочность могут иметь поднятые плакаты, если нельзя нарушать установленное ограничение.

Требуется написать программу, которая по заданной начальной красочности плакатов и последовательности их изменений определяет в начале, а также после каждого изменения, какой максимальной суммарной красочности поднятых плакатов можно добиться, не нарушая условие, что поднято не более трёх плакатов подряд.

Формат входных данных

Первая строка ввода содержит целое число n ($4 \leq n \leq 40\,000$) — количество друзей.

Вторая строка содержит n целых чисел a_i ($0 \leq a_i \leq 10^9$) — исходные значения красочности плакатов у друзей.

Третья строка содержит одно целое число q ($0 \leq q \leq 40\,000$) — количество изменений плакатов, которые вносили друзья.

Каждая из следующих q строк содержит два целых числа p_i и v_i ($1 \leq p_i \leq n$; $0 \leq v_i \leq 10^9$) — номер друга, плакат которого изменился, и новая красочность этого плаката.

Формат выходных данных

Выведите $q + 1$ число. Перед первым изменением, а также после каждого изменения плакатов выведите одно целое число — максимальное суммарное значение красочности поднятых плакатов, если нельзя поднимать больше трёх плакатов подряд.

Система оценивания

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи	Информация о проверке
1	11	$n \leq 10, q = 0$		первая ошибка
2	12	$n \leq 10, q \leq 10$	1	первая ошибка
3	13	$n \leq 1\,000, q \leq 1\,000$	1, 2	первая ошибка
4	17	$n \leq 40\,000, q = 0$	1	первая ошибка
5	47	$n \leq 40\,000, q \leq 40\,000$	1, 2, 3, 4	первая ошибка

Пример

стандартный ввод	стандартный вывод
6	17
1 2 3 4 5 6	13
2	15
6 0	
2 5	

Пояснение к примеру

Рассмотрим тест из примера.

Перед первым изменением плакаты следует поднять друзьям с номерами 2, 4, 5, 6. Суммарная красочность поднятых плакатов будет равняться 17.

После первого изменения плакат друга с номером 6 имеет красочность 0. Теперь плакаты следует поднять друзьям с номерами 1, 3, 4, 5. Суммарная красочность будет равняться 13.

После второго изменения плакат друга с номером 2 имеет красочность 5. Плакаты следует поднять друзьям с номерами 1, 2, 4, 5. Суммарная красочность будет равняться 15.