Задача 1. Последовательности

Источник:	базовая
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Посчитайте количество последовательностей длины N из нулей и единиц, не содержащих двух единиц подряд.

Формат входных данных

Во входном файле задано целое число N — длина последовательности ($1 \leqslant N \leqslant 40$).

Формат выходных данных

В выходной файл необходимо вывести целое число — количество последовательностей длины N с заданным свойством.

Пример

input.txt	output.txt
1	2
2	3

Задача 2. Делимость

Источник:	базовая
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Рассмотрим произвольную последовательность целых чисел. Можно поставить знаки операций `+` или `-` между целыми в данной последовательности, получая при этом различные арифметические выражения, которые при их вычислении имеют различные значения. Давайте, например, возьмем следующую последовательность: $17, 5, -21, 15$. Из нее можно получить восемь различных выражений:

```
17 + 5 + -21 + 15 = 16
17 + 5 + -21 - 15 = -14
17 + 5 - -21 + 15 = 58
17 + 5 - -21 - 15 = 28
17 - 5 + -21 + 15 = 6
17 - 5 + -21 - 15 = -24
17 - 5 - -21 + 15 = 48
17 - 5 - -21 - 15 = 18
```

Назовем последовательность делимой на K, если можно так расставить операции `+` или `-` между целыми в последовательности, что значение полученного выражения делилось бы нацело на K. В приведенном выше примере последовательность делима на 7 ($17 + 5 + -21 - 15 = -14$), но не делима на 5.

Напишите программу, которая определяет делимость последовательности целых чисел.

Формат входных данных

Первая строка входного файла содержит два целых числа N и K, записанных через пробел ($1 \leqslant N \leqslant 10000, 2 \leqslant K \leqslant 100$).

Во второй строке записано через пробел N целых чисел. Каждое число по модулю не превосходит 10000.

Формат выходных данных

В выходной файл нужно выдать слово `Divisible`, если данная последовательность делима на K. В противном случае нужно вывести `Not divisible`.

Пример

input.txt	output.txt
4 7 17 5 -21 15	Divisible
4 5 17 5 -21 15	Not divisible

Задача 3. Рюкзак

Источник:	базовая
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	разумное
Ограничение по памяти:	разумное

Имеется N предметов, для каждого предмета известны его вес и его стоимость. Хочется унести набор предметов максимальной суммарной стоимости. Однако унести предметы можно только в рюкзаке. Рюкзак один, и он выдерживает вес не более W.

Требуется определить, какие предметы надо забрать, так чтобы рюкзак выдержал их суммарный вес, и чтобы их суммарная стоимость была максимальна.

Формат входных данных

В первой строке дано два целых числа: N — количество предметов и W — какой максимальный вес выдерживает рюкзак ($1 \leqslant N \leqslant 200$, $1 \leqslant W \leqslant 5 \cdot 10^4$).

Далее идёт N строк, которые описывают предметы. В каждой строке записано два целых числа: w_i — вес предмета и c_i — стоимость предмета ($1 \leqslant w_i \leqslant W$, $1 \leqslant c_i \leqslant 10^6$).

Формат выходных данных

На выход нужно вывести оптимальное решение.

В первую строку нужно вывести три целых числа: K — количество взятых предметов, $\overline{W}$ — суммарный вес этих предметов и $\overline{C}$ — суммарная стоимость этих предметов. Во второй строке нужно вывести K целых чисел — номера взятых предметов в любом порядке. Предметы нумеруются в порядке их записи во входных данных, начиная с единицы.

Если решений несколько, можно вывести любое из них.

Пример

input.txt	output.txt
5 20 5 10 4 12 6 3 3 8 6 20	4 18 50 1 2 4 5

Задача 4. Число разбиений

Источник:	основная
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Разбиением числа N на слагаемые называется набор целых положительных чисел, сумма которых в точности равна N. Два разбиения считаются одинаковыми, если соответствующие наборы отличаются только порядком чисел. Нужно найти количество различных разбиений числа N на слагаемые.

Поскольку ответ может быть большим, найдите его остаток от деления на $10^9 + 7$.

Формат входных данных

В первой строке дано одно целое число N — число, разбиения которого нужно считать ($1 \leqslant N \leqslant 1\,000$).

Формат выходных данных

Найдите количество способ разбить число N на целые положительные слагаемые, и выведите остаток от деления этого числа на $1\,000\,000\,007$.

Пример

input.txt	output.txt
10	42
20	627
1000	709496666

Задача 5. Гангстеры

Источник:	основная
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

N гангстеров идут в ресторан. i-тый гангстер заходит в T_i-е время и имеет при себе P_i денег.

Дверь ресторана имеет $k+1$ стадий открытия, выраженных в целых числах от 0 до K. Состояние открытия может измениться на 1 в единицу времени, т.е. либо открыться на 1, либо закрыться на 1, либо остаться прежним. В начальный момент состояние двери закрытое $= 0$.

i-тый гангстер может войти в ресторан, если дверь открыта специально для него, т.е. состояние двери совпадает с шириной его плеч S_i. Если в момент времени, когда гангстер подошел к ресторану, состояние открытия двери не совпадает с шириной его плеч, то он уходит и никогда не возвращается.

Ресторан работает в интервале времени $[0, T]$.

Цель: собрать в ресторане гангстеров с максимальным количеством денег.

Формат входных данных

Первая строка входного файла содержит значения N, K и T, разделенные пробелами ($1 \leqslant N \leqslant 100, 1 \leqslant K \leqslant 100, 0 \leqslant T \leqslant 30000$).

Вторая строка содержит моменты времени, в которые гангстеры подходят к ресторану T_1, T_2, ... , T_N, разделенные пробелами ($0 \leqslant T_i \leqslant T$ для $i = 1, 2, ..., N$).

В третьей строке записаны суммы денег каждого гангстера P_1, P_2, ... , P_N, разделенные пробелами ($0 \leqslant P_i \leqslant 300$, для $i = 1, 2, ..., N$).

Четвертая строка содержит значения ширины плеч каждого гангстера, разделенные пробелами ($0 \leqslant S_i \leqslant K$ для $i = 1, 2, ..., N$).

Все значения целые.

Формат выходных данных

В выходной файл выдать одно целое число — максимальное значение достатка всех гангстеров, собранных в ресторане. Если ни один гангстер не может попасть в ресторан, выдать `0`.

Пример

input.txt	output.txt
4 10 20 10 16 8 16 10 11 15 1 10 7 1 8	26
2 17 100 5 0 50 33 6 1	0

Задача 6. Наибольшая возрастающая подпоследовательность

Источник:	основная
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	разумное
Ограничение по памяти:	разумное

Дан массив из N чисел. Нужно найти в этом массиве такую подпоследовательность, что:

1. Числа этой подпоследовательности строго возрастают (слева направо).
2. Количество элементов в этой подпоследовательности максимально возможное.

Формат входных данных

В первой строке дано одно целое число N — размер массива ($1 \leqslant N \leqslant 5\,000$). Во второй строке записано N знаковых 32-битных целых чисел через пробел.

Формат выходных данных

В первую строку нужно вывести целое число K — количество элементов в искомой подпоследовательности. Саму подпоследовательность нужно ввести в оставшихся K строках. Каждый элемент подпоследовательности следует выводить в формате "`A[i] = k`", где i — индекс элемента (нумеруя с единицы), а k — значение элемента. Естественно, элементы подпоследовательности нужно выводить в порядке возрастания.

Если решений несколько, можно вывести любое из них.

Пример

input.txt	output.txt
12 18 3 18 5 7 10 5 18 20 19 7 18	6 A[2] = 3 A[4] = 5 A[5] = 7 A[6] = 10 A[8] = 18 A[9] = 20

Задача 7. Телефонный номер

Источник:	повышенной сложности
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Если вы обратили внимание, то клавиатура многих телефонов выглядит следующим образом:

1	2 ABC	3 DEF
4 GHI	5 JKL	6 MN
7 PRS	8 TUV	9 WXY
	0 OQZ	

Использование изображенных на клавишах букв позволяет представить номер телефона в виде легко запоминающихся слов, что бывают часто более удобным, чем традиционная запись телефона в виде последовательности цифр. Многие фирмы пользуются этим и стараются подобрать себе номер телефона так, чтобы он содержал как можно больше букв из имени фирмы.

Требуется написать программу, которая преобразует исходный цифровой номер телефона в соответствующую последовательность букв и цифр, содержащую как можно больше символов из названия фирмы. При этом буквы из названия фирмы должны быть указаны в полученном номере в той же последовательности, в которой они встречаются в названии фирмы. Например, если фирма называется `IBM`, а исходный номер телефона — `246`, то замена его на `BIM` не допустима, тогда как замена его на `2IM` или `B4M` является правильной.

Формат входных данных

Первая строка входного файла содержит название фирмы. Она состоит только из заглавных букв латинского алфавита, количество которых не превышает 260 символов.

Вторая строка содержит номер телефона в виде последовательности цифр. Цифр в номере телефона также не более 260.

Формат выходных данных

В единственной строке выходного файла должно содержаться число букв из измененного номера.

Пример

input.txt	output.txt
IBM 246	2

Задача 8. Наибольшая возрастающая подпоследовательность+

Источник:	повышенной сложности
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	разумное
Ограничение по памяти:	разумное

Дан массив из N чисел. Нужно найти в этом массиве такую подпоследовательность, что:
1. Числа этой подпоследовательности строго возрастают (слева направо).
2. Количество элементов в этой подпоследовательности максимально возможное.

Формат входных данных

В первой строке дано одно целое число N — размер массива ($1 \leqslant N \leqslant 100\,000$). Во второй строке записано N знаковых 32-битных целых чисел через пробел.

Формат выходных данных

В первую строку нужно вывести целое число K — количество элементов в искомой подпоследовательности. Саму подпоследовательность нужно ввести в оставшихся K строках. Каждый элемент подпоследовательности следует выводить в формате "`A[i] = k`", где i — индекс элемента (нумеруя с единицы), а k — значение элемента. Естественно, элементы подпоследовательности нужно выводить в порядке возрастания.

Если решений несколько, можно вывести любое из них.

Пример

input.txt	output.txt
12 18 3 18 5 7 10 5 18 20 19 7 18	6 A[2] = 3 A[4] = 5 A[5] = 7 A[6] = 10 A[8] = 18 A[9] = 20

Комментарий

Один из возможных способов решения:
1. Если отсортировать все значения в последовательности и пронумеровать их по порядку, то можно заменить каждый элемент последовательности на его номер. После такого преобразования все элементы будут в диапазоне от 0 до $N - 1$.
2. На каждом шаге динамического программирования нужно выбрать максимальный известный результат среди тех элементов, которые больше/меньше некоторого порога. Этот выбор максимума сводится к выбору максимума на отрезке, и его можно ускорить с помощью блоков.