

Задача А. Покупка велосипеда

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Поликарп собирается в магазин, чтобы купить велосипед, о котором он так давно мечтал. Велосипед в магазине стоит n бурлей.

Поликарп живет в Берляндии, где нет денежных купюр, есть только два типа монет: монеты с номиналом 2 бурля и монеты с номиналом 5 бурлей. Поликарп накопил достаточно монет каждого типа для покупки велосипеда.

Все монеты весят одинаково, но при этом они достаточно тяжелые, причем монета с номиналом 2 бурля весит столько же, сколько весит монета с номиналом 5 бурлей. Поэтому Поликарп хочет взять с собой минимально возможное количество монет, которых хватит для покупки велосипеда. Также Поликарп хочет расплатиться без сдачи, то есть взять с собой монеты с суммарным номиналом ровно n бурлей.

Перед вами стоит задача определить минимально возможное количество монет, суммарный номинал которых ровно n бурлей.

Формат входных данных

В первой строке следует целое число n ($4 \leq n \leq 1\,000\,000$) — стоимость велосипеда в бурлях. Гарантируется, что ограничения на n таковы, что всегда возможно набрать суммарный номинал ровно n бурлей, используя монеты с номиналом 2 бурля и с номиналом 5 бурлей.

Формат выходных данных

Выведите минимально возможное количество монет, суммарный номинал которых ровно n бурлей.

Примеры

стандартный ввод	стандартный вывод
13	5
20	4
6	3

Замечание

В первом примере Поликарпу нужно взять одну монету с номиналом 5 бурлей и четыре монеты с номиналом 2 бурля. Таким образом, он возьмет $1 + 4 = 5$ монет с суммарным номиналом $1 \cdot 5 + 4 \cdot 2 = 5 + 8 = 13$ бурлей.

Во втором примере Поликарпу нужно взять четыре монеты с номиналом 5 бурлей. Их суммарный номинал равен $4 \cdot 5 = 20$ бурлей.

В третьем примере Поликарпу нужно взять три монеты с номиналом 2 бурля. Их суммарный номинал равен $3 \cdot 2 = 6$ бурлей.

Задача В. Голы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Футбольная команда Поликарпа завершила сезон, выиграв a матчей и проиграв b матчей, причем ни один из матчей не завершился вничью. Также известно, что за $a + b$ матчей команда Поликарпа суммарно забила x голов и пропустила y голов.

В футбольном матче победителем считается команда, которая забила больше голов, чем пропустила.

Поликарп не помнит счет каждого из матчей прошедшего сезона. Ему стало интересно, каково максимальное количество голов, которые могла забить его команда в каком-то одном из матчей, учитывая, что за весь сезон команда выиграла a матчей и проиграла b матчей, суммарно забив x голов и пропустив y голов.

Перед вами стоит задача помочь Поликарпу и найти максимальное количество голов, которые могла забить его команда в каком-то одном из матчей.

Формат входных данных

В первой строке записано одно целое число t ($1 \leq t \leq 1\,000$) — количество наборов входных данных.

В единственной строке каждого набора входных данных следуют четыре целых числа a, b, x и y ($0 \leq a, b, x, y \leq 10^9$, $a + b > 0$) — количество побед, количество поражений, суммарное количество забитых голов и суммарное количество пропущенных голов. Гарантируется, что за сезон команда Поликарпа сыграла хотя бы один матч.

Формат выходных данных

Для каждого набора входных данных выведите одно целое число — максимальное количество голов, которые команда Поликарпа могла забить в каком-то из матчей. Если набор входных данных противоречив, и нельзя выиграть a матчей и проиграть b матчей, при этом забив в них x голов и пропустив y голов, то для такого набора входных данных нужно вывести -1 .

Пример

стандартный ввод	стандартный вывод
5	4
2 1 5 4	2
0 3 2 6	-1
2 0 1 0	-1
1 3 4 2	0
0 1 0 1	

Замечание

В первом наборе входных данных команда Поликарпа могла выиграть первый матч со счетом $4 : 0$, выиграть второй матч со счетом $1 : 0$ и проиграть третий матч со счетом $0 : 4$, поэтому максимальное количество голов, забитых в одном матче, равно 4.

Во втором наборе входных данных команда Поликарпа могла проиграть первый матч со счетом $2 : 3$, проиграть второй матч со счетом $0 : 1$ и проиграть третий матч со счетом $0 : 2$, поэтому максимальное количество голов, забитых в одном матче, равно 2.

В третьем наборе входных данных ответ -1 , так как нельзя выиграть два матча, суммарно забив в них только один гол.

В четвертом наборе входных данных ответ -1 , так как нельзя проиграть три матча, суммарно пропустив в них всего два гола.

В пятом наборе входных данных команда Поликарпа проиграла единственный матч со счетом $0 : 1$, поэтому максимальное количество голов, забитых в одном матче, равно 0.

Задача С. Дерево обхода в глубину

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

Задан связный неориентированный граф G , состоящий из n вершин и m ребер. Вершины пронумерованы от 1 до n . Также задано дерево T , состоящее из тех же n вершин. Все ребра дерева T являются ребрами графа G .

Назовем списком смежности вершины v список таких вершин u , что существует ребро (v, u) . В каждом списке смежности вершины расположены в каком-то порядке.

Напомним классический алгоритм обхода в глубину графа G . Определим функцию **dfs**, которая принимает один целочисленный аргумент v — номер рассматриваемой вершины. На входе в функцию вершина v помечается посещенной. Затем перебираются все вершины u списка смежности v . Если u ранее была посещена, то алгоритм переходит к рассмотрению следующей вершины в списке смежности. Если же u не была посещена, то функция **dfs** запускается из вершины u рекурсивно. Когда рекурсивный вызов завершается, алгоритм снова переходит к рассмотрению следующей вершины в списке смежности. Когда следующей вершины в списке смежности нет, функция завершается.

Дерево обхода в глубину строится следующим образом. Когда функция **dfs** находится в вершине v и запускается рекурсивно из вершины u , ребро (v, u) добавляется в дерево. Можно показать, что граф, построенный таким алгоритмом, действительно является деревом.

Для каждой вершины i от 1 до n определите, можно ли так переупорядочить списки смежности всех вершин (возможно, не изменяя порядок), чтобы дерево обхода в глубину графа G , запущенного из вершины i , состояло из тех же ребер, что и дерево T . Обратите внимание: вы можете менять порядок вершин внутри каждого списка смежности, но не можете перемещать их между разными списками смежности.

Формат входных данных

В первой строке записаны два целых числа n и m ($2 \leq n \leq 2 \cdot 10^5$; $n - 1 \leq m \leq \min(\frac{n(n-1)}{2}, 3 \cdot 10^5)$).

В каждой из следующих m строк записаны два целых числа v и u ($1 \leq v, u \leq n$) — описание ребер графа G .

В каждой из следующих $n - 1$ строк записаны два целых числа v и u ($1 \leq v, u \leq n$) — описание ребер дерева T .

Дополнительные ограничения на входные данные:

- граф G связный, а также не содержит петель и кратных ребер;
- ребра графа T образуют дерево;
- все ребра дерева T являются ребрами графа G .

Формат выходных данных

Выведите бинарную строку длины n . i -й символ должен быть равен 1, если можно переупорядочить списки смежности всех вершин (возможно, не изменяя порядок), чтобы дерево обхода в глубину графа G , запущенного из вершины i , состояло из тех же ребер, что и дерево T . В противном случае i -й символ должен быть равен 0.

Примеры

стандартный ввод	стандартный вывод
4 4 1 2 2 3 2 4 1 3 1 2 2 3 4 2	1010
6 7 1 2 1 3 1 5 3 5 3 4 3 6 5 6 1 2 1 3 3 5 3 4 5 6	110001
4 6 1 2 3 1 3 2 2 4 3 4 1 4 2 1 4 1 1 3	0000

Задача D. Преобразование строки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

У Поликарпа есть строка s длины n , состоящая из строчных букв латинского алфавита.

Поликарп может производить со своей строкой следующую операцию произвольное количество раз:

- удалить первый элемент из строки s и вставить его в произвольную позицию строки s , то есть вставить его между любыми двумя соседними буквами, либо вставить в конец строки.

Перед вами стоит задача определить минимальное количество операций, которые необходимы, чтобы преобразовать строку s в заданную строку t , либо сообщить, что это невозможно.

Формат входных данных

В первой строке следует целое число n ($1 \leq n \leq 200\,000$) — длина строк s и t .

Во второй строке следует строка s длины n , состоящая из строчных букв латинского алфавита.

В третьей строке следует строка t длины n , состоящая из строчных букв латинского алфавита.

Формат выходных данных

Если невозможно преобразовать строку s в строку t с помощью описанных операций, выведите -1 . В противном случае выведите минимальное количество операций, которые необходимы, чтобы преобразовать строку s в строку t .

Примеры

стандартный ввод	стандартный вывод
4 abcd dcba	3
3 tot tot	0
7 cabaaba abacaba	1
2 ab bb	-1

Задача Е. Рассадка на экзамене

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Поликарп сегодня принимает экзамен у студентов. Экзамен пройдет в аудитории, в которой есть один длинный стол, за которым есть n мест для студентов.

Когда Поликарп зашел в аудиторию, все пришедшие на экзамен студенты уже сели на какие-то места.

Поликарп хочет предотвратить списывание на экзамене, поэтому пересадит некоторых студентов за свободные места так, чтобы выполнялись следующие условия:

- на самом левом месте за столом должен сидеть студент;
- на самом правом месте за столом должен сидеть студент;
- расстояние между **всеми парами соседних студентов** должно быть **одинаковым**.

Перед вами стоит задача помочь Поликарпу и определить **минимальное** количество студентов, которых нужно пересадить так, чтобы выполнялись описанные условия. Считайте, что после пересадки студента место, на котором он сидел, становится свободным.

Формат входных данных

В первой строке следует целое число n ($2 \leq n \leq 200\,000$) — количество мест за столом.

Во второй строке следует строка s , состоящая из n символов «F» и «S». Если i -й символ строки равен «F», то i -е слева место за столом свободно. Если i -й символ строки равен «S», то i -е слева место за столом занято ровно одним студентом. Гарантируется, что на экзамен пришли хотя бы два студента.

Формат выходных данных

Выведите минимальное количество студентов, которых нужно пересадить на свободные места, чтобы экзамен прошел без списывания. Если это невозможно, выведите -1 .

Примеры

стандартный ввод	стандартный вывод
7 FSSFFFS	2
11 SFSFSFSFSFS	0
4 SSFS	-1
17 FFSSSFFFFFFFFSSFFF	3
6 SSSSSS	0

Замечание

В первом примере можно, например, пересадить одного студента с места 2 на место 1 и пересадить другого студента с места 3 на место 4. Тогда рассадка студентов станет равна «SFFSFFS». Эта рассадка удовлетворяет всем условиям предотвращения списывания, поэтому ответ на этот пример равен 2.

Во втором примере никого пересаживать не нужно, так как изначальная рассадка удовлетворяет всем условиям предотвращения списывания, поэтому ответ на этот пример равен 0.

В третьем примере невозможно пересадить студентов так, чтобы рассадка удовлетворяла всем условиям предотвращения списывания, поэтому ответ на этот пример равен -1 .

В четвертом примере можно, например, пересадить одного студента с места 3 на место 17, пересадить другого студента с места 4 на место 1 и пересадить еще одного студента с места 14 на место 9. Тогда рассадка студентов станет равна «SFFFSFFFSFFFSFFFS». Эта рассадка удовлетворяет всем условиям предотвращения списывания, поэтому ответ на этот пример равен 3.

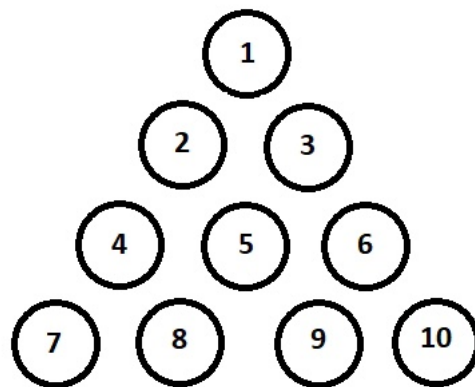
В пятом примере никого пересаживать не нужно, так как изначальная рассадка удовлетворяет всем условиям предотвращения списывания, поэтому ответ на этот пример равен 0.

Задача F. Бильярд

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Поликарп увлекся игрой в бильярд.

Изначально все шары в бильярде выстроены в форме пирамиды, которая состоит из n рядов. В первом ряду находится один шар, во втором ряду находятся два шара, в третьем ряду находятся три шара и так далее. Шары нумеруются, начиная с единицы, по рядам сверху вниз, а в каждом ряду слева направо. На рисунке изображен пример пирамиды для $n = 4$.



Поликарп сейчас учится делать первый удар в партии — подбоя пирамиды. Для этого используется специальный шар, который после удара попадает в какой-то из шаров в пирамиде. Поликарп делает подбой так, что специальный шар попадает ровно в один шар в пирамиде, причем этот шар может быть в любом месте пирамиды, это будет не обязательно крайний шар.

Пусть Поликарп попал специальным шаром в шар номер x в пирамиде. После этого шар x придет в движение и ударится в два шара, которые расположены под ним в следующем ряду (если шар номер x расположен не в последнем ряду пирамиды). После этого те два шара, в которые попал шар x , придут в движение и каждый ударится в два шара, которые расположены под ним в следующем ряду. Этот процесс продолжится до тех пор, пока не придут в движение шары в последнем ряду. Все остальные шары, которые не пришли в движение после описанного процесса, останутся на месте после удара.

Например, если $n = 4$ и Поликарп попал специальным шаром в шар номер 3, то он придет в движение, затем шары 5 и 6 придут в движение, затем шары 8, 9 и 10 придут в движение. После этого больше никакие шары не придут в движение, так как шары 8, 9 и 10 находятся в последнем ряду. Таким образом, суммарно 6 шаров придут в движение.

Перед вами стоит задача определить, сколько шаров в пирамиде придут в движение, если Поликарп в ходе подбоя пирамиды попал специальным шаром в шар номер x .

Формат входных данных

В первой строке записано одно целое число t ($1 \leq t \leq 1\,000$) — количество наборов входных данных.

В единственной строке каждого набора входных данных следуют два целых числа n и x ($1 \leq n \leq 10^9$, $1 \leq x \leq 10^{18}$) — количество рядов в пирамиде и номер шара в пирамиде, в который попадет специальный шар после подбоя пирамиды.

Для каждого набора входных данных гарантируется, что заданный номер x не превышает количества шаров в пирамиде, состоящей из n рядов.

Формат выходных данных

Для каждого набора входных данных выведите одно целое число — количество шаров в пирамиде, которые придут в движение, если в ходе подбоя пирамиды специальный шар попадет в шар номер x в пирамиде.

Пример

стандартный ввод	стандартный вывод
3	6
4 3	1
8 29	500000000500000000
1000000000 1	

Замечание

Первый набор входных данных разобран в условии.

Во втором наборе входных данных специальный шар попадет в самый левый шар в последнем (восьмом) ряду, поэтому только он один придет в движение.

В третьем наборе входных данных специальный шар попадет в шар номер 1 в пирамиде, поэтому все шары в пирамиде придут в движение.

Задача G. Задача про сортировку

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

Задана перестановка p размера n (напомним, что перестановка размера n — это такой массив размера n , в котором каждое целое число от 1 до n встречается ровно один раз). Вы должны отсортировать ее в возрастающем порядке.

Для этого можно выполнять следующие операции:

- выбрать подотрезок перестановки, в котором есть **хотя бы два элемента**, и развернуть его. Иными словами, вы выбираете два целых числа l и r ($1 \leq l < r \leq n$), а затем переворачиваете подмассив $[p_l, p_{l+1}, \dots, p_r]$.

Вы должны найти последовательность операций, удовлетворяющую следующим условиям:

- количество операций — четное число от 0 до $4n$;
- перестановка является отсортированной после применения всей последовательности операций (если она является отсортированной после какой-то операции, но последовательность операций не завершена — это не считается);
- количество операций, в которых разворачивается подотрезок с четным числом элементов, должно быть равно количеству операций, в которых разворачивается подотрезок с нечетным числом элементов.

Найдите подходящую последовательность операций или сообщите, что ее не существует. Обратите внимание, что минимизировать количество операций не обязательно.

Формат входных данных

В первой строке задано одно целое число t ($1 \leq t \leq 5000$) — количество наборов выходных данных.

Далее следуют t наборов входных данных. Каждый набор задается двумя строками:

- в первой из них задано одно целое число n ($3 \leq n \leq 100$);
- во второй задано n различных целых чисел $p_1, p_2, \dots, p_n$ ($1 \leq p_i \leq n$).

Дополнительное ограничение на входные данные: сумма n по всем наборам входных данных не превышает $2 \cdot 10^4$.

Формат выходных данных

На каждый набор входных данных выведите ответ следующим образом:

- если последовательности операций, удовлетворяющей всем условиям, не существует, выведите -1 ;
- иначе в первой строке выведите одно целое число q ($0 \leq q \leq 4n$) — количество операций. Затем выведите q строк, в i -й из которых выведите два целых числа l_i и r_i ($1 \leq l_i < r_i \leq n$) — границы подотрезка, который вы разворачиваете в ходе i -й операции.

Если подходящих последовательностей операций несколько, вы можете вывести любую из них.

Пример

стандартный ввод	стандартный вывод
5	0
3	-1
1 2 3	2
3	1 3
2 1 3	1 2
3	4
3 1 2	1 2
4	2 4
4 3 2 1	1 3
6	1 2
6 1 2 3 4 5	2
	2 6
	1 6

Задача Н. Вычитание

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

У Поликарпа есть целое положительное число a , не содержащее лидирующих нулей.

Перед вами стоит задача найти **минимальное** число b , которое удовлетворяет следующим условиям:

- b — целое положительное число;
- $b < a$;
- число $c = a - b$ является целым положительным числом, состоящим из **одинаковых цифр**.

Формат входных данных

В первой строке следует целое число a ($10 \leq a \leq 10^{1\,000\,000}$) — число Поликарпа. Гарантируется, что заданное число не содержит лидирующих нулей.

Формат выходных данных

Выведите минимальное число b без лидирующих нулей, удовлетворяющее всем описанным условиям.

Примеры

стандартный ввод	стандартный вывод
239	17
11	2
55	11
10055	56
123456789012345678901234567890	12345677901234567790123456779

Замечание

В первом примере минимальное $b = 17$, так как $239 - 17 = 222$ является целым положительным числом, состоящим из одинаковых цифр.

Во втором примере минимальное $b = 2$, так как $11 - 2 = 9$ является целым положительным числом, состоящим из одинаковых цифр.

В третьем примере минимальное $b = 11$, так как $55 - 11 = 44$ является целым положительным числом, состоящим из одинаковых цифр.

В четвертом примере минимальное $b = 56$, так как $10\,055 - 56 = 9\,999$ является целым положительным числом, состоящим из одинаковых цифр.

Задача I. Этажи, этажи...

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Поликарп должен доставить посылку по указанному адресу. Он уже подошел к нужному подъезду дома. Посылка должна быть доставлена в первый подъезд на этаж номер f в квартиру номер r . Считайте, что в нужном доме гарантированно не менее f этажей.

Поликарп знает, что квартиры в доме нумеруются с единицы, начиная с первого этажа, а на каждом этаже расположено одинаковое количество квартир.

Например, если количество квартир на одном этаже равно 4, то на первом этаже расположены квартиры с номерами 1, 2, 3 и 4, на втором этаже расположены квартиры с номерами 5, 6, 7 и 8, на третьем этаже расположены квартиры с номерами 9, 10, 11 и 12, и так далее.

Поликарпу стало интересно, какое минимально возможное и максимально возможное количество квартир может быть на одном этаже, если ему известно, что квартира номер r расположена на этаже номер f . Перед вами стоит задача помочь Поликарпу определить интересные ему значения.

Формат входных данных

В первой строке записано одно целое число t ($1 \leq t \leq 1000$) — количество наборов входных данных.

В единственной строке каждого набора входных данных следуют два целых числа f и r ($1 \leq f \leq r \leq 10^9$) — номер этажа и номер квартиры.

Формат выходных данных

Для каждого набора входных данных выведите в отдельную строку два целых числа — минимально возможное и максимально возможное количество квартир, которые могут быть на этаже, если квартира номер r расположена на этаже номер f .

Если минимально возможное количество квартир не может быть однозначно определено для очередного набора входных данных, либо квартира номер r не может быть на этаже номер f , первое выведенное число для этого набора входных данных должно быть равно -1 .

Если максимально возможное количество квартир не может быть однозначно определено для очередного набора входных данных, либо квартира номер r не может быть на этаже номер f , второе выведенное число для этого набора входных данных должно быть равно -1 .

Пример

стандартный ввод	стандартный вывод
4	4 5
3 11	9 -1
1 9	-1 -1
19 234	363 388
15 5433	

Замечание

В первом наборе входных данных минимально возможное количество квартир на одном этаже равно 4. Тогда квартира номер 11 будет на третьем этаже. Максимально возможное количество квартир на одном этаже равно 5. Тогда квартира номер 11 также будет на третьем этаже.

Во втором наборе входных данных минимально возможное количество квартир на одном этаже равно 9. Максимально возможное количество квартир не может быть однозначно определено, поэтому второе выведенное число для этого набора входных данных равно -1 .

В третьем наборе входных данных квартира номер 234 не может находиться на 19 этаже, поэтому нужно вывести два числа -1 .

Задача J. Поиск и замена

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

У Поликарпа есть последовательность a , состоящая из n целых чисел.
Поликарп может производить со своей последовательностью следующие операции:

- найти в своем массиве максимум (если максимумов несколько, нужно найти **самый левый** максимум);
- найти в своем массиве минимум (если минимумов несколько, нужно найти **самый правый** минимум);
- заменить оба найденных элемента на их среднее арифметическое, округленное вниз к ближайшему целому числу.

Например, если найденный максимум равен 10, а найденный минимум равен 3, то оба найденных элемента будут заменены на число 6, а если найденный максимум равен 11, а найденный минимум равен 7, то оба найденных элемента будут заменены на число 9.

Перед вами стоит задача определить, как будет выглядеть последовательность Поликарпа, если к ней последовательно применить ровно k описанных операций.

Формат входных данных

В первой строке следуют два целых числа n и k ($2 \leq n \leq 200\,000$, $1 \leq k \leq n$) — количество элементов в последовательности Поликарпа и количество операций.

Во второй строке следует последовательность целых чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$), где a_i равно i -му элементу последовательности Поликарпа.

Формат выходных данных

Выведите n целых чисел, причем i -е выведенное число должно быть равно i -му элементу в последовательности Поликарпа после последовательного применения k описанных операций.

Примеры

стандартный ввод	стандартный вывод
5 2 3 7 4 7 5	5 5 5 5 5
6 1 4 4 4 19 19 19	4 4 11 11 19 19
10 3 5 8 3 1 11 4 14 11 1 15	5 8 7 7 7 4 7 11 8 8

Замечание

В первом примере в ходе первой операции будет найден максимум в позиции 2 и минимум в позиции 1. Среднее арифметическое найденных чисел равно 5, поэтому после первой операции последовательность станет равна $[5, 5, 4, 7, 5]$. В ходе второй операции будет найден максимум в позиции 4 и минимум в позиции 3. Среднее арифметическое найденных чисел, округленное вниз, равно 5, поэтому после второй операции последовательность станет равна $[5, 5, 5, 5, 5]$.

Во втором примере в ходе первой операции будет найден максимум в позиции 4 и минимум в позиции 3. Среднее арифметическое найденных чисел, округленное вниз, равно 11, поэтому после первой операции последовательность станет равна $[4, 4, 11, 11, 19, 19]$.

Задача К. Раскраска матрицы

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

Рассмотрим матрицу из n строк и m столбцов. Изначально все клетки матрицы белые. Обозначим за (i, j) клетку на пересечении i -й строки и j -го столбца.

Мы можем проводить над ней следующую операцию любое количество раз (возможно, ноль):

- выбрать клетку (x, y) и цвет c (белый, синий или красный); покрасить всю строку x и весь столбец y в цвет c . Если какие-то клетки в этой строке или в этом столбце уже были покрашены — они перекрашиваются.

Назовем пару целых чисел (r, b) *достижимой*, если путем таких операций можно получить матрицу, в которой **ровно** r красных и b синих клеток.

Ваша задача — посчитать количество достижимых пар.

Формат входных данных

В единственной строке заданы два целых числа n и m ($2 \leq n, m \leq 16$).

Формат выходных данных

Выведите одно целое число — количество достижимых пар.

Примеры

стандартный ввод	стандартный вывод
2 2	9
3 3	45
16 15	29161
4 2	30

Замечание

В первом примере из условия достижимыми являются следующие пары: $(0, 0)$, $(0, 1)$, $(0, 3)$, $(0, 4)$, $(1, 0)$, $(1, 3)$, $(3, 0)$, $(3, 1)$, $(4, 0)$.

Чтобы получить, например, пару $(3, 1)$, можно сделать следующие действия:

- провести операцию на клетке $(1, 1)$ с красным цветом;
- провести операцию на клетке $(2, 2)$ с синим цветом;
- провести операцию на клетке $(1, 2)$ с красным цветом.

Тогда клетка $(2, 1)$ будет синей, а остальные три — красными.

Задача L. Идеальная фотография

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Поликарп является тренером команды по волейболу. Сейчас в его команде n игроков, причем рост i -го игрока равен a_i . Гарантируется, что в команде нет двух и более игроков с одинаковым ростом.

Поликарп хочет сделать идеальную фотографию игроков своей команды. Фотография считается идеальной, если выполняются следующие условия:

- все игроки команды выстроены в ряд;
- рост самого левого игрока может быть любым;
- рост каждого игрока, кроме самого левого, должен быть ровно на 1 больше, чем рост его соседа слева.

Поликарп может принять в команду любое количество игроков с любым ростом. Перед вами стоит задача определить минимальное количество новых игроков, которых нужно принять в команду, чтобы Поликарп смог сделать идеальную фотографию своей команды.

Формат входных данных

В первой строке следует целое число n ($2 \leq n \leq 200\,000$) — количество игроков в команде Поликарпа.

Во второй строке следует последовательность **различных** целых чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$), где a_i равно росту i -го игрока.

Формат выходных данных

Выведите минимальное количество новых игроков, которых нужно принять в команду, чтобы Поликарп смог сделать идеальную фотографию своей команды.

Примеры

стандартный ввод	стандартный вывод
4 200 195 197 199	2
3 180 182 181	0

Замечание

В первом примере нужно принять в команду двух игроков, одного с ростом 196, а другого с ростом 198.

Во втором примере не нужно никого принимать в команду, так как Поликарп может сделать идеальную фотографию изначального состава игроков.