

День добрый, ребята!

Рады приветствовать Вас на борту нашего судна “Журавлик 2019”! Перед тем, как отправиться в кругосветное путешествие, мы хотим пожелать Вам удачи в ваших приключениях!

Вперед программисты!



Задача А. ЧЕМОДАНЫ.

Перед взлётом Вам, как опытному путешественнику, поручили проверить вместимость багажного отделения. Известно, что есть N стеллажей, на каждый из которых помещено ровно M чемоданов. Также известно, что самолёт может взлететь, если количество чемоданов меньше K , в другом случае, самолёт считается перегруженным. Вам нужно проверить, будет ли самолёт перегружен.

Входные данные

Даны три натуральных числа: N , M и K ($N, M \leq 100$, $K \leq 10000$).

Выходные данные

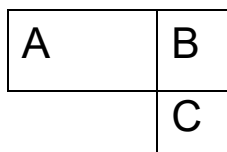
Выведите «YES», если самолёт перегружен и «NO» в противном случае.

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	4 5 20	YES
2	4 5 21	NO

Задача В. КОВРЫ.

Всем известны великолепные турецкие ковры ручной работы. Некоторые туристы, кстати, ездят в Турцию специально за ними. Вот и ваш знакомый попросил вас привезти ковёр для его коллекции. Он сообщил, что у него уже имеются три ковра с периметрами A , B и C , причём они расположены вот таким образом:



Ваш друг - перфекционист, и он хочет достроить четвёртым ковром прямоугольник. По заданным трём периметрам определите периметр четвёртого ковра, который друг просит Вас купить.

Входные данные

Вводятся три числа A , B , C – периметры имеющихся ковров.

Выходные данные

Вывести одно число – периметр четвёртого ковра.

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	24 22 20	22

Задача С. УДАЛЯЙ СЛЕВА.

Устав от выбора ковра, Вы решили отдохнуть на пляже. А где лучшие пляжи? Правильно – на Гоа, куда Вы незамедлительно отправились. Но, увы, Ваш пытливый ум не дал Вам покоя даже здесь! Подходя к воде, Вы заметили, что на песке друг под другом, перпендикулярно кромке воды, с началом у моря написаны два слова. Начался прилив, и волна начала медленно стирать буквы с песка. При этом длина слова каждый раз уменьшается на единицу. Вам стало интересно, какое наименьшее кол-во букв может стереть волна, прежде чем слова станут одинаковыми.

Например:

- После одной волны строка “where” превратится в “here”
- После одной волны строка “a” превратится в пустую строку “”

Напишите программу, которая найдет минимальное количество волн, чтобы сделать две заданные строки s и t равными.

Посмотрите в примеры для лучшего понимания.

Входные данные

В первой строке входных данных записана строка s . Во второй строке входных данных записана строка t . Строки состоят только из строчных букв латинского алфавита. Длина каждой строки от 1 до $2 \cdot 10^5$ букв.

Выходные данные

Выведите искомое наименьшее количество смытых букв. Возможно, что в итоге обе строки окажутся равны пустой строке и равны между собой. В этом случае, очевидно, ответ — это сумма длин заданных строк.

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	test west	2
2	codeforces yes	9
3	test yes	7

Примечание (*настоятельно рекомендуем прочитать*):

В первом примере волна смочет одну букву в первой строке и одну букву во второй строке. В результате обе строки будут равны «est».

Во втором примере волна смочет 8 букв в строке «codeforces». В результате будет получена строка «codeforces» → «es». В строке «yes» волна смочет только одну букву. В результате останется такая же строка «yes» → «es».

В третьем примере строки можно сделать равными только полностью их смыслом. То есть, в итоге обе строки окажутся пустыми.

Задача D. НОМЕР ТЕРМИНАЛА.

Вам надоел пляжный отдых, и вы решили отправиться куда-нибудь в Европу. К сожалению, вы забыли номер терминала, к которому Вам нужно. Добрый индус-сотрудник аэропорта рассказал Вам интересную вещь: чтобы узнать номер своего терминала, человек должен из номера своего рейса вычитать сумму его цифр до тех пор, пока не получится однозначное число. Вы решили написать программу, которая по заданному номеру рейса находит номер нужного терминала.

Входные данные

На входе содержится число a ($10 < a < 10^8$).

Выходные данные

Выведете номер терминала.

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	10	9

Задача E. ОЧЕРЕДЬ.

В Европе Вы первым делом отправились, конечно, в Лувр. Замечательное место, с не менее замечательными очередями. Из-за большого потока туристов, управление Лувра приняло решение – запускать в музей людей по парам. Пока Вы стояли в очереди, Вам стало интересно, сколькими способами можно выбрать пару из всей очереди, состоящей из N человек.

Справка:

А Вы знали, что если сказать число громче, то оно станет больше? Просто в математике запись $N!$ означает факториал – произведение чисел от 1 до N . Например, $5! = 1*2*3*4*5 = 120$. При этом $0! = 1$.

Формула, которая вам понадобится для подсчёта количества вариантов выборки пар выглядит так: $Answer = \frac{N!}{M!*(N-M)!}$. В нашем случае $M=2$

Входные данные

Дано число N – количество человек в очереди

$$2 \leq N \leq 10000$$

Выходные данные

Выведите единственное число – ответ на задачу.

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	5	10
2	2	1
3	1	0
4	15	105

Задача F. БИЛЕТ

После Франции Вы непременно надумали посетить маленькое соседнее королевство Монако. Но Вы знаете, что в Монако Вас пустят лишь тогда, когда Ваша виза удовлетворяет следующим условиям:

Если Вы можете разбить номер Вашей визы на **хотя бы два** отрезка (отрезок — это последовательность подряд идущих цифр). Другими словами, Вам нужно поставить разделители между некоторыми цифрами номера таким образом, что **каждый элемент принадлежит ровно одному отрезку** и, если результирующее разбиение будет представлено в виде последовательности целых чисел, то каждый следующий элемент этой последовательности будет **строго больше**, чем предыдущий.

Более формально: если результирующее разбиение последовательности — это t_1, t_2, \dots, t_k , где k — количество элементов в разбиении, то для каждого i от 1 до $k-1$ должно выполняться условие $t_i < t_{i+1}$ (используется **численное** сравнение, то есть сравниваются численные интерпретации строк).

Например, если $s=654$, то Вы можете разбить ее на части $[6,54]$ и это будет являться подходящим разбиением. Но если Вы разобьете ее на части $[65,4]$, то это будет являться плохим разбиением, потому что $65 > 4$. Если $s=123$, то Вы можете разбить его на части $[1,23]$ и $[1,2,3]$, но не на части $[12,3]$.

Ваша задача — найти подходящее разбиение номера визы с **минимальным** количеством чисел в группе и с **минимальным** меньшим членом, или сказать, что такого разбиения не существует, и Вы, увы, остаётесь во Франции.

Входные данные

Первая строка содержит одно целое число n ($2 \leq n \leq 300$) — количество цифр в номере.

Вторая строка содержит одну строку s длины n , состоящую только из цифр от 1 до 9.

Выходные данные

Если последовательность цифр не может быть разбита на **хотя бы две** части способом, описанном в условии задачи, выведите единственную строку «NO» для этого запроса.

Иначе в первой строке ответа на этот запрос выведите «YES», во второй строке выведите k — количество элементов в вашем разбиении последовательности и в третьей строке выведите k строк t_1, t_2, \dots, t_k — ваше разбиение. Части должны быть выведены в порядке следования цифр в заданной строке. Это означает, что если записать выведенные подстроки одну за другой, сохранив их порядок, то получится строка s .

Посмотрите в примеры для лучшего понимания.

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	3 123	YES 2 1 23
2	2 21	NO

Задача G. БРОСКИ КУБИКА.

В Монако Вы познакомились с Игорем – мастером азартных игр. Он показал Вам одну очень забавную игру.

У Игоря есть шестигранный игральный кубик. На каждой грани этого кубика записаны натуральные числа 2 до 7 (все числа на гранях различны, так что это почти обычный игральный кубик).

Игорь хочет получить ровно x очков, бросая свой кубик. Количество очков равно сумме чисел на верхней грани кубика по всем броскам, сделанным Игорем.

Вы должны угадать минимальное количество бросков, необходимое Игорю, чтобы он смог получить ровно x очков за такое количество бросков. Игорь настолько удачлив, что если есть ненулевая вероятность выбросить ровно x очков выбранным количеством бросков, то у него это получится. Ваша задача — найти это количество. Гарантируется, что ответ существует.

Входные данные

Первая строка входных данных содержит одно целое число x ($2 \leq x \leq 1000$) — количество очков, которое хочет получить Игорь.

Выходные данные

Выведите единственное число – ответ на задачу.

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	13	2
2	21	3
3	5	1

Примечание

В первом примере Игорь может бросить кубик два раза и получить 7 и 6 очков.

Во втором примере Игорь может бросить кубик 3 раза и получить очки 7, 7 и 7.

В третьем примере Игорь может бросить кубик один раз и получить 5 очков.

Задача N. СЧАСТЛИВОЕ ТАКСИ

Из Монако, счастливый с деньгами, Вы отправляетесь в Италию. Наилучший вариант – это такси. Вам повезло: вы попали в счастливое такси. Вы ничего не поняли, но таксист объяснил вам суть. Такси считается счастливым, если после описанного ниже преобразования с номером машины, в нём остаётся хотя бы одна цифра.

Суть преобразования: в последовательности цифр все цифры большие 5 разделить на 2 (отбросив остаток при необходимости), а затем удалить из полученной последовательности все чётные цифры.

Напишите программу, которая определяет, является ли такси счастливым.

Входные данные

Вводятся 4 цифры в одной строке без пробелов – номер такси.

Выходные данные

Выведите номер после преобразования. Если ответ – пустая строка – выведите «NO».

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	0586	53
2	0000	NO

Задача I. ЦЕЗАРЬ БЕЗ КУРИЦЫ

В Италии Вы первым делом решили посетить Римский Колизей. Экскурсовод рассказывал Вам о Цезаре, а точнее о его способности заниматься несколькими делами сразу.

На каждый свой день Цезарь планировал n дел. i -е дело он выполнял за a_i единиц времени. Гарантируется, что для любого $i \in [2..n]$ будет выполняться $a_i \geq a_{i-1}$, то есть последовательность времён отсортирована по неубыванию.

Также у Цезаря хватало сил на то, чтобы не более k любых дел выполнить за x единиц времени вместо a_i ($x < \min_{i=1}^n a_i$).

Цезарь был очень ответственным человеком, поэтому он выполнял все n дел. Вам стало интересно, за какое минимальное время он мог это сделать.

Входные данные

В первой строке входных данных задано три целых числа n, k, x ($1 \leq k \leq n \leq 100, 1 \leq x \leq 99$) — общее количество дел, количество дел, которые Цезарь может выполнить за x единиц времени, и само время x .

В следующей строке задано n целых чисел a_i ($2 \leq a_i \leq 100$) — время, которое Цезарю необходимо потратить на выполнение дела с номером i .

Гарантируется, что $x < \min_{i=1}^n a_i$, а также, что для любого $i \in [2..n]$ будет выполняться $a_i \geq a_{i-1}$.

Выходные данные

Выведите единственное целое число — минимальное время, которое понадобится Цезарю на выполнение всех n дел.

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	4 2 2 3 6 7 10	13
2	5 2 1 100 100 100 100 100	302

Примечание

В первом тестовом примере выгоднее всего выполнить дела с номерами 3 и 4 за время $x=2$ вместо a_3 и a_4 соответственно. Тогда ответ будет равняться $3 + 6 + 2 + 2 = 13$.

Во втором тестовом примере можно выбрать любые два дела и выполнить их за время x вместо a_i . Тогда ответ будет равняться $100 \cdot 3 + 2 \cdot 1 = 302$.

Задача J. ЯПОНСКИЙ РЕСТОРАН.

После солнечной Италии, Вы захотели чего-нибудь восточного. Не долго думая, Вы пакуете чемоданы и направляетесь в Японию.

Кто бы что ни говорил, а самое прекрасное в Японии – суши. Поэтому Вы сразу отправляетесь в местный ресторанчик, чтобы как следует полакомиться этим замечательным блюдом. Чтобы Вам было не так скучно, Вы решаете позвать с собой своего давнего японского друга Као Ши.

Ресторан оказался немного необычным: на выбор клиенту предлагаются n суши, выложенных в ряд, а клиент должен заказать себе некоторый подотрезок из подряд идущих суши.

Суши бывают двух видов: либо с тунцом, либо с угрем. Обозначим тип i -го слева суши как t_i , где $t_i = 1$ обозначает, что это суши с тунцом, а $t_i = 2$ означает, что оно с угрем.

Вы не любите суши с тунцом, а Као Ши не любит суши с угрем. Вы хотите выбрать последовательный подотрезок суши так, чтобы в нем было поровну суши обоих типов, и каждая половина отрезка содержала только суши одного типа. Например, подотрезок $[2,2,2,1,1,1]$ подходит, а подотрезок $[1,2,1,2,1,2]$ — нет, потому что в обеих половинах есть суши обоих типов.

Найдите длину наибольшего подотрезка из подряд идущих суши, чтобы полакомиться как можно большим количеством суши.

Входные данные

Первая строка содержит одно целое число n ($2 \leq n \leq 100000$) — количество суши.

Вторая строка содержит n целых чисел t_1, t_2, \dots, t_n ($t_i=1$, что означает суши с тунцом, или $t_i=2$, что означает суши с угрем), обозначающие типы суши слева направо.

Гарантируется, что есть хотя бы одно суши каждого из типов. Обратите внимание, это означает, что существует, по крайней мере, один подходящий Вам подотрезок.

Выходные данные

Выведите одно целое число — максимально возможную длину подходящего Вам отрезка из подряд идущих суши.

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	7 2 2 2 1 1 2 2	4
2	6 1 2 1 2 1 2	2
3	9 2 2 1 1 1 2 2 2 2	6

Примечание

В первом примере Вы можете выбрать подотрезок $[2,2,1,1]$ или подотрезок $[1,1,2,2]$, длина которых 4.

Во втором примере невозможно выбрать что-то другое, кроме $[2,1]$ или $[1,2]$, длина которых 2.

В третьем примере лучший выбор — подотрезок $[1,1,1,2,2,2]$.

Вот и завершилось наше кругосветное путешествие. Мы вернулись в Вятские Поляны и теперь ожидаем результатов турнира. Надеемся, вам понравились задачи, и вы получили удовольствие от их решения.