

Задача А. Отделы

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

В городе П. существует одна огромная it-компания КТР. В компании КТР есть n отделов; в i -м отделе изначально работает a_i сотрудников. Как и любая другая компания, она прежде всего гонится за лучшими показателями.

На этот раз менеджмент компании подсчитал *коэффициент полезности* каждого сотрудника в компании. Кроме того, менеджмент КТР считает, что чем больше среднее значение *коэффициента полезности* в отделе, тем лучше. Поэтому компания решила провести реорганизацию отделов: менеджмент хочет так перераспределить сотрудников по отделам (число отделов должно остаться прежним), чтобы сумма средних *коэффициентов полезности* по всем отделам была максимальной.

Кроме того, менеджмент считает, что в каждом отделе должно работать хотя бы **два** сотрудника, поэтому одного человека в отделе оставить нельзя.

Генеральный директор компании КТР Руслан Валерьевич лично пришел к Вам, чтобы поручить задачу о перераспределении сотрудников. Вам необходимо оценить, какой максимальной суммы средних *коэффициентов полезности* можно добиться.

Формат входных данных

В первой строке задано одно целое число n ($1 \leq n \leq 10^5$) — количество отделов.

Далее описываются n отделов, i -й из которых описывается двумя строками.

В первой задано целое число a_i ($2 \leq a_i \leq 2 \cdot 10^5$) — количество сотрудников в i -м отделе до перераспределения.

Во второй заданы a_i целых чисел $b_{i_1}, b_{i_2}, \dots, b_{i_{a_i}}$ ($1 \leq b_{i_j} \leq 10^9$) — *коэффициенты полезности* сотрудников в i -м отделе.

Гарантируется, что сумма по всем a_i не превышает $2 \cdot 10^5$.

Формат выходных данных

Выведите одно число — максимальную сумму средних *коэффициентов полезности* по всем отделам, которой можно добиться после перераспределения.

Ваш ответ будет считаться правильным, если его абсолютная или относительная ошибка не превосходит 10^{-6} .

Система оценки

Подгруппа	Доп. ограничения	Баллы	Необх. подгруппы
1	—	100	—

Обратите внимание, что для прохождения подгрупп ваша программа **не** обязана выдавать верный ответ на примерах из условия.

Примеры

стандартный ввод	стандартный вывод
3 2 20 40 2 10 30 3 60 60 30	115.0000000000
2 3 2 3 3 2 1 2	4.6666666667

Замечание

В первом примере можно перераспределить сотрудников так, чтобы:

- В первом отделе работали сотрудники с показателями 60 и 30: среднее коэффициентов равно 45;
- Во втором отделе работали сотрудники с показателями 10, 20 и 30: среднее коэффициентов равно 20;
- В третьем отделе работали сотрудники с показателями 60 и 40: среднее коэффициентов равно 50.

Тогда сумма средних равна $45 + 20 + 50 = 115$. Можно показать, что большую сумму средних набрать нельзя.

Задача В. Возвращение к мечте

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

В какой-то момент Игорю надоело, что вечно он плохо выступает на соревнованиях по спортивному программированию. В связи с этим он принял импульсивное решение бросить это дело и уйти с головой в работу.

Для этого как раз подвернулось предложение о стажировке в IT-компании «Ты», где Игорь отработал честные три месяца. Под конец стажировки он порефлексировал над предыдущими ошибками в спортивном программировании и решил вернуться. Разумеется, после промежутка в несколько месяцев свое возвращение стоило начать с задач попроще. Например, с поиска делителей в числе.

Для этого есть всем известный алгоритм:

```
long long cnt = 0;
for (long long i = 1; i * i <= n; i++) {
    if (n % i) continue;
    cnt++;
    if (i * i != n) cnt++;
}
return cnt;
```

Однако при написании кода Игорь очень спешил и вместо символа ‘%’ случайно написал символ ‘&’ (можете проверить — они находятся близко к друг другу на клавиатуре!):

```
long long cnt = 0;
for (long long i = 1; i * i <= n; i++) {
    if (n & i) continue;
    cnt++;
    if (i * i != n) cnt++;
}
return cnt;
```

Игорь долго не мог понять, что с его программой не так, но вскоре смог найти ошибку и исправить ее. Однако Игорю стало очень интересно, а что за алгоритм он написал в самом начале. Он хочет теперь узнать, какое значение вернет этот алгоритм при каком-то заданном n .

Формат входных данных

В первой строке задано одно целое число t ($1 \leq t \leq 10^3$) — количество наборов входных данных.

В единственной строке каждого набора входных данных дано одно целое число n ($1 \leq n \leq 2^{62}$) — число, по значению которого нужно посчитать значение, возвращаемое алгоритмом.

Формат выходных данных

Для каждого набора входных данных выведите ответ на задачу.

Система оценки

Подгруппа	Доп. ограничения		Баллы	Необх. подгруппы
	n	t		
1	$n = 2^k$ для некоторого целого k	—	15	—
2	$n \leq 10^6$	—	15	—
3	$n \leq 10^{12}$	$t = 1$	15	—
4	—	—	55	1, 2, 3

Обратите внимание, что для прохождения подгрупп ваша программа **не** обязана выдавать верный ответ на примерах из условия.

Пример

стандартный ввод	стандартный вывод
5	0
1	2
5	2
10	6
20	4
33	

Задача С. Симус, Петрозаводск и числа на доске

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 512 мегабайт

Симус Финниган решил, что, находясь на последнем курсе Школы Магии и Волшебства Хогвартс, доучиваться ему не хочется. Поэтому он решил отправиться жить в Петрозаводск!

Прежде, чем уехать в дальние-дальние страны, Симус Финниган оставил послание оставшимся в Лондоне. Послание представляло собой написанную на доске последовательность целых положительных чисел a длины n . Никто не знает, к чему эта последовательность была оставлена. Возможно, в ней вообще не было никакого смысла — кто знает этого Симуса!

Тем не менее, ни для кого не секрет, что волшебники любят придавать значение всему, что прибито и что не прибито. В нашем конкретном случае Невилл захотел обнаружить какие-нибудь интересные свойства у написанной на доске последовательности. Для начала он хочет найти такую пару индексов (i, j) , что $\gcd^{[1]}(a_i, a_j) = \min(a_i, a_j)$ и при этом значение $|a_j - a_i|$ максимально возможное.

Однако, поскольку Невилл — истинный волшебник, проводить расчеты самостоятельно он ну очень не желает. Поэтому Невилл обратился к Вам за помощью.

Формат входных данных

В первой строке задано одно целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных.

Первая строка каждого набора входных данных содержит целое число n ($1 \leq n \leq 4 \cdot 10^5$) — длина последовательности.

Вторая строка каждого набора входных данных содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^7$) — сама последовательность.

Гарантируется, что сумма n по всем наборам входных данных не превосходит $4 \cdot 10^5$

Формат выходных данных

Для каждого набора входных данных выведите искомые два числа: i, j ($1 \leq i, j \leq n$) в любом порядке.

Если ответов несколько, выведите любой из них.

Система оценки

За $\sum n$ обозначим сумму n по всем наборам входных данных.

Группа	Доп. ограничения			Баллы	Необх. подгруппы
	t	$\sum n$	a_i		
1	$t = 1$	—	—	40	—
2	$t \leq 100$	$\sum n \leq 10^3$	$a_i \leq 10^3$	10	—
3	—	$\sum n \leq 10^5$	$a_i \leq 10^5$	40	1
4	—	—	—	10	1, 2, 3

Обратите внимание, что для прохождения подгрупп ваша программа **не** обязана выдавать верный ответ на примерах из условия.

Пример

стандартный ввод	стандартный вывод
4	1 4
4	1 1
1 2 3 4	5 6
5	1 1
100 3 145 44 7	
10	
333 333 333 333 333 666 666 666 666 666	
1	
1	

Замечание

[1] $\gcd(x, y)$ — наибольший общий делитель чисел x и y .

Рассмотрим примеры из условия.

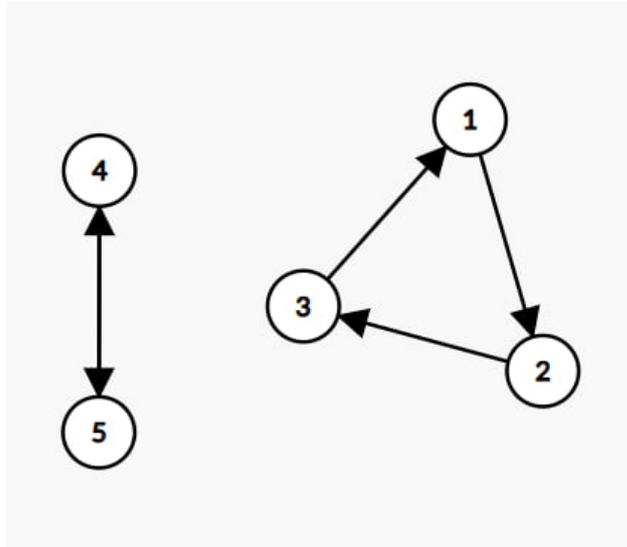
- В первом наборе входных данных верным ответом является только пара $(1, 4)$ (или или та же самая пара, но перевернутая - $(4, 1)$), так как $\gcd(1, 4) = 1$ и $|4 - 1| = 3$ - максимальное значение среди всех возможных разностей элементов массива.
- Во втором наборе входных данных нет такой пары, состоящей из различных индексов, которая подходит под условие. Поэтому ответ может состоять только из одинаковых индексов, а максимальная разница равна нулю.
- В третьем наборе входных данных максимальная разница, которой мы можем достигнуть — это 333. Сделать это можно выбрав сначала одно из пяти первых чисел, а затем одно из пяти последних.

Задача D. Две перестановки

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

Пусть у Вас есть перестановка целых чисел p .

Для перестановки p определим $G(p)$ как ориентированный граф, состоящий из n вершин, причем из каждой вершины v есть ровно одно ребро (v, p_v) .



Граф, получаемый из перестановки $[2, 3, 1, 5, 4]$.

У Игоря есть две перестановки p и q одинаковой длины n . Игорь называет пару чисел (i, j) *красивой*, если:

- $1 \leq i < j \leq n$;
- В графе $G(p)$ вершина с номером i достижима из вершины с номером j ;
- В графе $G(p)$ вершина с номером j достижима из вершины с номером i ;
- В графе $G(q)$ вершина с номером i достижима из вершины с номером j ;
- В графе $G(q)$ вершина с номером j достижима из вершины с номером i .

Теперь Игорь просит Вас помочь ему посчитать количество *красивых* пар. Поможете ему?

Формат входных данных

В первой строке задано одно целое число n ($1 \leq n \leq 2 \cdot 10^5$) — длина перестановок.

Во второй строке заданы n целых чисел p_1, p_2, \dots, p_n ($1 \leq p_i \leq n$) — первая перестановка. Гарантируется, что все p_i различны.

В третьей строке заданы n целых чисел q_1, q_2, \dots, q_n ($1 \leq q_i \leq n$) — вторая перестановка. Гарантируется, что все q_i различны.

Система оценки

Подгруппа	Доп. ограничения	Баллы	Необх. подгруппы
	n		
1	$n \leq 100$	25	—
2	$n \leq 2000$	25	1
3	—	50	1, 2

Обратите внимание, что для прохождения подгрупп ваша программа **не** обязана выдавать верный ответ на примерах из условия.

Формат выходных данных

Выведите одно целое число — количество *красивых* пар.

Примеры

стандартный ввод	стандартный вывод
5 3 1 2 5 4 2 3 1 5 4	4
5 1 2 3 4 5 5 4 3 2 1	0
4 2 1 4 3 3 1 2 4	1
1 1 1	0

Задача Е. Новый год, подарки, розы

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

На носу Новый Год! Разумеется, в честь этого события Игорь очень хочет сделать приятный подарок Диане. Он прекрасно знает любовь Дианы к розам, поэтому было решено купить розы! Много роз!

Придя в цветочный магазин, Игорь увидел n ваз, стоящих вдоль стены; причем в i -й вазе находится ровно одна роза цвета a_i . Игорь, ввиду своих убеждений, решил, что нужно купить розы так, чтобы вазы купленных роз образовывали подотрезок. То есть, если были куплены розы, находящиеся в l -й и r -й вазах ($l \leq r$), то розы, находящиеся в вазах с номерами $l + 1, l + 2, \dots, r - 1$ тоже должны быть куплены.

Кроме того, подотрезок купленных роз обязательно должен быть *прекрасным*! У Игоря есть любимое число k . Он считает, что подотрезок купленных роз будет *прекрасным*, если в нем есть **ровно** k роз одного цвета (таких цветов может быть несколько; важно, чтобы был хотя бы один такой цвет).

Например, пусть $a = [1, 2, 3, 2, 3, 3]$, $k = 2$. Тогда подотрезок $(2, 5) = [2, 3, 2, 3]$ будет *прекрасным*, потому что в нем сразу два цвета 2 и 3 встречаются $k = 2$ раз. С другой стороны, $(3, 6) = [3, 2, 3, 3]$ **не будет прекрасным**, потому что число 2 встречается один раз, число 3 встречается три раза, а остальные числа не встречаются вообще.

Игорю стало интересно, сколько подотрезков роз он мог бы купить.

Отметим, что во вселенной, в которой живет Игорь, никто не обращает на четность количества подаренных цветов, поэтому и в нашей задаче мы не будем обращать внимание на четность длины подотрезка.

Формат входных данных

В первой строке заданы два целых числа n и k ($1 \leq k \leq n \leq 2 \cdot 10^5$) — количество ваз с розами и любимое число Игоря.

Во второй строке заданы n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$) — цвета роз в вазах.

Система оценки

Подгруппа	Доп. ограничения	Баллы	Необх. подгруппы
	n		
1	$n \leq 100$	10	—
2	$n \leq 5000$	30	1
3	—	60	1, 2

Обратите внимание, что для прохождения подгрупп ваша программа **не** обязана выдавать верный ответ на примерах из условия.

Формат выходных данных

Выведите одно целое число — количество *прекрасных* подотрезков роз.

Примеры

стандартный ввод	стандартный вывод
6 2 1 2 3 2 3 3	9
6 3 1 2 3 2 3 3	3
6 4 1 2 3 2 3 3	0
6 1 1 2 3 2 3 3	18

Замечание

Приведем список *прекрасных* подотрезков из первого примера:

- $(1, 4) = [1, 2, 3, 2]$
- $(1, 5) = [1, 2, 3, 2, 3]$
- $(1, 6) = [1, 2, 3, 2, 3, 3]$
- $(2, 4) = [2, 3, 2]$
- $(2, 5) = [2, 3, 2, 3]$
- $(2, 6) = [2, 3, 2, 3, 3]$
- $(3, 5) = [3, 2, 3]$
- $(4, 6) = [2, 3, 3]$
- $(5, 6) = [3, 3]$