

### Задача А. Рюкзак

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

Найдите максимальный вес золота, который можно унести в рюкзаке вместительностью  $S$ , если есть  $N$  золотых слитков с заданными весами.

#### Формат входного файла

В первой строке входного файла записаны два числа —  $S$  и  $N$  ( $1 \leq S \leq 10\,000$ ,  $1 \leq N \leq 300$ ). Далее следует  $N$  неотрицательных целых чисел, не превосходящих 100 000 — веса слитков.

#### Формат выходного файла

Выведите искомый максимальный вес.

#### Пример

<code>knapsack.in</code>	<code>knapsack.out</code>
10 3 5 7 4	9

### Задача В. Наибольшая общая подпоследовательность

Имя входного файла: `lcs.in`  
Имя выходного файла: `lcs.out`  
Ограничение по времени: 1 секунда  
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 входного файла

В первой строке входного файла записано число  $N$  — длина первой последовательности ( $1 \leq N \leq 1000$ ).

Во второй строке записаны члены первой последовательности (через пробел) — целые числа, не превосходящие 10 000 по модулю.

В третьей строке записано число  $M$  — длина второй последовательности ( $1 \leq M \leq 1000$ ).

В четвертой строке записаны члены второй последовательности (через пробел) — целые числа, не превосходящие 10 000 по модулю.

#### Формат выходного файла

В выходной файл требуется вывести единственное целое число: длину наибольшей общей подпоследовательности, или число 0, если такой не существует.

### Пример

<code>lcs.in</code>	<code>lcs.out</code>
3 1 2 3 4 2 1 3 5	2

### Задача С. Футбол

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

Олег — большой любитель футбола и статистики. Недавно он нашел результаты участия его любимой команды в каком-то давнем чемпионате. К сожалению, единственной сохранившейся информацией оказалось то, сколько матчей было сыграно и сколько очков набрала команда. Напоминаем, что если матч завершается победой команды, то ей присуждается три очка, ничьей — одно очко, и если команда проигрывает матч, то она не получает ни одного очка.

Олегу стало интересно, сколько различных вариантов прохождения чемпионата было у его любимой команды. Различными считаются варианты, если результат хотя бы одного матча различен, причем счет не принимается во внимание, а учитывается только то, завершился матч победой, ничьей или проигрышем.

#### Формат входного файла

Первая строка входного файла содержит два целых числа:  $n$  ( $0 \leq n \leq 36$ ) — количество набранных командой очков, и  $k$  ( $1 \leq k \leq 12$ ) — количество матчей, сыгранных этой командой в чемпионате.

#### Формат выходного файла

В выходной файл выведите число различных вариантов прохождения чемпионата любимой команды Олега.

#### Примеры

<code>football.in</code>	<code>football.out</code>
3 2	2
4 3	6

### Задача D. Наибольшая возрастающая подпоследовательность

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

Дана последовательность, требуется найти длину её наибольшей возрастающей

подпоследовательности.

### Формат входного файла

В первой строке входных данных задано число  $N$  — длина последовательности ( $1 \leq N \leq 100000$ ). Во второй строке задается сама последовательность (разделитель — пробел). Элементы последовательности — целые числа, не превосходящие 10000 по модулю.

### Формат выходного файла

Требуется вывести длину наибольшей возрастающей подпоследовательности.

### Пример

sequence.in	sequence.out
6 3 29 5 5 28 6	3