

А. Сумма круглых чисел

ограничение по времени на тест: 1 секунда
 ограничение по памяти на тест: 256 мегабайт
 ввод: стандартный ввод
 вывод: стандартный вывод

Положительное целое число называется *круглым*, если оно имеет вид $d00 \dots 0$. Иными словами, положительное целое число является круглым, если все его цифры, кроме самой левой (старшей), равны нулю. В частности, все числа от 1 до 9 (включительно) являются круглыми.

Например, следующие числа являются круглыми: 4000, 1, 9, 800, 90. Например, следующие числа **не** являются круглыми: 110, 707, 222, 1001.

Вам задано целое положительное число n ($1 \leq n \leq 10^4$). Представьте число n как сумму круглых чисел, используя минимальное количество слагаемых. Иными словами, вам надо разложить заданное число n в сумму наименьшего количества слагаемых, каждое из которых является круглым числом.

Входные данные

В первой строке записано целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных в тесте. Далее следуют t наборов входных данных.

Каждый набор представляет собой строку, в которой записано целое число n ($1 \leq n \leq 10^4$).

Выходные данные

Выведите t ответов на наборы входных данных. Каждый ответ должен начинаться с целого числа k — минимального количества слагаемых. Далее должны следовать k слагаемых, каждое из которых является круглым числом, а их сумма равна n . Слагаемые можно выводить в любом порядке. Если ответов несколько, то выведите любой из них.

Пример

входные данные	Скопировать
5 5009 7 9876 10000 10	
выходные данные	Скопировать
2 5000 9 1 7 4 800 70 6 9000 1 10000 1 10	

В. Сумма слагаемых одинаковой четности

ограничение по времени на тест: 1 секунда
 ограничение по памяти на тест: 256 мегабайт
 ввод: стандартный ввод
 вывод: стандартный вывод

Заданы два целых положительных числа n ($1 \leq n \leq 10^9$) и k ($1 \leq k \leq 100$). Представьте число n в виде суммы k положительных целых чисел одинаковой четности (с одинаковыми остатками от деления на 2).

Иными словами, найдите такие a_1, a_2, \dots, a_k , что все $a_i > 0$, $n = a_1 + a_2 + \dots + a_k$ и либо все a_i одновременно четные, либо все a_i одновременно нечетные. Если такого представления не существует, то сообщите об этом.

Входные данные

В первой строке записано целое число t ($1 \leq t \leq 1000$) — количество наборов входных данных в тесте. Далее записаны t наборов входных данных по одному в строке.

Каждый набор представляет собой два целых положительных числа n ($1 \leq n \leq 10^9$) и k ($1 \leq k \leq 100$).

Выходные данные

Для каждого набора входных данных выведите:

- YES и искомые a_i , если ответ существует (если ответов несколько, то выведите любой из них);
- NO, если ответа не существует.

Буквы в словах YES и NO можно выводить в любом регистре.

Пример

входные данные	Скопировать
8 10 3 100 4 8 7 97 2 8 8 3 10 5 3 100000000 9	
выходные данные	Скопировать
YES 4 2 4 YES	

```
55 5 5 35  
NO  
NO  
YES  
1 1 1 1 1 1 1 1  
NO  
YES  
3 1 1  
YES  
111111110 111111110 111111110 111111110 111111110 111111110 111111110 111111110 111111110 111111110
```

С. К-е неделящееся на n

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Заданы два целых положительных числа: n и k . Выведите k -е положительное целое число, которое не делится на n .

Например, если $n = 3$, а $k = 7$, то все числа, которые не делятся на 3, имеют вид: 1, 2, 4, 5, 7, 8, 10, 11, 13 . . . 7-е число среди них равно 10.

Входные данные

В первой строке записано целое число t ($1 \leq t \leq 1000$) — количество наборов входных данных в тесте. Далее записаны t наборов входных данных по одному в строке.

Каждый набор представляет собой два целых положительных числа n ($2 \leq n \leq 10^9$) и k ($1 \leq k \leq 10^9$).

Выходные данные

Для каждого набора входных данных выведите k -е целое положительное число, которое не делится на n .

Пример

входные данные	Скопировать
6 3 7 4 12 2 1000000000 7 97 1000000000 1000000000 2 1	
выходные данные	Скопировать
10 15 1999999999 113 1000000001 1	

D. Алиса, Боб и конфеты

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

В ряд выложены n конфет, которые пронумерованы слева направо от 1 до n . Размер i -й конфеты равен a_i .

Алиса и Боб играют в интересную и вкусную игру — они едят конфеты. Алиса будет есть конфеты **слева направо**, а Боб — **справа налево**. Игра заканчивается, когда все конфеты съедены.

Процесс состоит из ходов. Во время хода игрок съедает одну или более конфет со своей стороны (Алиса ест слева, Боб — справа). Первый ход делает Алиса. Во время первого хода она съест 1 конфету (ее размер равен a_1). Затем каждый следующий ход стороны чередуются — то есть второй ход совершает Боб, затем Алиса, затем снова Боб и так далее.

На каждом ходу игрок считает суммарный размер конфет, съеденных за текущий ход. Как только это число становится строго больше, чем суммарный размер конфет, съеденных другим игроком на предыдущем ходу, текущий игрок завершает ход. Иными словами, на очередном ходу игрок ест наименьшее возможное количество конфет, при котором сумма размеров съеденных в этот ход конфет **строго больше** суммы размеров конфет, которые съел другой игрок на **предыдущем** ходу. Если конфет недостаточно, чтобы совершить ход таким образом, то игрок доедает все оставшиеся конфеты и игра заканчивается.

Например, если $n = 11$ и $a = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5]$, то:

- ход 1: Алиса съест одну конфету размера 3 и последовательность конфет примет вид $[1, 4, 1, 5, 9, 2, 6, 5, 3, 5]$;
- ход 2: на предыдущем ходу Алиса съела 3, значит, Боб должен съесть 4 или более — Боб съест одну конфету размера 5 и последовательность конфет примет вид $[1, 4, 1, 5, 9, 2, 6, 5, 3]$;
- ход 3: на предыдущем ходу Боб съел 5, значит, Алиса должна съесть 6 или более — Алиса съест три конфеты суммарным размером $1 + 4 + 1 = 6$ и последовательность конфет примет вид $[5, 9, 2, 6, 5, 3]$;
- ход 4: на предыдущем ходу Алиса съела 6, значит, Боб должен съесть 7 или более — Боб съест две конфеты суммарным размером $3 + 5 = 8$ и последовательность конфет примет вид $[5, 9, 2, 6]$;
- ход 5: на предыдущем ходу Боб съел 8, значит, Алиса должна съесть 9 или более — на пятом ходу Алиса съест две конфеты суммарным размером $5 + 9 = 14$ и последовательность конфет примет вид $[2, 6]$;
- ход 6 (последний): на предыдущем ходу Алиса съела 14, значит, Боб должен съесть 15 или более — это невозможно, поэтому Боб съест две оставшиеся конфеты и игра закончится.

Выведите количество ходов в этой игре и два числа:

- a — суммарный размер всех конфет, съеденных Алисой за всю игру;
- b — суммарный размер всех конфет, съеденных Бобом за всю игру.

Входные данные

В первой строке записано целое число t ($1 \leq t \leq 5000$) — количество наборов входных данных. Далее следуют описания t наборов входных данных.

Каждый набор состоит из двух строк. В первой строке содержится целое число n ($1 \leq n \leq 1000$) — количество конфет. Во второй строке содержится последовательность целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 1000$) — размеры конфет в порядке их расположения слева направо.

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

Выходные данные
Для каждого набора входных данных выведите три целых числа — количество ходов в игре и искомые величины a и b .

Пример

входные данные	Скопировать
<pre>7 11 3 1 4 1 5 9 2 6 5 3 5 1 1000 3 1 1 1 13 1 2 3 4 5 6 7 8 9 10 11 12 13 2 2 1 6 1 1 1 1 1 1 7 1 1 1 1 1 1 1</pre>	
выходные данные	Скопировать
<pre>6 23 21 1 1000 0 2 1 2 6 45 46 2 2 1 3 4 2 4 4 3</pre>	

Е. Особые элементы

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 64 мегабайта
ввод: стандартный ввод
вывод: стандартный вывод

Обратите внимание на нестандартное ограничение по памяти в этой задаче.

С целью отсеечения эффективных решений от неэффективных в этой задаче ограничение времени довольно строгое. Предпочтите использование компилируемых статически типизированных языков (например, C++). Если используете Python, то отправляйте решения на PyPy. Постарайтесь написать в самом деле эффективное решение.

Задан массив $a = [a_1, a_2, \dots, a_n]$ ($1 \leq a_i \leq n$). Его элемент a_i называется особым, если существует такая пара индексов l и r ($1 \leq l < r \leq n$), что $a_i = a_l + a_{l+1} + \dots + a_r$. Иными словами, элемент называется особым, если он представим в виде суммы **двух или более подряд идущих элементов** массива (не важно, особых или нет).

Выведите количество особых элементов заданного массива a .

Например, если $n = 9$ и $a = [3, 1, 4, 1, 5, 9, 2, 6, 5]$, то ответ равен 5:

- $a_3 = 4$ — особый элемент, так как $a_3 = 4 = a_1 + a_2 = 3 + 1$;
- $a_5 = 5$ — особый элемент, так как $a_5 = 5 = a_2 + a_3 = 1 + 4$;
- $a_6 = 9$ — особый элемент, так как $a_6 = 9 = a_1 + a_2 + a_3 + a_4 = 3 + 1 + 4 + 1$;
- $a_8 = 6$ — особый элемент, так как $a_8 = 6 = a_2 + a_3 + a_4 = 1 + 4 + 1$;
- $a_9 = 5$ — особый элемент, так как $a_9 = 5 = a_2 + a_3 = 1 + 4$.

Обратите внимание, что среди элементов массива a могут быть равные — если несколько элементов равны и являются особыми, то все они должны быть посчитаны в ответе.

Входные данные

В первой строке записано целое число t ($1 \leq t \leq 1000$) — количество наборов входных данных в тесте. Далее следуют описания t наборов входных данных.

Каждый набор задается двумя строками. В первой строке записано целое число n ($1 \leq n \leq 8000$) — длина массива a . Во второй строке записаны целые числа a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$).

Гарантируется, что сумма значений n по всем наборам входных данных не превосходит 8000.

Выходные данные

Выведите t чисел — количества особых элементов для каждого из заданных массивов.

Пример

входные данные	Скопировать
<pre>5 9 3 1 4 1 5 9 2 6 5 3 1 1 2 5 1 1 1 1 1 8 8 7 6 5 4 3 2 1 1 1</pre>	
выходные данные	Скопировать
<pre>5 1 0 4 0</pre>	

Ф. Восстановление 01-строки

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Для некоторой 01-строки s (то есть такой, что каждый символ s_i — это либо '0', либо '1') выписали все пары подряд идущих символов (все подстроки длины 2). Для каждой пары (подстроки длины 2) посчитали количество символов '1' (единиц) в ней.

Вам заданы три числа:

- n_0 — количество пар подряд идущих символов (подстрок), в которых 0 единиц;
- n_1 — количество пар подряд идущих символов (подстрок), в которых 1 единица;
- n_2 — количество пар подряд идущих символов (подстрок), в которых 2 единицы.

Например, для строки $s = \text{«1110011110»}$ были бы выписаны следующие подстроки: «11», «11», «10», «00», «01», «11», «11», «11», «10». Таким образом, $n_0 = 1, n_1 = 3, n_2 = 5$.

Ваша задача — по заданным n_0, n_1, n_2 восстановить **любую** подходящую 01-строку (бинарную строку) s . Гарантируется, что хотя бы одно из чисел n_0, n_1, n_2 отлично от 0. Также гарантируется, что решение существует.

Входные данные

В первой строке записано целое число t ($1 \leq t \leq 1000$) — количество наборов входных данных в тесте. Далее во входных данных записаны сами наборы.

Каждый набор состоит из одной строки, которая содержит три целых числа n_0, n_1, n_2 ($0 \leq n_0, n_1, n_2 \leq 100; n_0 + n_1 + n_2 > 0$). Гарантируется, что ответ для заданных n_0, n_1, n_2 существует.

Выходные данные

Выведите t строк. Каждая из строк должна содержать ответ в виде 01-строки (бинарной строки) на соответствующий набор входных данных. Если возможных ответов несколько, выведите любой из них.

Пример

входные данные	Скопировать
7 1 3 5 1 1 1 3 9 3 0 1 0 3 1 2 0 0 3 2 0 0	
выходные данные	Скопировать
1110011110 0011 0110001100101011 10 0000111 1111 000	

G. Особая перестановка

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Перестановкой длины n называется такой массив $p = [p_1, p_2, \dots, p_n]$, который содержит каждое число от 1 до n (включительно) и притом ровно по одному разу. Например, $p = [3, 1, 4, 2, 5]$ — перестановка длины 5.

Для заданного числа n ($n \geq 2$) найдите такую перестановку p , в которой разница (то есть модуль разности) любых двух соседних элементов находится в диапазоне от 2 до 4, включительно. Формально, для перестановки p должно выполняться $2 \leq |p_i - p_{i+1}| \leq 4$ для всех i ($1 \leq i < n$).

Выведите любую такую перестановку для заданного значения n или определите, что ее не существует.

Входные данные

В первой строке записано целое число t ($1 \leq t \leq 100$) — количество наборов входных данных в тесте. Далее следуют описания t наборов входных данных.

Каждый набор описывается единственной строкой, в которой содержится целое число n ($2 \leq n \leq 1000$).

Выходные данные

Выведите t строк. Очередная строка должна содержать -1 , если для соответствующего набора входных данных искомой перестановки не существует. В противном случае выведите искомую перестановку. Если таких перестановок существует несколько, то выведите любую из них.

Пример

входные данные	Скопировать
6 10 2 4 6 7 13	
выходные данные	Скопировать
9 6 10 8 4 7 3 1 5 2 -1 3 1 4 2 5 3 6 2 4 1 5 1 3 6 2 4 7 13 9 7 11 8 4 1 3 5 2 6 10 12	