

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

В. Троичная строка

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Вам дана строка s , каждый символ которой — 1, 2 или 3. Вы должны выбрать кратчайшую непрерывную подстроку s , в которой каждый из трех символов встречается хотя бы один раз.

Непрерывная подстрока s — строка, которую можно получить из s удалением какого-то количества (возможно, ни одного) символов из начала строки s и какого-то количества (возможно, ни одного) символов из конца строки s .

Входные данные

В первой строке задано одно целое число t ($1 \leq t \leq 20000$) — количество наборов входных данных.

Каждый набор входных данных состоит из одной строки s ($1 \leq |s| \leq 200000$). Гарантируется, что каждый символ s — 1, 2 или 3.

Сумма длин всех строк не превосходит 200000.

Выходные данные

Для каждого набора тестовых данных выведите одно число — длину кратчайшей подстроки s , содержащей символы всех трех типов. Если такой подстроки нет, выведите 0.

Пример

входные данные	Скопировать
7 123 1222213333332 112233 332211 12121212 333333 31121	
выходные данные	Скопировать
3 3 4 4 0 0 4	

Примечание

Рассмотрим пример из условия:

В первом наборе входных данных можно использовать подстроку 123.

Во втором наборе входных данных можно использовать подстроку 213.

В третьем наборе входных данных можно использовать подстроку 1223.

В четвертом наборе входных данных можно использовать подстроку 3221.

В пятом наборе входных данных в s нету символа 3.

В шестом наборе входных данных в s нету символа 1.

В седьмом наборе входных данных можно использовать подстроку 3112.

С. Чередующая подпоследовательность

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

Напомним, что последовательность b является подпоследовательностью последовательности a , если b может быть получена из a путем удаления нуля или более элементов без изменения порядка оставшихся элементов. Например, если $a = [1, 2, 1, 3, 1, 2, 1]$, то возможные подпоследовательности: $[1, 1, 1, 1]$, $[3]$ и $[1, 2, 1, 3, 1, 2, 1]$, но не $[3, 2, 3]$ и $[1, 1, 1, 1, 2]$.

Вам задана последовательность a , состоящая из n положительных и отрицательных элементов (в последовательности нет нулей).

Ваша задача выбрать **максимальную по размеру** (длине) *чередующуюся* подпоследовательность заданной последовательности (то есть знак каждого следующего элемента противоположен знаку текущего элемента, например, положительный-отрицательный-положительный и так далее или отрицательный-положительный-отрицательный и так далее). Из всех таких подпоследовательностей вам нужно выбрать ту, которая имеет **максимальную сумму** элементов.

Другими словами, если максимальная длина *чередующейся* подпоследовательности равна k , то ваша задача — найти **максимальную сумму** элементов какой-то *чередующейся* подпоследовательности длины k .

Вам нужно ответить на t независимых наборов тестовых данных.

Входные данные

Первая строка теста содержит одно целое число t ($1 \leq t \leq 10^4$) — количество наборов тестовых данных. Затем следуют t наборов тестовых данных.

Первая строка набора тестовых данных содержит одно целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество элементов в a . Вторая строка набора входных данных содержит n целых чисел a_1, a_2, \dots, a_n ($-10^9 \leq a_i \leq 10^9, a_i \neq 0$), где a_i — i -й элемент в a .

Гарантируется, что сумма чисел n по всем наборам тестовых данных не превосходит $2 \cdot 10^5$ ($\sum n \leq 2 \cdot 10^5$).

Выходные данные

Для каждого набора тестовых данных выведите ответ на него — **максимальную сумму максимальной по размеру** (длине) *чередующейся* подпоследовательности a .

Пример

входные данные	Скопировать
4 5 1 2 3 -1 -2 4 -1 -2 -1 -3 10 -2 8 3 8 -4 -15 5 -2 -3 1 6 1 -1000000000 1 -1000000000 1 -1000000000	
выходные данные	Скопировать
2 -1 6 -2999999997	

Примечание

В первом наборе тестовых данных примера одним из возможных ответов является $[1, 2, 3, \underline{-1}, -2]$.

Во втором наборе тестовых данных примера одним из возможных ответов является $[-1, -2, \underline{-1}, -3]$.

В третьем наборе тестовых данных примера одним из возможных ответов является $[\underline{-2}, 8, 3, \underline{8}, \underline{-4}, -15, \underline{5}, \underline{-2}, -3, \underline{1}]$.

В четвертом наборе тестовых данных примера одним из возможных ответов является $[\underline{1}, \underline{-1000000000}, \underline{1}, \underline{-1000000000}, \underline{1}, \underline{-1000000000}]$.

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

В. Камни Курияма Мираи

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Курияма Мираи убила много чудовищ и получила в награду много (а именно n) камней. Все камни Курияма Мираи пронумерованы от 1 до n . Цена i -го камня равняется v_i . Курияма Мираи хочет узнать некоторую информацию о своих камнях, с этой целью она задает вам вопросы. Каждый вопрос имеет один из следующих видов:

1. Она называет вам два числа, l и r ($1 \leq l \leq r \leq n$), а вы должны сказать ей, чему равна сумма $\sum_{i=l}^r v_i$.
2. Пусть u_i равняется цене i -го по дешевизне камня (цена, которая будет на i -ом месте, если расположить все цены камней в порядке неубывания). На этот раз она называет вам два числа, l и r ($1 \leq l \leq r \leq n$), а вы должны сказать ей, чему равна сумма $\sum_{i=l}^r u_i$.

Вам нужно ответить правильно на ее вопросы, или Курияма Мираи скажет «fuyukai desu» (яп. неприятно) и расстроится.

Входные данные

В первой строке записано целое число n ($1 \leq n \leq 10^5$). Во второй строке записано n целых чисел: v_1, v_2, \dots, v_n ($1 \leq v_i \leq 10^9$) – цены камней.

В третьей строке записано целое число m ($1 \leq m \leq 10^5$) — количество вопросов Курияма Мираи. Затем следует m строк, в каждой строке записано три целых числа: $type, l$ и r ($1 \leq l \leq r \leq n; 1 \leq type \leq 2$), описывающих текущий вопрос. Если $type$ равняется 1, то вам нужно вывести ответ на вопрос первого типа, в противном случае надо вывести ответ на вопрос второго типа.

Выходные данные

Выведите m строк. В каждой строке должно быть записано целое число — ответ на вопрос Курияма Мираи. Выводите ответы на вопросы в порядке их следования во входных данных.

Примеры

входные данные	Скопировать
6 6 4 2 7 2 7 3 2 3 6 1 3 4 1 1 6	
выходные данные	Скопировать
24 9 28	

входные данные	Скопировать
4 5 5 2 3 10 1 2 4 2 1 4 1 1 1 2 1 4 2 1 2 1 1 1 1 3 3 1 1 3 1 4 4 1 2 2	
выходные данные	Скопировать
10 15 5 15 5 5	

2
12
3
5

Примечание

Пожалуйста, обратите внимание, что для хранения ответов на вопросы может быть не достаточно 32-битных целых чисел.

[Codeforces](#) (с) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 04.11.2024 20:21:53^{UTC+5} (h1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#)

При поддержке

**ІІТМО**

CODEFORCES

Sponsored by TON

|





[Кар6502](#) | [Выйти](#)

ЗАДАЧИ

ОТОСЛАТЬ

СТАТУС

ПОЛОЖЕНИЕ

ЗАПУСК

А. Игра с переворачиванием

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

Яхубу стало скучно и он придумал игру, в которую надо играть на бумаге.

Яхуб выписывает n целых чисел a_1, a_2, \dots, a_n . Каждое из этих целых чисел может быть равно 0 или 1. Ему разрешено выполнить ровно одно действие: выбрать два индекса i и j ($1 \leq i \leq j \leq n$) и перевернуть все значения a_k , позиции которых находятся на отрезке $[i, j]$ (то есть $i \leq k \leq j$). Перевернуть значение x , значит выполнить операцию $x = 1 - x$.

Цель игры — получить максимальное количество единиц после **ровно** одного хода. Напишите программу, которая решает маленькую игру Яхуба.

Входные данные
В первой строке содержится целое число n ($1 \leq n \leq 100$). Во второй строке записано n целых чисел: a_1, a_2, \dots, a_n .
Гарантируется, что каждое из этих n чисел равняется либо 0, либо 1.

Выходные данные
Выведите целое число — максимальное количество единиц, которое можно получить после ровно одного хода.

Примеры

входные данные

5
1 0 0 1 0

выходные данные

4

входные данные

4
1 0 0 1

выходные данные

4

Примечание
В первом случае надо перевернуть отрезок от 2 до 5 ($i = 2, j = 5$). Такой переворот превращает последовательность в: [1 1 1 0 1]. Итого, в ней теперь четыре единицы. Мы никак не сможем превратить последовательность в [1 1 1 1 1].

Во втором случае, если мы перевернем только второй и третий элементы ($i = 2, j = 3$), все числа будут равны 1.

В. Бал в БерлГУ

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

По случаю 100500-летия Берляндского государственного университета совсем скоро состоится бал! Уже n юношей и m девушек во всю репетируют вальс, менуэт, полонез и кадрили.

Известно, что на бал будут приглашены несколько пар юноша-девушка, причем уровень умений танцевать партнеров в каждой паре должен отличаться не более чем на единицу.

Для каждого юноши известен уровень его умения танцевать. Аналогично, для каждой девушки известен уровень ее умения танцевать. Напишите программу, которая определит наибольшее количество пар, которое можно образовать из n юношей и m девушек.

Входные данные

В первой строке записано целое число n ($1 \leq n \leq 100$) — количество юношей. Вторая строка содержит последовательность a_1, a_2, \dots, a_n ($1 \leq a_i \leq 100$), где a_i — умение танцевать i -го юноши.

Аналогично, третья строка содержит целое m ($1 \leq m \leq 100$) — количество девушек. В четвертой строке содержится последовательность b_1, b_2, \dots, b_m ($1 \leq b_j \leq 100$), где b_j — умение танцевать j -й девушки.

Выходные данные

Выведите единственное число — искомое максимальное возможное количество пар.

Примеры

входные данные	Скопировать
4 1 4 6 2 5 5 1 5 7 9	
выходные данные	Скопировать
3	
входные данные	Скопировать
4 1 2 3 4 4 10 11 12 13	
выходные данные	Скопировать
0	
входные данные	Скопировать
5 1 1 1 1 1 3 1 2 3	
выходные данные	Скопировать
2	

В. Илья и запросы

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Лев Илья хочет помочь всем своим друзьям со сдачей экзаменов. Чтобы сдать экзамен по информатике нужно решить следующую задачу.

Даны строка $s = s_1s_2...s_n$ (n — длина строки), состоящая только из символов «.», «#», и m запросов. Каждый запрос описывается парой целых чисел l_i, r_i ($1 \leq l_i < r_i \leq n$). Ответ на запрос l_i, r_i — это количество таких целых чисел i ($l_i \leq i < r_i$), что $s_i = s_{i+1}$.

Лев Илья хочет помочь друзьям, но кто же поможет ему. Помогите Льву Илье, решите задачу.

Входные данные

В первой строке записана строка s длины n ($2 \leq n \leq 10^5$). Гарантируется, что заданная строка состоит только из символов «.», «#».

В следующей строке записано целое число m ($1 \leq m \leq 10^5$) — количество запросов. В каждой из следующих m строк записано описание соответствующего запроса. В i -той строке записаны целые числа l_i, r_i ($1 \leq l_i < r_i \leq n$).

Выходные данные

Выведите m целых чисел — ответы на запросы в том порядке, в котором запросы заданы во входных данных.

Примеры

входные данные	Скопировать
..... 4 3 4 2 3 1 6 2 6	
выходные данные	Скопировать
1 1 5 4	

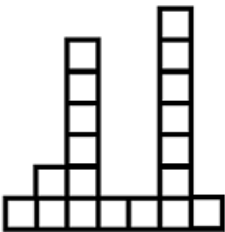
входные данные	Скопировать
#...### 5 1 3 5 6 1 5 3 6 3 4	
выходные данные	Скопировать
1 1 2 2 0	

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

В. Забор

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

Перед домом Поликарпа находится забор, который состоит из n одинаковых по ширине досок, расположенных одна за другой слева направо. Высота i -ой доски составляет h_i метров, разные доски могут иметь различные высоты.



Забор для $n = 7$ и $h = [1, 2, 6, 1, 1, 7, 1]$

Поликарп приобрел рояль и теперь ищет наилучший способ занести рояль в свой дом. Для того, чтобы осуществить задуманное, ему придется выломать ровно k подряд идущих досок в заборе. Так как высокие доски выламывать сложнее, Поликарп хочет найти такие k последовательных досок, что сумма их высот минимальна.

Напишите программу, которая найдет номера k последовательных досок с наименьшей суммой высот. Обратите внимание, забор не окружает дом Поликарпа, а находится перед ним (другими словами, забор не зациклен).

Входные данные

В первой строке входных данных содержатся целые числа n и k ($1 \leq n \leq 1.5 \cdot 10^5$, $1 \leq k \leq n$) — количество досок в заборе и ширина проема для рояля. Вторая строка содержит последовательность целых чисел h_1, h_2, \dots, h_n ($1 \leq h_i \leq 100$), где h_i — высота i -ой доски забора.

Выходные данные

Выведите такое j , что сумма высот досок $j, j + 1, \dots, j + k - 1$ — наименьшая возможная. Если таких j несколько, то выведите любое из них.

Примеры

входные данные	Скопировать
7 3 1 2 6 1 1 7 1	
выходные данные	Скопировать
3	

Примечание

В примере требуется найти три последовательные доски с минимальной суммой высот. В данном случае три доски с номерами 3, 4 и 5 обладают требуемым свойством и имеют суммарную высоту 8.



В. Интересный напиток

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Рабочий Василий очень любит отдыхать после работы, поэтому его часто можно встретить в каком-нибудь баре. Как и все программисты, Василий очень любит напиток «Пикола», который продаётся в n различных магазинах города. Известно, что в i -м магазине бутылка напитка стоит x_i монет.

Василий планирует покупать одну бутылку своего любимого напитка на протяжении q дней. Он знает, что в i -й день у него с собой будет m_i монет, и теперь он хочет для каждого из дней узнать, в каком количестве магазинов он сможет купить одну бутылочку Пиколы.

Входные данные

В первой строке входных данных содержится целое число n ($1 \leq n \leq 100\,000$) — количество магазинов в городе, продающих любимый напиток Василия.

Во второй строке входных данных содержится n чисел x_i ($1 \leq x_i \leq 100\,000$) — цена за одну бутылку напитка в i -м магазине.

В третьей строке входных данных содержится число q ($1 \leq q \leq 100\,000$) — количество дней, в течение которых Василий планирует покупать напиток.

В следующих q строках входных данных содержатся целые числа m_i ($1 \leq m_i \leq 10^9$) — количество денег, которое есть у Василия в i -й день.

Выходные данные

Выведите q целых чисел, i -е из которых должно равняться количеству магазинов, в которых Василий может купить одну бутылочку любимого напитка в день i .

Пример

входные данные	Скопировать
<pre>5 3 10 8 6 11 4 1 10 3 11</pre>	
выходные данные	Скопировать
<pre>0 4 1 5</pre>	

Примечание

В первом запросе ни в одном магазине Василию не хватит денег.

Во втором запросе Василию хватит денег, чтобы купить напиток в магазинах под номерами 1, 2, 3 и 4.

В третьем запросе Василию хватит денег, чтобы купить напиток в магазине под номером 1.

И в последнем запросе Василий может купить свой напиток в любом магазине.

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

С. Малинки

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Вам дан массив целых чисел a_1, a_2, \dots, a_n и число k ($2 \leq k \leq 5$). За одну операцию вы можете сделать следующее:

- Выбрать индекс $1 \leq i \leq n$,
- Сделать $a_i = a_i + 1$.

Найдите наименьшее количество операций, которые нужно сделать, чтобы произведение всех чисел массива $a_1 \cdot a_2 \cdot \dots \cdot a_n$ делилось на k .

Входные данные

Каждый тест состоит из нескольких наборов входных данных. Первая строка содержит единственное целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных. Далее следует их описание.

Первая строка каждого набора входных данных содержит два целых числа n и k ($2 \leq n \leq 10^5$, $2 \leq k \leq 5$) — размер массива a и число k .

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

Гарантируется, что сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных выведите наименьшее количество операций, которое нужно сделать, чтобы произведение всех чисел массива делилось на k .

Пример

входные данные	Скопировать
15 2 5 7 3 3 3 7 4 1 5 2 9 7 7 3 9 5 5 5 4 1 2 3 7 4 9 5 1 5 9 5 1 3 4 6 3 6 3 4 6 1 5 3 4 1 5 9 4 4 1 4 1 1 3 4 3 5 3 4 5 8 9 9 3 2 5 1 6 2 5 10 10 4 5 1 6 1 1 2 5 7 7	
выходные данные	Скопировать
2 2 1 0 2 0 1	

2
0
1
1
4
0
4
3

Примечание

В первом наборе входных данных нужно два раза выбрать индекс $i = 2$. После этого массив будет равен $a = [7, 5]$.
Произведение всех чисел массива равно 35.

В четвертом наборе входных данных произведение чисел массива равно 120, что уже делится на 5, поэтому не нужно применять операции.

В восьмом наборе входных данных можно сделать две операции, выбрав $i = 2$ и $i = 3$ в любом порядке. После этого массив будет равен $a = [1, 6, 10]$. Произведение чисел массива равно 60.

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 04.11.2024 20:21:22^{UTC+5} (h1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#)

При поддержке

**ИТМО**

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

В. Рудольф и 121

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

У Рудольфа есть массив a из n целых чисел, элементы которого пронумерованы от 1 до n .

За одну операцию он может выбрать индекс i ($2 \leq i \leq n - 1$) и присвоить:

- $a_{i-1} = a_{i-1} - 1$;
- $a_i = a_i - 2$;
- $a_{i+1} = a_{i+1} - 1$.

Рудольф может применить эту операцию любое число раз. Любой индекс i может быть использован ноль или более раз.

Может ли он с помощью этой операции сделать все элементы массива равными нулю?

Входные данные

Первая строка входных данных содержит одно целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных в тесте.

Первая строка каждого набора содержит одно целое число n ($3 \leq n \leq 2 \cdot 10^5$) — количество элементов массива.

Вторая строка каждого набора содержит n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_j \leq 10^9$) — элементы массива.

Гарантируется, что сумма значений n по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных выведите «YES», если с помощью описанных операций можно сделать все элементы массива нулями. Иначе выведите «NO».

Вы можете выводить каждую букву в любом регистре (строчную или заглавную). Например, строки «yEs», «yes», «Yes» и «YES» будут приняты как положительный ответ.

Пример

входные данные	Скопировать
<pre> 7 5 1 3 5 5 2 5 2 4 4 5 1 5 0 1 3 3 1 6 5 6 0 2 3 0 4 1 2 7 2 3 7 1 0 4 1 1 1 1 </pre>	
выходные данные	Скопировать
<pre> YES NO YES NO NO NO NO NO </pre>	

Примечание

В первом примере исходный массив равен $[1, 3, 5, 5, 2]$, чтобы сделать все его элементы равными нулю, Рудольф может действовать так:

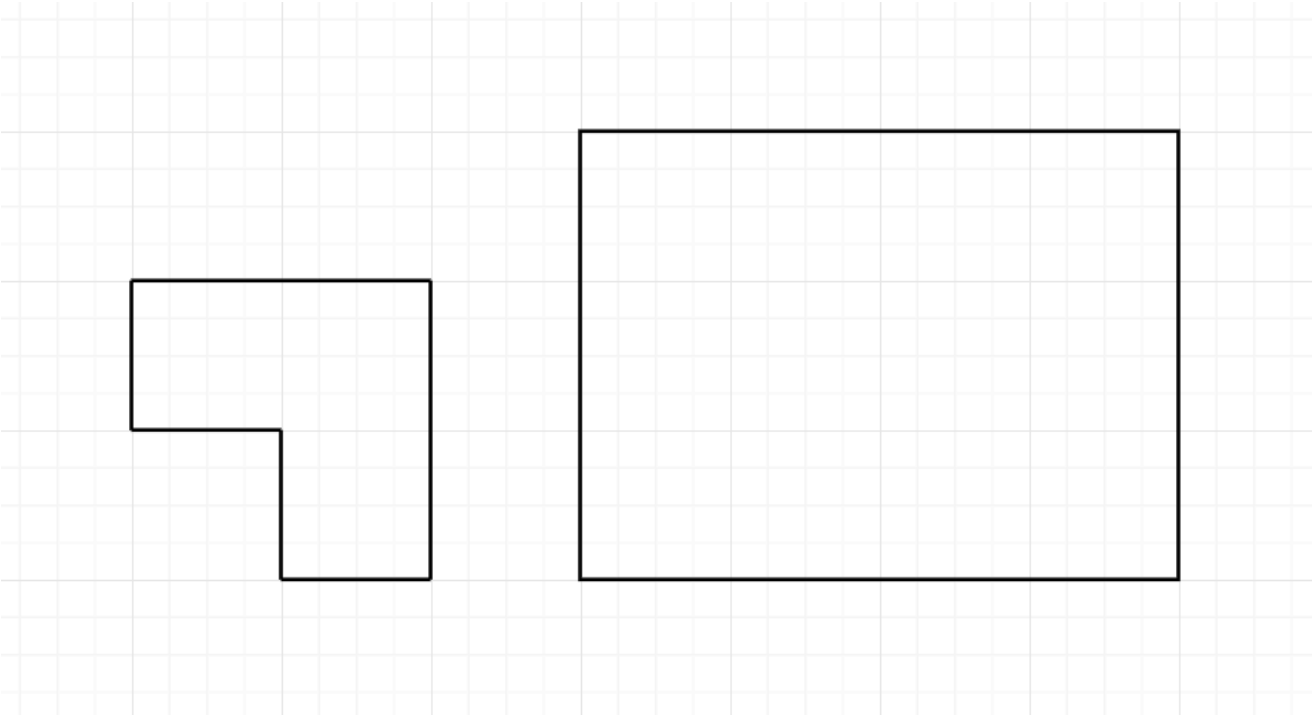
- применить операцию на $i = 4$ и получить массив $[1, 3, 4, 3, 1]$;
- применить операцию на $i = 3$ и получить массив $[1, 2, 2, 2, 1]$;
- применить операцию на $i = 2$ и получить массив $[0, 0, 1, 2, 1]$;
- применить операцию на $i = 4$ и получить массив $[0, 0, 0, 0, 0]$.

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

А. Заполнение формами

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

Вам дано целое число n . Найдите количество способов заполнить все $3 \times n$ плитки формой, описанной на рисунке ниже. Нельзя, чтобы после заполнения были пустые плитки. Формы не могут перекрываться.



Эта картина описывает ситуацию, когда $n = 4$. Слева — форма, а справа — $3 \times n$ плиток.

Входные данные

Первая строка содержит одно целое число n ($1 \leq n \leq 60$) — длину.

Выходные данные

Выведите количество способов.

Примеры

входные данные	Скопировать
4	
выходные данные	Скопировать
4	
входные данные	Скопировать
1	
выходные данные	Скопировать
0	

Примечание

В первом примере есть всего 4 возможных способов заполнения.

Во втором примере вы не можете заполнить 3×1 плиток.

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

В. Новогоднее число

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Поликарпу запомнился 2020-й год, и он рад приходу нового 2021-го года. Чтобы запомнить такой замечательный момент, Поликарп хочет представить число n в виде суммы некоторого количества чисел 2020 и некоторого количества чисел 2021.

Например, если:

- $n = 4041$, то число n представимо в виде суммы $2020 + 2021$;
- $n = 4042$, то число n представимо в виде суммы $2021 + 2021$;
- $n = 8081$, то число n представимо в виде суммы $2020 + 2020 + 2020 + 2021$;
- $n = 8079$, то число n нельзя представить в виде суммы чисел 2020 и 2021.

Помогите Поликарпу понять, может ли число n быть представлено в виде суммы некоторого количества чисел 2020 и некоторого количества чисел 2021.

Входные данные

Первая строка содержит одно целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных. Далее следуют t наборов входных данных.

Каждый набор входных данных содержит одно целое число n ($1 \leq n \leq 10^6$) — число, которое Поликарп хочет представить в виде суммы чисел 2020 и 2021.

Выходные данные

Для каждого набора входных данных в отдельной строке выведите:

- «YES», если число n представимо в виде суммы некоторого количества чисел 2020 и некоторого количества чисел 2021;
- «NO» в противном случае.

Вы можете выводить «YES» и «NO» в любом регистре (например, строки yEs, yes, Yes и YES будут распознаны как положительный ответ).

Пример

входные данные	Скопировать
5 1 4041 4042 8081 8079	
выходные данные	Скопировать
NO YES YES YES NO	



A. Кефа и первые шаги

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Кефа решил подзаработать денег, занимаясь различной деятельностью в интернете на протяжении ровно n дней. Он знает, что в i -й день ($1 \leq i \leq n$) он заработает a_i монет. Кефа любит прогресс, поэтому он хочет узнать длину максимального неубывающего подотрезка в последовательности a_i . Напомним, что подотрезок последовательности — это её непрерывный фрагмент. Подотрезок чисел называется неубывающим, если числа в нём следуют в порядке неубывания.

Помогите Кефе справиться с этой задачей!

Входные данные

В первой строке содержится целое число n ($1 \leq n \leq 10^5$).

Во второй строке заданы n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$).

Выходные данные

Выведите единственное целое число — длину максимального неубывающего подотрезка последовательности a .

Примеры

входные данные	Скопировать
6 2 2 1 3 4 1	
выходные данные	Скопировать
3	

входные данные	Скопировать
3 2 2 9	
выходные данные	Скопировать
3	

Примечание

В первом тесте максимальный неубывающий подотрезок это числа с третьего по пятое.

Во втором тесте максимальный неубывающий подотрезок это числа с первого по третье.

В. Торт - это ложь

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Есть клеточное поле размера $n \times m$. Вы стоите в клетке $(1, 1)$ и хотите попасть в клетку (n, m) .

Вы можете перемещаться в соседние клетки вправо или вниз. Другими словами, предположим вы находитесь в клетке (x, y) . Вы можете:

- сдвинуться вправо в клетку $(x, y + 1)$ — это перемещение стоит x бурлей;
- сдвинуться вниз в клетку $(x + 1, y)$ — это перемещение стоит y бурлей.

Можете ли вы достигнуть клетки (n, m) , потратив **ровно** k бурлей?

Входные данные

В первой строке задано одно целое число t ($1 \leq t \leq 100$) — количество наборов входных данных.

В первой и единственной строке каждого набора заданы три целых числа n, m и k ($1 \leq n, m \leq 100$; $0 \leq k \leq 10^4$) — размеры поля и точное количество денег, которое вы должны потратить.

Выходные данные

Для каждого набора входных данных, если вы можете попасть в клетку (n, m) , потратив **ровно** k бурлей, выведите YES. В противном случае выведите NO.

Вы можете вывести каждую букву в любом регистре (например, YES, Yes, yes, yEs будут распознаны как положительный ответ).

Пример

входные данные	Скопировать
<pre>6 1 1 0 2 2 2 2 2 3 2 2 4 1 4 3 100 100 10000</pre>	
выходные данные	Скопировать
<pre>YES NO YES NO YES NO</pre>	

Примечание

В первом наборе, вы уже в нужной клетке, потратив 0 бурлей.

Во втором, третьем и четвертом наборах, есть только два пути из $(1, 1)$ в $(2, 2)$: $(1, 1) \rightarrow (1, 2) \rightarrow (2, 2)$ или $(1, 1) \rightarrow (2, 1) \rightarrow (2, 2)$. Оба способа стоят $1 + 2 = 3$ бурлей, то есть это единственно возможное количество денег, которое вы можете потратить.

В пятом наборе, есть единственный путь из $(1, 1)$ в $(1, 4)$ и его стоимость равна $1 + 1 + 1 = 3$ бурлей.

Codeforces

(с) Copyright 2010-2024 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 04.11.2024 20:21:02^{UTC+5} (h1).

Мобильная версия, переключиться на десктопную.

Privacy Policy

При поддержке

https://codeforces.com/problemset/problem/1519/B?mobile=true

1/2