

## Задача А. Наибольшая общая подпоследовательность

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 64 мегабайта

Даны две последовательности. Найдите длину их наибольшей общей подпоследовательности (подпоследовательность — это то, что можно получить из данной последовательности вычеркиванием некоторых элементов).

### Формат входных данных

В первой строке входных данных записано целое число  $N$  — длина первой последовательности ( $1 \leq N \leq 1000$ ). Во второй строке записаны  $N$  чисел — члены первой последовательности. В третьей строке записано целое число  $M$  — длина второй последовательности ( $1 \leq M \leq 1000$ ). В четвертой строке записаны  $M$  чисел — члены второй последовательности. Члены последовательностей — целые числа, не превосходящие 10 000 по модулю.

### Формат выходных данных

Программа должна вывести единственное целое число — длину наибольшей общей подпоследовательности, или число 0, если такой не существует.

### Примеры

<code>stdin</code>	<code>stdout</code>
3 1 2 3 4 2 1 3 5	2

## Задача В. Калькулятор

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 4 секунды  
Ограничение по памяти: 64 мегабайта

Имеется калькулятор, который выполняет следующие операции:

- умножить число  $X$  на 2;
- умножить число  $X$  на 3;
- прибавить к числу  $X$  единицу.

Определите, какое наименьшее количество операций требуется, чтобы получить из числа 1 число  $N$ .

### Формат входных данных

Во входном файле написано натуральное число  $N$ , не превосходящее  $10^6$ .

### Формат выходных данных

В первой строке выходного файла выведите минимальное количество операций. Во второй строке выведите числа, последовательно получающиеся при выполнении операций. Первое из них должно быть равно 1, а последнее  $N$ . Если решений несколько, выведите любое.

### Примеры

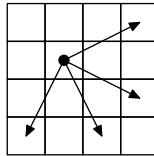
<code>stdin</code>	<code>stdout</code>
1	0 1
5	3 1 3 4 5
962340	17 1 3 9 27 54 55 165 495 1485 4455 8910 17820 17821 53463 160389 160390 481170 962340

## Задача С. Ход конём - 2

Имя входного файла: `knight2.in`  
Имя выходного файла: `knight2.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Дана прямоугольная доска  $N \times M$  ( $N$  строк и  $M$  столбцов). В левом верхнем углу находится шахматный конь, которого необходимо переместить в правый нижний угол доски.

При этом конь может ходить следующим образом:



Необходимо определить, сколько существует различных маршрутов, ведущих из левого верхнего в правый нижний угол.

### Формат входных данных

Входной файл содержит два натуральных числа  $N$  и  $M$  ( $1 \leq N, M \leq 50$ ). Гарантируется, что ответ на задачу не превышает 1000000000.

### Формат выходных данных

В выходной файл выведите единственное число — количество способов добраться конём до правого нижнего угла доски.

### Примеры

<code>knight2.in</code>	<code>knight2.out</code>
4 4	2
2 3	1

## Задача D. Покупка билетов

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

За билетами на премьеру нового мюзикла выстроилась очередь из  $N$  человек, каждый из которых хочет купить 1 билет. На всю очередь работала только одна касса, поэтому продажа билетов шла очень медленно, приводя «постояльцев» очереди в отчаяние. Самые сообразительные быстро заметили, что, как правило, несколько билетов в одни руки кассир продаёт быстрее, чем когда эти же билеты продаются по одному. Поэтому они предложили нескольким подряд стоящим людям отдавать деньги первому из них, чтобы он купил билеты на всех.

Однако для борьбы со спекулянтами кассир продавала не более 3-х билетов в одни руки, поэтому договориться таким образом между собой могли лишь 2 или 3 подряд стоящих человека.

Известно, что на продажу  $i$ -му человеку из очереди одного билета кассир тратит  $A_i$  секунд, на продажу двух билетов —  $B_i$  секунд, трех билетов —  $C_i$  секунд. Напишите программу, которая подсчитает минимальное время, за которое могли быть обслужены все покупатели.

Обратите внимание, что билеты на группу объединившихся людей всегда покупает первый из них. Также никто в целях ускорения не покупает лишних билетов (то есть билетов, которые никому не нужны).

### Формат входных данных

В первой строке входного файла находится число  $N$  ( $1 \leq N \leq 5000$ ) - количество человек в очереди. В следующих  $N$  строках написаны по три натуральных числа  $A_i$ ,  $B_i$ ,  $C_i$ , каждое из этих чисел не превышает 3600. Люди в очереди нумеруются начиная от кассы.

### Формат выходных данных

В выходной файл выведите одно число — минимальное время в секундах, за которое могли быть обслужены все покупатели.

### Примеры

стандартный ввод	стандартный вывод
5	12
5 10 15	
2 10 15	
5 5 5	
20 20 1	
20 1 1	

## Задача Е. Путь в ад

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

**Предупреждение:** условие этой задачи не рекомендуется к прочтению слаборезным и пожилым людям, беременным женщинам и пассажирам с детьми.

В стране Мурляндии открылся новый аттракцион с говорящим названием «Путь в ад». Все котки сразу уже устремились в парк развлечений, чтобы опробовать его, и за билетами выстроились огромные очереди. Пока котки ждут своей очереди, им становится очень скучно, и приходится придумывать чем себя занять. Самый умный котик Пирожок решил посчитать количество способов прокатиться на этом аттракционе.

Аттракцион представляет из себя три параллельных рельса длины  $n - 1$  метров, по любому из которых может (достаточно медленно, чтобы котки не боялись) катиться вагонетка. Каждый метр на рельсах установлены препятствия, которые заставляют вагонетку подскакивать и приземляться обратно, но, возможно, на соседний рельс. Таким образом, если вагонетка ехала по крайнему рельсу, она может приземлиться на него или на центральный рельс, а если ехала по центральному — может приземлиться на любой.

Каждый способ прокатиться на аттракционе задается последовательностью номеров рельсов, на которых вагонетка находилась на каждом метре. Стартовать вагонетка может только с центрального. Помогите Пирожку посчитать количество способов прокатиться на аттракционе!

### Формат входных данных

В единственной строке ввода дано целое число  $n$  ( $1 \leq n \leq 1000$ ) — количество метровых отметок на рельсах «Пути в ад» (таким образом, длина рельсов равна  $n - 1$ ).

### Формат выходных данных

Выведите единственное целое число — количество способов прокатить котика до ада в парке развлечений Мурляндии по модулю  $10^9 + 7$

### Примеры

стандартный ввод	стандартный вывод
3	7

## Задача F. Наибольшая возрастающая подпоследовательность

Имя входного файла: `lis.in`  
Имя выходного файла: `lis.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Дана последовательность, требуется найти длину её наибольшей строго возрастающей подпоследовательности. Подпоследовательность — это часть последовательности, получающаяся из нее вычеркиванием каких-то элементов.

### Формат входных данных

В первой строке входного файла задано целое число  $N$  — длина последовательности ( $1 \leq N \leq 1000$ ). Во второй строке задаётся сама последовательность (разделитель — пробел). Элементы последовательности — целые числа, не превосходящие 10 000 по модулю.

### Формат выходных данных

Требуется вывести длину наибольшей возрастающей подпоследовательности.

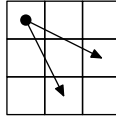
### Примеры

<code>lis.in</code>	<code>lis.out</code>
6 3 29 5 5 28 6	3

## Задача G. Ход конём

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Дана прямоугольная доска  $N \times M$  ( $N$  строк и  $M$  столбцов). В левом верхнем углу находится шахматный конь, которого необходимо переместить в правый нижний угол доски. В данной задаче конь может перемещаться на две клетки вниз и одну клетку вправо или на одну клетку вниз и две клетки вправо.



Необходимо определить, сколько существует различных маршрутов, ведущих из левого верхнего в правый нижний угол.

### Формат входных данных

Входной файл содержит два натуральных числа  $N$  и  $M$  ( $1 \leq N, M \leq 50$ ).

### Формат выходных данных

В выходной файл выведите единственное число — количество способов добраться конём до правого нижнего угла доски.

### Примеры

<code>stdin</code>	<code>stdout</code>
3 2	1
31 34	293930

## Задача Н. Мафия

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Пока у школьников идёт зачёт, преподаватели в Тайной комнате играют в мафию. В кругу сидит  $N$  преподавателей. Ведущий должен раздать кому-то из них карты с тузами (тузов любое количество, возможно, 0) — эти преподаватели будут мафией. Однако никакие два мафиози не должны сидеть рядом. Сколько способов раздать карты есть у ведущего? (Два способа считаются различными, если есть хотя бы один преподаватель, который является мафией в одном случае, но не является в другом).

### Формат входных данных

В первой строчке число  $N$  — количество преподавателей, которые сидят в кругу ( $1 \leq N \leq 100\,000$ ).

### Формат выходных данных

Выведите одно число — количество способов раздать карты. Так как ответ может быть очень большим, выведите остаток от деления его на  $10^9 + 7$ .

### Примеры

<code>stdin</code>	<code>stdout</code>
4	7
5	11

## Задача I. Необходимые разбиения

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Демид, Никита, Тимофей и Олег решили отправиться вверх по Волге на лодке, от Костромы до Ярославля. Прежде чем отправиться в путь, они собирают чемоданы и берут с собой самое необходимое.

Демид сказал, что в таком трудном и опасном путешествии нельзя обойтись без упорядоченных разбиений числа  $n$  на слагаемые. Напомним, что *упорядоченным разбиением числа  $n$  на слагаемые* называется любая последовательность натуральных чисел, в сумме дающая  $n$ . Тимофей заметил, что брать с собой разбиения, в которых есть числа, большие  $r$  — пустая трата и без того небольшого места в чемоданах. Никита же заявил, что они вполне могут обойтись и без разбиений, в которых есть числа, меньшие  $l$ . Нельзя сказать, чтобы он путешествовал хоть раз без них, но знал людей, которые путешествовали. Олег же лишь одобрительно кивнул.

Помогите друзьям посчитать количество таких разбиений (то есть разбиений, в которых каждое слагаемое принадлежит отрезку  $[l; r]$ ). Так как искомое число может быть очень большим, выведите остаток от деления этого числа на  $10^9 + 7$ .

### Формат входных данных

Первая строка входных данных содержит три целых числа  $n, l$  и  $r$  ( $1 \leq l \leq r \leq n \leq 500\,000$ ).

### Формат выходных данных

В единственную строку выведите ответ на задачу — остаток от деления искомого количества разбиений на число  $10^9 + 7$ .

### Примеры

стандартный ввод	стандартный вывод
5 2 5	3
10 3 8	8

### Замечание

В первом примере подходят разбиения:

$$5 = 5$$

$$5 = 2 + 3$$

$$5 = 3 + 2$$

Во втором примере:

$$10 = 7 + 3$$

$$10 = 3 + 7$$

$$10 = 6 + 4$$

$$10 = 4 + 6$$

$$10 = 5 + 5$$

$$10 = 3 + 3 + 4$$

$$10 = 3 + 4 + 3$$

$$10 = 4 + 3 + 3$$

## Задача J. Наибольшая последовательнократная подпоследовательность

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Для заданной числовой последовательности  $a_1, a_2, \dots, a_n$  требуется найти длину максимальной последовательнократной подпоследовательности.

Для последовательнократной подпоследовательности  $a_{k_1}, a_{k_2}, \dots, a_{k_t}$  ( $k_1 < k_2 < \dots < k_t$ ) верно, что  $a_{k_i} | a_{k_j}$  при  $1 \leq i < j \leq t$  (утверждение « $a|b$ » эквивалентно « $b$  кратно  $a$ »). Подпоследовательность из одного элемента полагается последовательнократной по определению.

### Формат входных данных

В первой строке входного файла записаны  $N$  натуральных чисел ( $1 \leq N \leq 1000$ ), не превосходящих  $2 \cdot 10^9$  — последовательность.

### Формат выходных данных

Вывести единственное число, равное длине максимальной последовательнократной подпоследовательности.

### Примеры

стандартный ввод	стандартный вывод
3 6 5 12	3

## Задача К. Трудная жизнь преподавателя

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

После кошмарной ночи преподаватель Андрей принимал решения задачи у детей. Он был очень уставшим, и хотел было отдохнуть, но задачи не ждут! Помогите ему, пожалуйста, сэкономить силы.

По итогам проверки решения Андрей может поставить одним из трёх вердиктов: *OK*, *RJ* или *DQ*. В то же время решение школьника заслуживает одного из этих трёх вердиктов: *OK*, *RJ* или *DQ*. Так как Андрей — злой и несправедливый препод, то если решение школьника должно получить *OK*, то Андрей хочет поставить *RJ*. Если школьник должен получить *RJ*, то препод хочет поставить *DQ*. А при должном вердикте *DQ*, Андрею становится жалко ученика, и он ставит ему *OK*.

Однако так как злой Андрей очень устал, он хочет сделать как можно меньше движений для, собственно, перемещения курсора между разными кнопками. Поэтому Андрей может переместить курсор мыши между разными кнопками (*OK*, *RJ* и *DQ*) не более  $K$  раз.

Скажите, какое максимальное количество вердиктов он сможет поставить, соблюдая свои принципы.

### Формат входных данных

Первая строка содержит два числа  $N$  — число решений, присланных школьниками, и  $K$  — число перемещений мышки, которые под силу Андрею ( $1 \leq N \leq 20\,000$ ,  $0 \leq K \leq 20$ ).

В следующих  $N$  строках записано по одному символу: *O*, *R* или *D* (без кавычек), где каждая буква соответствует первому символу вердикта, который должен получить школьник за посылку (соответственно *OK*, *RJ* и *DQ*).

### Формат выходных данных

Выведите максимальное количество правильных (в понимании Андрея) вердиктов, которые он сможет поставить, при условии что он не может переместить курсор мышки более, чем  $K$  раз.

### Примеры

стандартный ввод	стандартный вывод
5 1 R R O R D	4

## Задача L. Невозрастающая подпоследовательность

Имя входного файла: `subseq.in`  
Имя выходного файла: `subseq.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Вам требуется написать программу, которая по заданной последовательности находит максимальную невозрастающую её подпоследовательность (т.е такую последовательность чисел  $a_{i_1}, a_{i_2}, \dots, a_{i_k}$  ( $i_1 < i_2 < \dots < i_k$ ), что  $a_{i_1} \geq a_{i_2} \geq \dots \geq a_{i_k}$  и не существует последовательности с теми же свойствами длиной  $k + 1$ ).

### Формат входных данных

В первой строке задано число  $n$  — количество элементов последовательности ( $1 \leq n \leq 239\,017$ ). В последующих строках идут сами числа последовательности  $a_i$ , отделенные друг от друга произвольным количеством пробелов и переводов строки (все числа не превосходят по модулю  $2^{31} - 2$ ).

### Формат выходных данных

Вам необходимо выдать в первой строке выходного файла число  $k$  — длину максимальной невозрастающей подпоследовательности. Во второй строке должны быть выведены все номера элементов исходной последовательности  $i_j$ , образующих искомую подпоследовательность. Номера выводятся в порядке возрастания. Если оптимальных решений несколько, разрешается выводить любое.

### Примеры

<code>subseq.in</code>	<code>subseq.out</code>
5	3
5 8 10 4 1	1 4 5