

## Задача А. Генетический кот

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

Геном котов с планеты Уран состоит из последовательности четырёх химических соединений, обозначаемых символами  $A$ ,  $C$ ,  $T$  и  $G$ . При этом:

- Геном каждого кота состоит ровно из  $N$  элементов
- Геном не может начинаться с  $C$
- В геноме не могут встречаться участки  $AA$ ,  $CA$ ,  $AT$  и  $GC$
- Геном не может заканчиваться на  $C$  и  $T$

Окрас уранского кота однозначно задаётся его геномом. Котики просят вас посчитать, сколько различных окрасов у них может существовать. Так как котики не любят больших чисел, то ответ надо вывести по модулю  $10^9 + 7$ .

### Формат входных данных

В единственной строке задано одно число  $N$  ( $1 \leq N \leq 50000$ ).

### Формат выходных данных

Выведите единственное число — количество различных окрасов по модулю  $10^9 + 7$ .

### Примеры

<code>genome.in</code>	<code>genome.out</code>
3	14
100	927396138

## Задача В. Плавные числа

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

Назовём натуральное число плавным, если значения соседних цифр отличаются не более, чем на

1. Определите количество  $N$ -значных плавных чисел. Запись числа не может начинаться с цифры 0.

### Формат входных данных

Программа получает на вход одно целое число  $N$  ( $1 \leq N \leq 20$ ).

### Формат выходных данных

Программа должна вывести одно целое число — искомое количество плавных чисел.

### Примеры

<code>numbers.in</code>	<code>numbers.out</code>
2	26

## Задача С. 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 D. Наибольшая общая подпоследовательность

Имя входного файла: `lcs.in`  
Имя выходного файла: `lcs.out`  
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>lcs.in</code>	<code>lcs.out</code>
3 1 2 3 4 2 1 3 5	2

## Задача Е. Гладиолус

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

Числовая последовательность задана рекуррентной формулой:  $a_{i+1} = (ka_i + b) \bmod m$ . Найдите её наибольшую возрастающую подпоследовательность.

### Формат входных данных

Программа получает на вход пять целых чисел: длину последовательности  $n$  ( $1 \leq n \leq 10^5$ ), начальный элемент последовательности  $a_1$ , параметры  $k, b, m$  для вычисления последующих членов последовательности ( $1 \leq m \leq 10^4, 0 \leq k < m, 0 \leq b < m, 0 \leq a_1 < m$ ).

### Формат выходных данных

Требуется вывести наибольшую возрастающую подпоследовательность данной последовательности, разделяя числа пробелами. Если таких последовательностей несколько, необходимо вывести одну (любую) из них.

### Примеры

<code>gladiolus.in</code>	<code>gladiolus.out</code>
5 41 2 1 100	41 67 71

### Замечание

В данном примере последовательность состоит из 5 элементов:  $a_1 = 41, a_{i+1} = (2a_i + 1) \bmod 100$ , то есть последовательность имеет вид 41, 83, 67, 35, 71.

## Задача F. 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

Во входном файле записано  $N$  троек натуральных чисел  $A_i, B_i, C_i$  ( $1 \leq N \leq 5000$ ). 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	