

## Задача А. Число Фибоначчи

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

Числа Фибоначчи  $F_0, F_1, F_2, \dots, F_n$  определяются следующим образом:  $F_0 = F_1 = 1$ , а для любого  $n > 1$  выполнено равенство  $F_n = F_{n-1} + F_{n-2}$ .

По заданному числу  $n$  выведите число Фибоначчи  $F_n$ .

### Формат входных данных

В первой строке входного файла задано единственное число  $n$  ( $0 \leq n \leq 45$ ).

### Формат выходных данных

Выполните число  $F_n$  в первой строке выходного файла.

### Примеры

fib.in	fib.out
1	1
2	2
3	3
4	5
5	8
6	13

## Задача В. Деньги

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

У вас имеется неограниченное количество монеток  $N$  заданных достоинств. Вам требуется определить, можно ли с их помощью разменять заданные  $K$  суммы денег.

### Формат входных данных

В первой строке задано число  $N$ , а далее во второй строке —  $N$  чисел, задающих достоинства монеток. В третьей строке задано число  $K$ , а далее в четвёртой —  $K$  чисел, определяющих размеры сумм. Все числа во входном файле натуральные (целые положительные) и не превосходят 1000.

### Формат выходных данных

В единственной строке  $K$  чисел — для каждой суммы 0, если её разменять нельзя, и 1, если можно.

### Пример

money.in	money.out
2	1 1 0 1 1
3 5	
5	
3 6 7 11 12	

## Задача С. Гвоздики

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

На прямой дощечке вбиты гвоздики. Любые два гвоздика можно соединить ниточкой. Требуется

соединить какие-то пары гвоздиков ниточками так, чтобы к каждому гвоздику была привязана хотя бы одна ниточка, а суммарная длина всех ниточек была минимальна.

### Формат входных данных

В первой строке входного файла записано число  $N$  — количество гвоздиков ( $2 \leq N \leq 100$ ). В следующей строке записано  $N$  чисел — координаты всех гвоздиков (неотрицательные целые числа, не превосходящие 10 000).

### Формат выходных данных

В выходной файл нужно вывести единственное число — минимальную суммарную длину всех ниточек.

### Пример

nails.in	nails.out
5	6
4 10 0 12 2	

## Задача Д. 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 входных данных

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 выходных данных

Выполните искомый максимальный вес.

### Примеры

knapsack.in	knapsack.out
10 3	9
1 4 8	
20 4	19
5 7 12 18	

## Задача Е. Наилучшее приближение

Имя входного файла: nearest.in  
Имя выходного файла: nearest.out  
Ограничение по времени: 5 секунд  
Ограничение по памяти: 64 мегабайта

Вам даны  $N$  целых чисел. Ваша задача — вставить ровно по одному знаку “+” или “-” между каждой парой соседних таким образом, чтобы сделать значение получившегося выражения максимально близким к заданному числу  $A$ .

### Формат входных данных

Первая строка входного файла содержит два целых числа:  $N$  ( $1 \leq N \leq 10\,000$ ) и  $A$ , которое по модулю не превосходит 10 000. Далее следуют  $N$  строк, в каждой из которых содержится ровно одно целое число  $X_i$ , не превосходящее по модулю 10 000. Кроме того,

гарантируется, что сумма абсолютных величин всех  $N$  чисел также не превосходит 10 000.

#### Формат выходных данных

В первой строке необходимо вывести значение получившегося выражения (которое должно быть настолько близко к  $A$ , насколько это возможно). Во второй строке необходимо вывести само выражение, дающее такое значение, в форме  $X_1[+|-]X_2[+|-]\dots X_{N-1}[+|-]X_N$ . Если оптимальных решений несколько, то разрешается выводить любое из них.

#### Пример

nearest.in	nearest.out
3 0	0
3	3+-2-1
-2	
1	

### Задача F. 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 входных данных

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 выходных данных

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

#### Примеры

lcs.in	lcs.out
3	2
1 2 3	
4	
2 1 3 5	

### Задача G. 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. В последующих строках должны быть выведены (по одному числу в каждой строке) все номера элементов исходной последовательности  $i_j$ , образующих искомую подпоследовательность. Номера выводятся в порядке возрастания. Если оптимальных решений несколько, разрешается выводить любое.

#### Пример

subseq.in	subseq.out
5	3
5 8	2
10 4 1	4
	5