

В. Настя и табло

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

Денис, справившийся с покупкой цветов и конфет (об этой истории вы узнаете в следующей задаче), поехал на встречу с Настей, чтобы предложить ей стать парой.

И вот, они сидят вместе в кафе, болтают. Решительный Денис наконец-то предлагает быть им вместе, но... Но Настя не даёт никакого ответа.

Как же сильно расстроился бедный парень. Так сильно, что он пнул какое-то табло с цифрами. Цифры отображаются таким же образом, как на электронных часах: каждая позиция для цифры состоит из 7 сегментов, которые могут быть включены или выключены, чтобы отображать различные цифры. На картинке показано, как изображаются все 10 десятичных цифр:



После пинка некоторые палочки перестали работать, то есть некоторые палочки могли перестать гореть, если горели раньше. Но Денис запомнил, сколько палочек горело и сколько горит сейчас. Пусть сломалось **ровно**  $k$  палочек и известно, какие палочки горят сейчас. Денис задался вопросом: какое максимальное число может гореть на табло, если включить ровно  $k$  палочек (из тех которые сейчас выключены)?

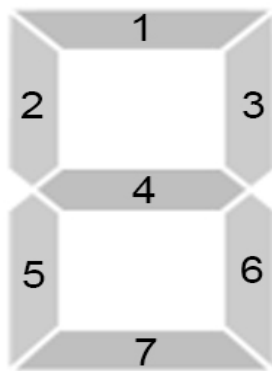
**Разрешается**, чтобы в числе были лидирующие нули.

**Входные данные**

В первой строке дано число  $n$  ( $1 \leq n \leq 2000$ ) — количество цифр на табло и  $k$  ( $0 \leq k \leq 2000$ ) — количество сломанных палочек.

В следующих  $n$  строках находится по одной бинарной строке длины 7,  $i$ -я из которых кодирует  $i$ -ю цифру табло.

Каждая цифра на табло состоит из 7 палочек. Пронумеруем их, как на картинке ниже, и пусть на  $i$ -м месте бинарной строки будет 0, если  $i$ -я палочка не горит и 1, если горит. Тогда бинарная строка длины 7 будет задавать то, какие палочки горят.



Таким образом, последовательности «1110111», «0010010», «1011101», «1011011», «0111010», «1101011», «1101111», «1010010», «1111111», «1111011» кодируют по порядку все цифры от 0 до 9 включительно.

**Выходные данные**

Выведите единственное число, состоящее из  $n$  разрядов — максимальное число, которое можно получить, если включить ровно  $k$  палочек или  $-1$ , если невозможно так включить ровно  $k$  палочек, чтобы на табло получилась какая-то последовательность цифр.

<b>Примеры</b>	
<b>входные данные</b>	<a href="#">Скопировать</a>
<pre>1 7 0000000</pre>	
<b>выходные данные</b>	<a href="#">Скопировать</a>

8

<b>входные данные</b>	<a href="#">Скопировать</a>
2 5 0010010 0010010	
<b>выходные данные</b>	<a href="#">Скопировать</a>
97	

<b>входные данные</b>	<a href="#">Скопировать</a>
3 5 0100001 1001001 1010011	
<b>выходные данные</b>	<a href="#">Скопировать</a>
-1	

**Примечание**  
В первом тесте мы обязаны включить все 7 палочек и получить на табло одну цифру 8.

Во втором тесте у нас включены палочки таким образом, что образуются единицы. За 5 дополнительно включённых палочек можно получить числа 07, 18, 34, 43, 70, 79, 81 и 97, из них выбираем максимальное — 97.

В третьем тесте невозможно так включить ровно 5 палочек, чтобы на табло получилась какая-то последовательность цифр.



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

## Н. Не вините меня

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

К сожалению, автор задачи не смог придумать интересную историю, поэтому он просто просит вас решить следующую задачу.

Дан массив  $a$ , состоящий из  $n$  положительных целых чисел. Посчитайте количество подпоследовательностей, для которых побитовое **AND** элементов в подпоследовательности имеет ровно  $k$  единичных битов в двоичном представлении. Ответ может быть большим, поэтому выведите его по модулю  $10^9 + 7$ .

Напомним, что подпоследовательность массива  $a$  - это последовательность, которую можно получить из  $a$ , удалив некоторые (возможно, ни одного) элементы. Например,  $[1, 2, 3]$ ,  $[3]$ ,  $[1, 3]$  являются подпоследовательностями  $[1, 2, 3]$ , но  $[3, 2]$  и  $[4, 5, 6]$  - нет.

Обратите внимание, что **AND** обозначает [логическую операцию AND](#).

### Входные данные

Каждый тест содержит несколько наборов входных данных. Первая строка содержит количество наборов  $t$  ( $1 \leq t \leq 10^4$ ). Затем следует описание наборов входных данных.

Первая строка каждого тестового случая состоит из двух целых чисел  $n$  и  $k$  ( $1 \leq n \leq 2 \cdot 10^5$ ,  $0 \leq k \leq 6$ ) — длина массива и количество единичных битов, которое должно быть у побитового **AND** подсчитанных подпоследовательностей в двоичном представлении.

Вторая строка каждого тестового случая состоит из  $n$  целых чисел  $a_i$  ( $0 \leq a_i \leq 63$ ) — массив  $a$ .

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

### Выходные данные

Для каждого тестового случая выведите одно целое число - количество подпоследовательностей, у которых в двоичном представлении значение побитового **AND** имеет ровно  $k$  установленных битов. Ответ может быть большим, поэтому выведите его по модулю  $10^9 + 7$ .

### Пример

входные данные	Скопировать
6 5 1 1 1 1 1 1 4 0 0 1 2 3 5 1 5 5 7 4 2 1 2 3 12 0 0 2 0 2 0 2 0 2 0 2 10 6 63 0 63 5 5 63 63 4 12 13	
выходные данные	Скопировать
31 10 10 1 4032 15	

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

## D. Борис и его восхитительная прическа

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Борис думает, что шахматы это скучная игра, поэтому он ушел со своего турнира пораньше. Он пошел в парикмахерскую, ведь его прическа была *немного* неаккуратная.

В настоящий момент его волосы можно описать массивом  $a_1, a_2, \dots, a_n$ , где  $a_i$  — высота волоса на позиции  $i$ . Его желаемая прическа описывается массивом  $b_1, b_2, \dots, b_n$  аналогично.

У парикмахера есть  $m$  бритв. Каждая бритва имеет некоторый размер и может быть использована **не более** одного раза. За одну операцию парикмахер может выбрать одну бритву и обрезать с помощью нее отрезок волос Бориса. Формально, операция выглядит следующим образом:

- Выбрать ранее неиспользованную бритву, пусть ее размер равен  $x$ ;
- Выбрать отрезок  $[l, r]$  ( $1 \leq l \leq r \leq n$ );
- Присвоить  $a_i := \min(a_i, x)$  для всех  $l \leq i \leq r$ .

Обратите внимание, что некоторые бритвы могут иметь одинаковый размер. Парикмахер может выбирать размер  $x$  максимум столько раз, сколько у него есть бритв размера  $x$ .

Парикмахер может выполнить сколько угодно операций, не используя одну бритву дважды. Необходимо, чтобы в конце выполнялось  $a_i = b_i$  для всех  $1 \leq i \leq n$ . Он **не обязан** использовать все бритвы.

Определите, может ли парикмахер сделать необходимую Борису стрижку.

### Входные данные

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число  $t$  ( $1 \leq t \leq 20\,000$ ) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит положительное целое число  $n$  ( $3 \leq n \leq 2 \cdot 10^5$ ) — длину массивов  $a$  и  $b$ .

Вторая строка содержит  $n$  положительных целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ) — описание текущих волос Бориса.

Третья строка содержит  $n$  положительных целых чисел  $b_1, b_2, \dots, b_n$  ( $1 \leq b_i \leq 10^9$ ) — описание необходимой стрижки.

Четвертая строка содержит положительное целое число  $m$  ( $1 \leq m \leq 2 \cdot 10^5$ ) — количество бритв.

Пятая строка содержит  $m$  положительных целых чисел  $x_1, x_2, \dots, x_m$  ( $1 \leq x_i \leq 10^9$ ) — размеры бритв.

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

### Выходные данные

Для каждого набора входных данных выведите «YES», если парикмахер может выполнить необходимую стрижку, и «NO» иначе.

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 3 3 3 3 2 1 2 2 1 2 6 3 4 4 6 3 4 3 1 2 3 2 3 3 3 2 3 10 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 10 10 1 2 3 4 5 6 7 8 9 10 </pre>	

3  
1 1 1  
1 1 2  
12  
4 2 4 3 1 5 6 3 5 6 2 1  
13  
7 9 4 5 3 3 3 6 8 10 3 2 5  
5 3 1 5 3 2 2 5 8 5 1 1 5  
8  
1 5 3 5 4 2 3 1  
13  
7 9 4 5 3 3 3 6 8 10 3 2 5  
5 3 1 5 3 2 2 5 8 5 1 1 5  
7  
1 5 3 4 2 3 1  
3  
19747843 2736467 938578397  
2039844 2039844 2039844  
1  
2039844

выходные данные

Скопировать

YES  
NO  
YES  
NO  
YES  
NO  
YES

Примечание

В первом примере волосы Бориса изначально равны  $[3, 3, 3]$ . Опишем последовательность из 2 операций, которые может выполнить парикмахер:

- Использовать бритву размера 1 на отрезке  $[2, 2]$ ; волосы становятся  $[3, 1, 3]$ .
- Использовать бритву размера 2 на отрезке  $[1, 3]$ ; волосы становятся  $[2, 1, 2]$ , что и требуется.

В третьем примере можно ничего не делать, так как волосы уже в желаемом состоянии.

В четвертом примере нельзя обрезать волосы так, чтобы их длина увеличилась, и  $[1, 1, 1]$  превратился в  $[1, 1, 2]$ .



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

## С. Рекора и батуты

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Есть парк батутов с  $n$  батутами, расположенными в ряд.  $i$ -й из них имеет силу  $S_i$ .

Рекора может прыгать по батутам в несколько проходов. Она начинает проход, прыгая на любой батут по своему выбору.

Если сейчас Рекора прыгает на батут  $i$ , то батут подбросит ее в позицию  $i + S_i$ , а  $S_i$  станет равно  $\max(S_i - 1, 1)$ . Иначе говоря,  $S_i$  уменьшится на 1, кроме случая  $S_i = 1$ , когда  $S_i$  останется равно 1.

Если в позиции  $i + S_i$  нет батута, то этот проход завершен. В противном случае Рекора продолжит проход прыжком с батута в позиции  $i + S_i$  по принципу, описанному выше.

**Рекора не может перестать прыгать во время прохода, пока не приземлится на позицию больше  $n$  (в которой нет батута).** Бедная Рекора!

Рекора — непослушный кролик, и хочет разрушить парк батуты, уменьшив все  $S_i$  до 1. Какое минимальное количество проходов ей нужно, чтобы уменьшить все  $S_i$  до 1?

### Входные данные

В первой строке содержится одно целое число  $t$  ( $1 \leq t \leq 500$ ) — количество наборов входных данных.

Первая строка каждого набора входных данных содержит одно целое число  $n$  ( $1 \leq n \leq 5000$ ) — количество батуты.

Вторая строка каждого набора входных данных содержит  $n$  целых чисел  $S_1, S_2, \dots, S_n$  ( $1 \leq S_i \leq 10^9$ ), где  $S_i$  — это сила  $i$ -го батута.

Гарантируется, что сумма  $n$  по всем наборам входных данных не превышает 5000.

### Выходные данные

Для каждого набора входных данных выведите одно целое число — минимальное количество проходов, за которое Рекора может уменьшить все  $S_i$  до 1.

### Пример

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

### Примечание

Для первого набора входных данных, ниже приведена оптимальная последовательность проходов, которые Рекора может использовать. (Жирным выделены позиции батуты, куда прыгает Рекора в очередной проход).

- [1, 4, **2**, 2, **2**, 2, **2**]
- [1, **4**, 1, 2, 1, **2**, 1]
- [1, **3**, 1, 2, **1**, **1**, 1]
- [1, **2**, 1, **2**, 1, **1**, 1]

Для второго набора входных данных оптимальная последовательность проходов приведена ниже.

- [**2**, 3]
- [1, **3]**
- [1, **2]**

Для третьего набора входных данных, все  $S_i$  уже равны 1.

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

C. Торг

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

Порой сложно прийти к конечной цене в торге. Вот и Саша с Вовой никак не могут прийти к соглашению – Саша стремится назвать цену как можно больше, а Вова — вычеркнуть из нее как можно больше цифр. Т. е. получается так, что Саша называет какое-то целое число  $n$ , Вова вычеркивает из него непустую последовательность подряд идущих цифр, а оставшиеся цифры схлопываются в одно число.

Например, если из числа 1213121 вычеркнуть подстроку 1312, то получится число 121.

Допустима, что итоговая цена содержит ведущие нули. Если Вова вычеркнет все цифры, то считается, что получился 0.

Саша хочет как-то ограничить Вову, чтобы тот не вычеркнул все число, но, чтобы у него были правильные аргументы для убеждения, ему нужно знать сумму всех чисел, которые могут получиться после того, как Вова сделает вычеркивание.

Помогите Саше посчитать. А так как число может получиться достаточно большим, то считайте его по модулю  $10^9 + 7$ .

Входные данные

В первой и единственной строке задается целое число  $n$  ( $1 \leq n < 10^{10^5}$ ).

Выходные данные

В единственной строке выведите искомую сумму по модулю  $10^9 + 7$ .

Примеры

входные данные	Скопировать
107	
выходные данные	Скопировать
42	

входные данные	Скопировать
100500100500	
выходные данные	Скопировать
428101984	

Примечание

Рассмотрим первый пример.

Вова может вычеркнуть 1, 0, 7, 10, 07 или 107. Получившиеся числа равны 07, 17, 10, 7, 1, 0. Их сумма равна 42.



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

### С. Покраска деревьев

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

Кодер ZS и Бабуин Крис прибыли в Удайлэнд! Они прогуливались по парку, в котором растут  $n$  деревьев. Друзья решили пошалить и покрасить деревья в парке. Деревья пронумерованы целыми числами от 1 до  $n$  слева направо.

Изначально, дерево  $i$  покрашено в цвет  $c_i$ . Кодер ZS и Бабуин Крис знают только  $m$  цветов, поэтому  $0 \leq c_i \leq m$ , где  $c_i = 0$  означает, что дерево  $i$  еще не покрашено.

Кодер ZS и бабуин Крис решили красить только не покрашенные деревья, то есть деревья с  $c_i = 0$ . Они могут покрасить каждое из них в один из  $m$  цветов от 1 до  $m$ . Покраска  $i$ -ого дерева в цвет  $j$  требует  $p_{i,j}$  литров краски.

Друзья определяют *красоту* покраски деревьев как **наименьшее** количество групп последовательных деревьев (каждая группа содержит некоторый подотрезок деревьев), на которые можно разбить все  $n$  деревьев так, что каждая из групп содержит деревья одного цвета. К примеру, если деревья слева направо раскрашены в цвета 2, 1, 1, 1, 3, 2, 2, 3, 1, 3, красота покраски равна 7, так как можно разбить деревья на 7 групп последовательных деревьев одного цвета:  $\{2\}, \{1, 1, 1\}, \{3\}, \{2, 2\}, \{3\}, \{1\}, \{3\}$ .

Кодер ZS и Бабуин Крис хотят покрасить все не покрашенные деревья так, что красота покраски будет **в точности** равняться  $k$ . Друзьям нужна ваша помощь в определении минимального количества краски (в литрах), которая им понадобится для свершения задуманного.

Заметьте, что друзья не могут красить уже покрашенные деревья.

#### Входные данные

Первая строка содержит три целых числа  $n, m$  и  $k$  ( $1 \leq k \leq n \leq 100, 1 \leq m \leq 100$ ) — количество деревьев, количество цветов и красоту результата покраски соответственно.

Вторая строка содержит  $n$  целых чисел  $c_1, c_2, \dots, c_n$  ( $0 \leq c_i \leq m$ ) — исходные цвета деревьев.  $c_i$  равно 0, если дерево номер  $i$  не покрашено, в противном случае цвет  $i$ -го дерева равен  $c_i$ .

Далее следуют  $n$  строк. Каждая из них содержит  $m$  целых чисел.  $j$ -ое число в  $i$ -ой из них равно  $p_{i,j}$  ( $1 \leq p_{i,j} \leq 10^9$ ) — количество литров краски, необходимое для того, чтобы покрасить  $i$ -ое дерево в цвет  $j$ .  $p_{i,j}$  определены и для изначально покрашенных деревьев, но эти деревья все равно не могут быть покрашены.

#### Выходные данные

Выведите единственное целое число — минимальное количество краски, необходимое для покраски деревьев. Если корректной покраски деревьев красоты  $k$  не существует, выведите - 1.

#### Примеры

<b>входные данные</b>	Скопировать
3 2 2 0 0 0 1 2 3 4 5 6	
<b>выходные данные</b>	Скопировать
10	
<b>входные данные</b>	Скопировать
3 2 2 2 1 2 1 3 2 4 3 5	
<b>выходные данные</b>	Скопировать
-1	
<b>входные данные</b>	Скопировать



3 2 2  
2 0 0  
1 3  
2 4  
3 5

выходные данные

Скопировать

5

входные данные

Скопировать

3 2 3  
2 1 2  
1 3  
2 4  
3 5

выходные данные

Скопировать

0

Примечание

В первом примере из условия покраска деревьев в цвета 2, 1, 1 минимизирует количество используемой краски, равное  $2 + 3 + 5 = 10$ . Заметьте, что 1, 1, 1 не является корректной покраской, так как ее красота равна 1 ( $\{1, 1, 1\}$  является способом разбиения деревьев на одну группу деревьев одного цвета).

Во втором примере из условия все деревья покрашены, но красота покраски равна 3, поэтому правильной покраски не существует и ответ равен - 1.

В последнем примере из условия все деревья покрашены и красота покраски равна  $k$ , поэтому краска не используется и ответ равен 0.

[Codeforces](#) (с) Copyright 2010-2024 Михаил Мирзаянов  
Соревнования по программированию 2.0  
Время на сервере: 29.11.2024 22:01:32<sup>UTC+5</sup> (k1).  
Мобильная версия, переключиться на [десктопную](#).  
[Privacy Policy](#)

При поддержке



ІІТМО

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

С. Лесенка

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

Вам задан массив, состоящий из  $n$  целых чисел  $a_1, a_2, \dots, a_n$ . Также задано  $m$  запросов,  $i$ -ый запрос описывается двумя целыми числами  $l_i, r_i$ . Числа  $l_i, r_i$  обозначают подотрезок исходного массива, то есть последовательность чисел  $a_{l_i}, a_{l_i+1}, a_{l_i+2}, \dots, a_{r_i}$ . Для каждого запроса нужно проверить, является ли соответствующий ему подотрезок лесенкой.

Лесенкой назовем последовательность целых чисел  $b_1, b_2, \dots, b_k$  такую, что сначала она не убывает, а затем не возрастает. Другими словами, существует такое целое число  $x$  ( $1 \leq x \leq k$ ), что выполняется неравенство:  $b_1 \leq b_2 \leq \dots \leq b_x \geq b_{x+1} \geq b_{x+2} \geq \dots \geq b_k$ . Обратите внимание, что неубывающая и невозрастающая последовательности также считаются лесенками.

Входные данные

В первой строке заданы два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 10^5$ ) — количество элементов массива и количество запросов соответственно. Во второй строке задана последовательность целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ), в которой число  $a_i$  обозначает  $i$ -ый элемент массива.

Далее в  $m$  строках заданы описания запросов. В  $i$ -ой строке задано описание  $i$ -ого запроса, состоящее из двух целых чисел  $l_i, r_i$  ( $1 \leq l_i \leq r_i \leq n$ ) — границы подотрезка исходного массива.

Числа в строках разделяются одиночными пробелами.

Выходные данные

Выведите  $m$  строк, в  $i$ -ой строке выведите слово «Yes» (без кавычек), если подотрезок, соответствующий  $i$ -ому запросу, является лесенкой, или слово «No» (без кавычек) в противном случае.

Примеры

<b>входные данные</b>	<a href="#">Скопировать</a>
<pre>8 6 1 2 1 3 3 5 2 1 1 3 2 3 2 4 8 8 1 4 5 8</pre>	
<b>выходные данные</b>	<a href="#">Скопировать</a>
<pre>Yes Yes No Yes No No Yes</pre>	



Г. Короткая задача

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 512 мегабайт

Обозначим за  $d(n)$  сумму всех делителей числа  $n$ , т.е.  $d(n) = \sum_{k|n} k$ .

Например,  $d(1) = 1$ ,  $d(4) = 1 + 2 + 4 = 7$ ,  $d(6) = 1 + 2 + 3 + 6 = 12$ .

Для заданного числа  $c$ , найдите минимальное  $n$  такое, что  $d(n) = c$ .

Входные данные

В первой строке содержится одно целое число  $t$  ( $1 \leq t \leq 10^4$ ). Далее следуют  $t$  наборов входных данных.

Каждый набор входных данных характеризуется одним целым числом  $c$  ( $1 \leq c \leq 10^7$ ).

Выходные данные

Для каждого набора входных данных выведите:

- «-1», если не существует  $n$ , такого что  $d(n) = c$ ;
- $n$ , иначе.

Пример

входные данные	Скопировать
12 1 2 3 4 5 6 7 8 9 10 39 691	
выходные данные	Скопировать
1 -1 2 3 -1 5 4 7 -1 -1 18 -1	



С. Минимизация суммы

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

Задан целочисленный массив  $a$  размера  $n$ .

Вы можете выполнить следующую операцию: выбрать элемент массива и заменить его значением любого из его соседей.

Например, если  $a = [3, 1, 2]$ , вы можете получить один из массивов  $[3, 3, 2]$ ,  $[3, 2, 2]$  и  $[1, 1, 2]$  за одну операцию, но не  $[2, 1, 2]$  и  $[3, 4, 2]$ .

Ваша задача — посчитать минимально возможную сумму массива, если вы можете выполнить вышеописанную операцию не более  $k$  раз.

**Входные данные**

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

Первая строка каждого набора содержит два целых числа  $n$  и  $k$  ( $1 \leq n \leq 3 \cdot 10^5$ ;  $0 \leq k \leq 10$ ).

Вторая строка содержит  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ).

Дополнительное ограничение на входные данные: сумма  $n$  по всем наборам входных данных не превышает  $3 \cdot 10^5$ .

**Выходные данные**

Для каждого набора входных данных выведите одно целое число — минимально возможную сумму массива, если вы можете выполнить вышеописанную операцию не более  $k$  раз.

Пример

<b>входные данные</b>	Скопировать
4 3 1 3 1 2 1 3 5 4 2 2 2 1 3 6 3 4 1 2 2 4 3	
<b>выходные данные</b>	Скопировать
4 5 5 10	

**Примечание**

В первом примере одна из возможных последовательностей операций:  $[3, 1, 2] \rightarrow [1, 1, 2]$ .

Во втором примере не нужно применять операцию.

В третьем примере одна из возможных последовательностей операций:  $[2, 2, 1, 3] \rightarrow [2, 1, 1, 3] \rightarrow [2, 1, 1, 1]$ .

В четвертом примере одна из возможных последовательностей операций:  
 $[4, 1, 2, 2, 4, 3] \rightarrow [1, 1, 2, 2, 4, 3] \rightarrow [1, 1, 1, 2, 4, 3] \rightarrow [1, 1, 1, 2, 2, 3]$ .

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

### D. Омкар и Война кроватей

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

Омкар играет в свою любимую видеоигру Война кроватей! В Войне кроватей  $n$  игроков располагаются по кругу, так что для всех  $j$  таких, что  $2 \leq j \leq n$ , игрок  $j - 1$  находится слева от игрока  $j$ , а игрок  $j$  — справа от игрока  $j - 1$ . Кроме того, слева от игрока 1 находится игрок  $n$ , а справа от игрока  $n$  — игрок 1.

В настоящее время каждый игрок атакует либо левого, либо правого игрока. Это означает, что каждого игрока в настоящее время атакует либо 0, 1 либо 2 других игроков. Ключевым элементом стратегии Войны кроватей является то, что если на игрока нападает ровно 1 другой игрок, то в ответ он, логически, должен атаковать этого игрока. Если же на игрока нападают либо 0 либо 2 других игроков, то стратегия Войны кроватей гласит, что игрок может атаковать любого из соседних игроков.

К сожалению, может случиться так, что некоторые игроки в этой игре не следуют правильной стратегии Войны кроватей нужным образом. Омкар знает, на кого в настоящее время нападает каждый игрок, и он может поговорить с любым количеством из  $n$  игроков в игре, чтобы заставить их вместо этого атаковать другого игрока — то есть, если игрок в настоящее время атаковал игрока слева от него, Омкар может убедить его вместо этого атаковать игрока справа от него; если он в настоящее время атаковал игрока справа от него, Омкар может убедить его вместо этого атаковать игрока слева от него.

Омкар хотел бы, чтобы все игроки действовали логично. Найдите минимальное количество игроков, с которыми Омкар должен поговорить, чтобы после того, как все игроки, с которыми он поговорил (если таковые имеются), изменили игрока, которого они атакуют, все игроки действовали логически в соответствии со стратегией Войны кроватей.

#### Входные данные

Каждый тест содержит несколько наборов входных данных. В первой строке указано количество наборов входных данных  $t$  ( $1 \leq t \leq 10^4$ ). Ниже приведены описания наборов входных данных.

Первая строка каждого набора входных данных содержит одно целое число  $n$  ( $3 \leq n \leq 2 \cdot 10^5$ ) — количество игроков (а значит и кроватей) в этой игре Bed Wars.

Вторая строка каждого набора входных данных содержит строку  $s$  длиной  $n$ .  $j$ -й символ  $s$  равен L, если  $j$ -й игрок атакует игрока слева от него, и R, если  $j$ -й игрок атакует игрока справа от него.

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

#### Выходные данные

Для каждого набора входных данных выведите одно целое число: минимальное количество игроков, с которыми Омкар должен поговорить, чтобы сделать так, чтобы все игроки действовали логично в соответствии со стратегией Войны кроватей.

Можно доказать, что Омкар всегда может достичь этого при заданных ограничениях.

#### Пример

входные данные	Скопировать
5 4 RLRL 6 LRRRRL 8 RLLRRLL 12 LLLLRRLRRLL 5 RRRRR	
выходные данные	Скопировать
0 1 1 3 2	

#### Примечание

В первом наборе входных данных игроки 1 и 2 атакуют друг друга, а игроки 3 и 4 — друг друга. Каждый игрок атакует ровно 1 другого игрока, и каждый игрок атакует игрока, который атакует их, так что все игроки уже действуют логично в соответствии со

стратегией Войны кроватей, и Омкару не нужно разговаривать ни с одним из них, следовательно, ответ 0.

Во втором наборе входных данных, не каждый игрок действует логически: например, игрока 3 атакует только игрок 2, но он не атакует его в ответ. Омкару достаточно поговорить с игроком 3 для изменения схемы атаки на LRLRRL, в которой все игроки действуют логично в соответствии со стратегией Войны кроватей, делая ответ 1.

---

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов  
Соревнования по программированию 2.0  
Время на сервере: 29.11.2024 22:01:13<sup>UTC+5</sup> (k1).  
Мобильная версия, переключиться на [десктопную](#).  
[Privacy Policy](#)

При поддержке



С. Штрихкод

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

Имеется картинка размера  $n \times m$  пикселей. Каждый пиксел может быть белым или черным. Требуется изменить цвета как можно меньшего количества пикселей так, чтобы получилась картинка-штрихкод.

Картинка является штрихкодом если выполняются следующие условия:

- В каждом столбце все пиксеты одного цвета.
- Ширина каждой одноцветной вертикальной полосы не менее  $x$  и не более  $y$  пикселей. Другими словами, если сгруппировать все соседние столбцы пикселей одного цвета, то не должно получиться группы размера менее  $x$  или более  $y$ .

Входные данные

В первой строке записаны четыре целых числа через пробел  $n, m, x$  и  $y$  ( $1 \leq n, m, x, y \leq 1000; x \leq y$ ).

Далее идет  $n$  строк, описывающих исходную картинку. В каждой из этих строк содержится ровно  $m$  символов. Символ «. » обозначает белый пиксел, а «#» — черный. Никаких других символов кроме «. » и «#» в описании картинки не содержится.

Выходные данные

В первой строке выведите наименьшее количество пикселей, которое нужно перекрасить. Гарантируется, что ответ существует.

Примеры

входные данные	Скопировать
6 5 1 2 ##.##. .###. ###.. #...# .##.# ###..	
выходные данные	Скопировать
11	

входные данные	Скопировать
2 5 1 1 ##### .....	
выходные данные	Скопировать
5	

Примечание

В первом тестовом примере картинка после перекрашивания может выглядеть следующим образом:

##. .  
##. .  
##. .  
##. .  
##. .  
##. .

Во втором тестовом примере картинка после перекрашивания может выглядеть следующим образом:

.#.#.  
.#.#.

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

D. Программа

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

Задана программа, состоящая из  $n$  инструкций. Изначально единственная переменная  $x$  присваивается 0. После этого следуют инструкции двух типов:

- увеличить  $x$  на 1;
- уменьшить  $x$  на 1.

Даны  $m$  запросов следующего типа:

- запрос  $l\ r$  — сколько различных значений принимает переменная  $x$ , если все инструкции между  $l$ -й и  $r$ -й включительно пропускаются, а остальные исполняются без изменения порядка?

Входные данные

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

Затем следует описание  $t$  наборов входных данных.

В первой строке каждого набора входных данных записаны два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 2 \cdot 10^5$ ) — количество инструкций в программе и количество запросов.

Во второй строке каждого набора входных данных записана программа — последовательность из  $n$  символов: каждый символ равен либо '+', либо '-' — инструкция увеличения и уменьшения, соответственно.

В каждой из следующих  $m$  строк записаны по два целых числа  $l$  и  $r$  ( $1 \leq l \leq r \leq n$ ) — описание запроса.

Сумма  $n$  по всем наборам входных данных не превосходит  $2 \cdot 10^5$ . Сумма  $m$  по всем наборам входных данных не превосходит  $2 \cdot 10^5$ .

Выходные данные

На каждый набор входных данных выведите  $m$  целых чисел — для каждого запроса  $l, r$  выведите количество различных значений, которые принимает переменная  $x$ , если все инструкции между  $l$ -й и  $r$ -й включительно пропускаются, а остальные исполняются без изменения порядка.

Пример

входные данные	Скопировать
2 8 4 -+---++ 1 8 2 8 2 5 1 1 4 10 +-++ 1 1 1 2 2 2 1 3 2 3 3 3 1 4 2 4 3 4 4 4	
выходные данные	Скопировать
1 2 4 4 3 3 4 2 3	



2  
1  
2  
2  
2

**Примечание**

Инструкции, которые остаются в каждом запросе первого набора входных данных:

1. пустая программа —  $x$  был равен только 0;
2. «-» —  $x$  принимал значения 0 и -1;
3. «- - - +» —  $x$  принимал значения 0, -1, -2, -3, -2 — среди них 4 различных значения;
4. «+ - - + - - +» — различные значения — это 1, 0, -1, -2.

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов  
Соревнования по программированию 2.0  
Время на сервере: 29.11.2024 22:01:15<sup>UTC+5</sup> (k1).  
Мобильная версия, переключиться на [десктопную](#).  
[Privacy Policy](#)

При поддержке

**ІТМО**

D. Running Miles

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

Есть улица с  $n$  достопримечательностями, достопримечательность с номером  $i$  находится в  $i$  милях от начала улицы. Достопримечательность номер  $i$  обладает красотой  $b_i$ . Вы хотите стартовать утреннюю пробежку в  $l$  милях и закончить в  $r$  милях от начала улицы. Пока вы бежите, вы будете пробегать мимо каких-то достопримечательностей (включая достопримечательности в  $l$  и  $r$  милях от начала). Вам интересны 3 наиболее красивых достопримечательности с вашей пробежки, но вы устаете с каждой милей, которую пробегаете.

Выберите  $l$  и  $r$  такие, что вы пробежите мимо хотя бы 3 достопримечательностей, и сумма красот 3 самых красивых достопримечательностей минус дистанция в милях, которую вы пробегаете, максимальна. Более формально, выберите  $l$  и  $r$  такие, что  $b_{i_1} + b_{i_2} + b_{i_3} - (r - l)$  максимально, где  $i_1, i_2, i_3$  — индексы трех самых красивых достопримечательностей в диапазоне  $[l, r]$ .

Входные данные

В первой строке находится единственное целое число  $t$  ( $1 \leq t \leq 10^5$ ) — число наборов входных данных.

Первая строка каждого набора входных данных содержит единственное целое число  $n$  ( $3 \leq n \leq 10^5$ ).

Вторая строка каждого набора входных данных содержит  $n$  целых чисел  $b_i$  ( $1 \leq b_i \leq 10^8$ ) — красоты достопримечательностей в  $i$  милях от начала улицы.

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

Выходные данные

Для каждого набора входных данных выведите одно целое число — максимальное значение  $b_{i_1} + b_{i_2} + b_{i_3} - (r - l)$  для некоторого отрезка  $[l, r]$ .

Пример

входные данные	Скопировать
4 5 5 1 4 2 3 4 1 1 1 1 6 9 8 7 6 5 4 7 100000000 1 100000000 1 100000000 1 100000000	
выходные данные	Скопировать
8 1 22 299999996	

Примечание

В первом примере мы можем выбрать  $l$  и  $r$  равными 1 и 5. Так мы посетим все достопримечательности, и три достопримечательности с максимальной красотой имеют индексы 1, 3 и 5 с красотоми 5, 4 и 3 соответственно. Откуда суммарное значение равно  $5 + 4 + 3 - (5 - 1) = 8$ .

Во втором примере отрезок  $[l, r]$  может быть равен  $[1, 3]$  или  $[2, 4]$ , с суммарным значением, равным  $1 + 1 + 1 - (3 - 1) = 1$ .

D. Загадочная посылка

ограничение по времени на тест: 1 second  
ограничение по памяти на тест: 64 megabytes

Петя решил поздравить своего друга из Австралии с предстоящим днем рождения, отправив ему открытку. Чтобы сделать свой сюрприз более загадочным, он решил создать цепь. Цепью назовем такую последовательность конвертов  $A = \{a_1, a_2, \dots, a_n\}$ , что ширина и высота  $i$ -го конверта строго больше ширины и высоты  $(i - 1)$ -го соответственно, а размером цепи — количество конвертов в ней. Из имеющихся у Пети конвертов он хочет создать наибольшую по размеру цепь, в которую можно было бы вложить открытку. Открытку можно вложить в цепь, если ширина и высота открытки меньше ширины и высоты меньшего конверта в цепи соответственно. Поворачивать конверты или открытку запрещено. Поскольку у Пети очень много конвертов и очень мало времени, то эта нелегкая задача возлагается на Вас.

Входные данные

Первая строка содержит целые числа  $n, w, h$  ( $1 \leq n \leq 5000, 1 \leq w, h \leq 10^6$ ) — количество конвертов, имеющихся у Пети в распоряжении, ширину и высоту открытки соответственно. Далее содержится  $n$  строк, в каждой из которых находится два целых числа  $w_i$  и  $h_i$  — ширина и высота  $i$ -го конверта ( $1 \leq w_i, h_i \leq 10^6$ ).

Выходные данные

В первую строку выведите размер максимальной цепи. Во вторую строку выведите через пробелы номера конвертов, образующих такую цепь, начиная с номера наименьшего конверта. Помните, что в наименьший конверт должна поместиться открытка. Если существует несколько цепей максимального размера, выведите любую.

Если открытка не помещается ни в один конверт, то выведите единственную строку, содержащую число 0.

Примеры

входные данные	Скопировать
2 1 1 2 2 2 2	
выходные данные	Скопировать
1 1	

входные данные	Скопировать
3 3 3 5 4 12 11 9 8	
выходные данные	Скопировать
3 1 3 2	

При поддержке



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

## С. Оптимизация дорог

ограничение по времени на тест: 3 секунды

ограничение по памяти на тест: 128 мегабайт

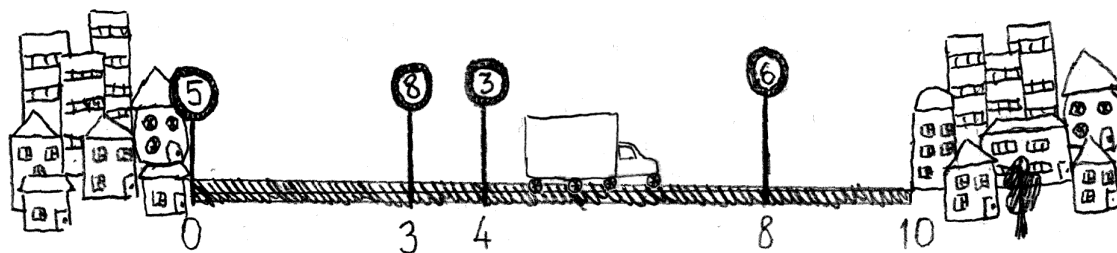
Правительство Марса заинтересовано не только в оптимизации космических перелетов, но и в улучшении дорожной системы планеты.

Одна из самых важных магистралей Марса соединяет Олимп-Сити и Кцолоп, столицу Кидонии. В рамках этой задачи будем рассматривать только путь от Кцолопа до Олимп-Сити, но не обратный путь (от Олимп-Сити до Кцолопа).

Дорога от Кцолопа до Олимп-Сити имеет длину  $\ell$  километров. Каждая точка дороги имеет координату  $x$  ( $0 \leq x \leq \ell$ ) — расстояние до Кцолопа в километрах. Таким образом, Кцолоп находится в точке с координатой 0, а Олимп-Сити — в точке с координатой  $\ell$ .

На дороге стоит  $n$  дорожных знаков, на  $i$ -м из которых написано ограничение скорости  $a_i$ . Это ограничение означает, что следующий километр следует проехать за  $a_i$  минут и действует до тех пор, пока по пути не встретится следующий дорожный знак. Один из дорожных знаков стоит в самом начале дороги (т. е. в точке с координатой 0) и задает начальную скорость.

Зная расположение всех дорожных знаков, нетрудно вычислить время поездки от Кцолопа до Олимп-Сити. Рассмотрим пример:



В данном случае необходимо проехать первые три километра за пять минут каждый, затем — один километр за восемь минут, затем — еще четыре километра за три минуты каждый и, наконец, последние два километра за шесть минут каждый. Итого время в пути составит  $3 \cdot 5 + 1 \cdot 8 + 4 \cdot 3 + 2 \cdot 6 = 47$  минут.

С целью оптимизации движения правительство Марса решило убрать не более  $k$  дорожных знаков. При этом знак в начале дороги убирать нельзя, иначе на начальном отрезке пути не будет никаких ограничений. Убирать знаки необходимо таким образом, чтобы минимизировать время в пути от Кцолопа до Олимп-Сити.

В Кидонии сосредоточены крупные промышленные предприятия Марса, поэтому оптимизация дороги до Олимп-Сити является приоритетной задачей. По этой причине правительство Марса поручило Вам решить эту задачу и выяснить, какие знаки необходимо убрать.

### Входные данные

В первой строке входных данных находится три целых числа  $n$ ,  $\ell$  и  $k$  ( $1 \leq n \leq 500$ ,  $1 \leq \ell \leq 10^5$ ,  $0 \leq k \leq n - 1$ ) — количество знаков на дороге от Кцолопа до Олимп-Сити, расстояние между этими городами и максимальное количество знаков, которые разрешается убрать.

Во второй строке входных данных находится  $n$  целых чисел  $d_i$  ( $d_1 = 0$ ,  $d_i < d_{i+1}$ ,  $0 \leq d_i \leq \ell - 1$ ) — координаты знаков.

В третьей строке входных данных находится  $n$  целых чисел  $a_i$  ( $1 \leq a_i \leq 10^4$ ) — ограничения скорости.

### Выходные данные

Выведите одно целое число — минимально возможное время в пути от Кцолопа до Олимп-Сити (в минутах), если убрать не более  $k$  дорожных знаков.

### Примеры

<b>входные данные</b>	<a href="#">Скопировать</a>
4 10 0 0 3 4 8 5 8 3 6	
<b>выходные данные</b>	<a href="#">Скопировать</a>
47	

<b>входные данные</b>	Скопировать
<div>4 10 2</div> <div>0 3 4 8</div> <div>5 8 3 6</div>	
<b>выходные данные</b>	Скопировать
<div>38</div>	

Примечание

В первом примере знаки удалять нельзя, и ответ равен 47, как сказано в условии задачи выше.

Во втором примере следует удалить второй и четвертый знаки. Тогда придется проехать первые четыре километра за  $4 \cdot 5 = 20$  минут, а последние шесть километров – за  $6 \cdot 3 = 18$  минут. Итого получаем  $4 \cdot 5 + 6 \cdot 3 = 38$  минут.

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов  
Соревнования по программированию 2.0  
Время на сервере: 29.11.2024 22:01:14<sup>UTC+5</sup> (k1).  
Мобильная версия, переключиться на [десктопную](#).  
[Privacy Policy](#)

При поддержке



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

### D. Солдат и игра с числами

ограничение по времени на тест: 3 секунды  
ограничение по памяти на тест: 256 мегабайт

Два солдата играют в игру. Сначала один солдат выбирает целое положительное число  $n$  и называет его второму солдату. Затем второй солдат пытается совершить наибольшее количество ходов. На каждом ходу солдат выбирает положительное целое число  $x > 1$ , такое, что  $n$  делится на  $x$ , и заменяет число  $n$  числом  $n / x$ . Когда  $n$  становится равно 1 и больше нет возможных ходов, тогда игра завершается и счет второго солдата равен количеству совершённых им ходов.

Чтобы сделать игру поинтереснее, в качестве  $n$  солдат выбирает число вида  $a! / b!$  для некоторых положительных целых чисел  $a$  и  $b$  ( $a \geq b$ ). Здесь под  $k!$  имеется в виду *факториал* числа  $k$ , который определяется как произведение всех положительных целых чисел, не превосходящих  $k$ .

Определите максимально возможный счет второго солдата.

#### Входные данные

В первой строке входа записано единственное целое число  $t$  ( $1 \leq t \leq 1\,000\,000$ ) обозначающее количество игр, в которые играют солдаты.

Затем следуют  $t$  строк, в каждой строке записана пара чисел  $a$  и  $b$  ( $1 \leq b \leq a \leq 5\,000\,000$ ), обозначающая значение  $n$  для игры.

#### Выходные данные

Для каждой игры выведите максимальный счет, которого может достичь второй солдат.

#### Примеры

<b>входные данные</b>	<a href="#">Скопировать</a>
2 3 1 6 3	
<b>выходные данные</b>	<a href="#">Скопировать</a>
2 5	

При поддержке



Е. Режим сна

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

У Вовы довольно странный режим сна. В сутках  $h$  часов. Вова будет спать ровно  $n$  раз. В  $i$ -й раз он будет спать ровно после  $a_i$  часов после пробуждения. Предположим, что Вова просыпается ровно в начале этой истории (изначальное время равно 0). Каждый раз Вова спит ровно сутки (другими словами,  $h$  часов).

Вова думает, что  $i$ -й раз, когда он спит, — **хороший**, если он идет спать между  $l$  и  $r$  часами включительно.

Вова может контролировать себя и перед  $i$ -м разом он может выбрать одно из двух: пойти спать после  $a_i$  часов или после  $a_i - 1$  часов.

Ваша задача — назвать максимальное число **хороших** раз, которые Вова может получить, если будет действовать оптимально.

Входные данные

Первая строка теста содержит четыре целых числа  $n, h, l$  и  $r$  ( $1 \leq n \leq 2000, 3 \leq h \leq 2000, 0 \leq l \leq r < h$ ) — количество раз, которое Вова будет спать, количество часов в сутках и отрезок **хорошего** времени для сна.

Вторая строка теста содержит  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i < h$ ), где  $a_i$  — количество часов, после которого Вова пойдет спать в  $i$ -й раз.

Выходные данные

Выведите одно целое число — максимальное количество **хороших** раз, которое Вова может получить, если будет действовать оптимально.

Пример

<b>входные данные</b>	<a href="#">Скопировать</a>
7 24 21 23 16 17 14 20 20 11 22	
<b>выходные данные</b>	<a href="#">Скопировать</a>
3	

Примечание

Максимальное количество **хороших** раз в тестовом примере равно 3.

История начинается с  $t = 0$ . Затем Вова идет спать после  $a_1 - 1$  часов, теперь время равно 15. Это время не является хорошим. Затем Вова идет спать после  $a_2 - 1$  часов, теперь время равно  $15 + 16 = 7$ . Это время также не является хорошим. Затем Вова идет спать после  $a_3$  часов, теперь время равно  $7 + 14 = 21$ . Это время является **хорошим**. Затем Вова идет спать после  $a_4 - 1$  часов, теперь время равно  $21 + 19 = 16$ . Это время не является хорошим. Затем Вова идет спать после  $a_5$  часов, теперь время равно  $16 + 20 = 12$ . Это время не является хорошим. Затем Вова идет спать после  $a_6$  часов, теперь время равно  $12 + 11 = 23$ . Это время является **хорошим**. Затем Вова идет спать после  $a_7$  часов, теперь время равно  $23 + 22 = 21$ . Это время также является **хорошим**.



В. Раскраска забора

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

Игорь влюбился в Таню. Чтобы показать, насколько сильно он ее любит, Игорь хочет написать на заборе число напротив дома Тани. Игорь считает, что чем больше число он напишет, тем больше у него шансов покорить Таню.

К сожалению, Игорь смог добыть только  $v$  литров краски. Сделав расчеты, Игорь решил, что на цифру  $d$  ему понадобится  $a_d$  литров краски. К тому же до Игоря дошли слухи, что Таня терпеть не может нули. Поэтому Игорь не будет их использовать в написании числа.

Помогите Игорю определить максимальное число, которое он может написать на заборе.

Входные данные

В первой строке записано целое число  $v$  ( $0 \leq v \leq 10^6$ ). Во второй строке записано девять положительных целых чисел  $a_1, a_2, \dots, a_9$  ( $1 \leq a_i \leq 10^5$ ).

Выходные данные

Выведите самое большое число, которое удастся написать Игорю на заборе. Если краски не хватит ни на одну цифру (Игорь не сможет ничего написать) выведите -1.

Примеры

входные данные	Скопировать
5 5 4 3 2 1 2 3 4 5	
выходные данные	Скопировать
55555	
входные данные	Скопировать
2 9 11 1 12 5 8 9 10 6	
выходные данные	Скопировать
33	
входные данные	Скопировать
0 1 1 1 1 1 1 1 1 1	
выходные данные	Скопировать
-1	





С. Юра и работа

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

Совсем недавно вышел новый iPhone 6, и Юра очень захотел себе его купить. К сожалению, денег у него не хватало, поэтому Юра устроился работать программистом. На работе Юра столкнулся со следующей задачей:

Задана последовательность из  $n$  чисел  $p_1, p_2, \dots, p_n$ . Нужно выбрать  $k$  пар целых чисел:

$[l_1, r_1], [l_2, r_2], \dots, [l_k, r_k] \ (1 \leq l_1 \leq r_1 < l_2 \leq r_2 < \dots < l_k \leq r_k \leq n; r_i - l_i + 1 = m),$   
так чтобы сумма  $\sum_{i=1}^k \sum_{j=l_i}^{r_i} p_j$  была как можно больше. Помогите Юре справиться с этим заданием.

**Входные данные**  
В первой строке содержится три целых числа  $n, m$  и  $k \ (1 \leq (m \times k) \leq n \leq 5000)$ . Во второй строке содержится  $n$  целых чисел  $p_1, p_2, \dots, p_n \ (0 \leq p_i \leq 10^9)$ .

**Выходные данные**  
В единственной строке выведите целое число — максимальное значение суммы.

Примеры

<b>входные данные</b>	<a href="#">Скопировать</a>
5 2 1 1 2 3 4 5	
<b>выходные данные</b>	<a href="#">Скопировать</a>
9	

<b>входные данные</b>	<a href="#">Скопировать</a>
7 1 3 2 10 7 18 5 33 0	
<b>выходные данные</b>	<a href="#">Скопировать</a>
61	



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

D. Легионы Цезаря

ограничение по времени на тест: 2 seconds  
ограничение по памяти на тест: 256 megabytes

Знаменитый полководец Гай Юлий Цезарь любил выстраивать воинов своей армии в шеренгу. Всего в армии было  $n_1$  пехотинцев и  $n_2$  всадников. Цезарь считал, что расстановка **не** красивая, если где-то в строю стоит подряд строго больше  $k_1$  пехотинцев или строго больше  $k_2$  всадников. Найдите количество красивых расстановок воинов.

Учтите, что в каждой расстановке должны присутствовать все  $n_1 + n_2$  воинов. Все пехотинцы считаются неразличимыми между собой. Аналогично, все всадники считаются неразличимыми между собой.

**Входные данные**  
В единственной строке через пробел записаны четыре целых числа  $n_1, n_2, k_1, k_2$  ( $1 \leq n_1, n_2 \leq 100, 1 \leq k_1, k_2 \leq 10$ ) — количество пехотинцев и всадников в армии, а также наибольшее допустимое количество стоящих подряд пехотинцев и всадников, соответственно.

**Выходные данные**  
Выведите количество красивых расстановок войск по модулю  $100000000$  ( $10^8$ ), то есть количество таких расстановок, где подряд стоит не более  $k_1$  пехотинцев и не более  $k_2$  всадников.

Примеры

<b>входные данные</b>	<a href="#">Скопировать</a>
2 1 1 10	
<b>выходные данные</b>	<a href="#">Скопировать</a>
1	

<b>входные данные</b>	<a href="#">Скопировать</a>
2 3 1 2	
<b>выходные данные</b>	<a href="#">Скопировать</a>
5	

<b>входные данные</b>	<a href="#">Скопировать</a>
2 4 1 1	
<b>выходные данные</b>	<a href="#">Скопировать</a>
0	

**Примечание**  
Обозначим пехотинца как 1, а всадника как 2.

В первом примере единственное красивое построение: 121

Во втором примере существует 5 красивых построений: 12122, 12212, 21212, 21221, 22121



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

### F. Последовательная подпоследовательность

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

Вам задан массив целых чисел длины  $n$ .

Вам необходимо выбрать такую подпоследовательность этого массива максимальной длины, что эта подпоследовательность является возрастающей последовательностью подряд идущих целых чисел. Иными словами подпоследовательность должна иметь вид  $[x, x + 1, \dots, x + k - 1]$  для некоторого  $x$  и длины  $k$ .

Подпоследовательность массива может быть получена при помощи удаления некоторого (возможно, нулевого) количества элементов из массива. Можно удалять элементы, которые не идут подряд. Оставшиеся элементы должны сохранить свой порядок. Например, для массива  $[5, 3, 1, 2, 4]$  следующие массивы являются подпоследовательностями:  $[3]$ ,  $[5, 3, 1, 2, 4]$ ,  $[5, 1, 4]$ , а массив  $[1, 3]$  не является.

#### Входные данные

Первая строка входных данных содержит целое число  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) — длину массива. Вторая строка входных данных содержит  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ) — сам массив.

#### Выходные данные

В первой строке выведите  $k$  — максимальную длину подпоследовательности заданного массива, которая образует возрастающую последовательность подряд идущих целых чисел.

Во второй строке выведите последовательность индексов **любой** подпоследовательности заданного массива максимальной длины, которая образует возрастающую последовательность подряд идущих целых чисел.

#### Примеры

<b>входные данные</b>	<a href="#">Скопировать</a>
7 3 3 4 7 5 6 8	
<b>выходные данные</b>	<a href="#">Скопировать</a>
4 2 3 5 6	
<b>входные данные</b>	<a href="#">Скопировать</a>
6 1 3 5 2 4 6	
<b>выходные данные</b>	<a href="#">Скопировать</a>
2 1 4	
<b>входные данные</b>	<a href="#">Скопировать</a>
4 10 9 8 7	
<b>выходные данные</b>	<a href="#">Скопировать</a>
1 1	
<b>входные данные</b>	<a href="#">Скопировать</a>
9 6 7 8 3 4 5 9 10 11	
<b>выходные данные</b>	<a href="#">Скопировать</a>
6 1 2 3 7 8 9	

#### Примечание

Все возможные правильные ответы для первого тестового примера (как последовательности индексов):

- $[1, 3, 5, 6]$

- [2, 3, 5, 6]

Все возможные правильные ответы для второго тестового примера :

- [1, 4]
- [2, 5]
- [3, 6]

Все возможные правильные ответы для третьего тестового примера :

- [1]
- [2]
- [3]
- [4]

Все возможные правильные ответы для четвертого тестового примера:

- [1, 2, 3, 7, 8, 9]

---

[Codeforces](#) (с) Copyright 2010-2024 Михаил Мирзаянов  
Соревнования по программированию 2.0  
Время на сервере: 29.11.2024 22:00:21<sup>UTC+5</sup> (к1).  
Мобильная версия, переключиться на [десктопную](#).  
[Privacy Policy](#)

При поддержке



**ІІТМО**

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

D. Цветы

ограничение по времени на тест: 1.5 секунд  
ограничение по памяти на тест: 256 мегабайт

Мы видели маленькую игру, которую Сурок приготовил для Крота на обед. Теперь Сурку пора ужинать — а мы все знаем, что Сурок ест цветы. За каждым ужином он ест немного красных и немного белых цветов. Таким образом, ужин можно представить как последовательность белых и красных цветов.

Но для того, чтобы ужин был вкусный, есть одно правило: когда Сурок хочет отведать белых цветов, он потребляет их только группами размера  $k$ .

Теперь Сурку интересно, сколько существует способов съесть от  $a$  до  $b$  цветов по его правилам. Так как количество способов может быть очень большим, выведите его по модулю  $1000000007$  ( $10^9 + 7$ ).

Входные данные

На вход подаётся несколько наборов входных данных.

В первой строке записано два целых числа  $t$  и  $k$  ( $1 \leq t, k \leq 10^5$ ), где  $t$  обозначает количество тестовых примеров.

В следующих  $t$  строках записано по два целых числа  $a_i$  и  $b_i$  ( $1 \leq a_i \leq b_i \leq 10^5$ ), описывающих  $i$ -й тест.

Выходные данные

Выведите  $t$  строк. В  $i$ -й строке должно содержаться количество способов, которыми Сурок может съесть от  $a_i$  до  $b_i$  цветков за ужином, взятое по модулю  $1000000007$  ( $10^9 + 7$ ).

Примеры

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

Примечание

- При  $K = 2$  и длине 1 Сурок может съесть  $(R)$ .
- При  $K = 2$  и длине 2 Сурок может съесть  $(RR)$  и  $(WW)$ .
- При  $K = 2$  и длине 3 Сурок может съесть  $(RRR)$ ,  $(RWW)$  и  $(WWR)$ .
- При  $K = 2$  и длине 4 Сурок может съесть, например,  $(WWWW)$  или  $(RWWR)$ . Но вот  $(WWWR)$ , к примеру, съесть он не может.

