

## Задача А. Строки в книге

Имя входного файла: a1.in  
Имя выходного файла: a1.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

В книге на одной странице помещается  $K$  строк. Таким образом, на 1-й странице печатаются строки с 1-й по  $K$ -ю, на второй - с  $(K + 1)$ -й по  $(2K)$ -ю и т.д. Напишите программу, которая по номеру строки в тексте определяет номер страницы, на которой будет напечатана эта строка, и порядковый номер этой строки на странице.

### Формат входного файла

Входной файл содержит число  $K$  - количество строк, которое печатается на странице, и число  $N$  - номер строки ( $1 \leq K \leq 200$ ,  $1 \leq N \leq 20000$ ).

### Формат выходного файла

В выходной файл выведите два числа - номер страницы, на которой будет напечатана эта строка и номер строки на странице.

### Примеры

a1.in	a1.out
50 1	1 1
20 25	2 5
15 43	3 13
1 1	1 1
20 10	1 10
20 20	1 20

## Задача В. Симметричная последовательность

Имя входного файла: b1.in  
Имя выходного файла: b1.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Последовательность чисел назовем *симметричной*, если она одинаково читается как слева направо, так и справа налево. Например, следующие последовательности являются симметричными:

1 2 3 4 5 4 3 2 1  
1 2 1 2 2 1 2 1

Вашей программе будет дана последовательность чисел. Требуется определить, какое минимальное количество и каких чисел надо приписать в конец этой последовательности, чтобы она стала симметричной.

### Формат входного файла

Во входном файле записано сначала число  $N$  - количество элементов исходной последовательности. Далее записано  $N$  чисел - элементы этой последовательности.  $1 \leq N \leq 100$ , элементы последовательности - натуральные числа от 1 до 9.

### Формат выходного файла

В выходной файл выведите сначала число  $M$  - минимальное количество элементов, которое надо дописать к последовательности, а потом  $M$  чисел (каждое - от 1 до 9) - числа, которые надо дописать к последовательности.

### Примеры

b1.in	b1.out
9 1 2 3 4 5 4 3 2 1	0
5 1 2 1 2 2	3 1 2 1
5 1 2 3 4 5	4 4 3 2 1

## Задача С. Кинотеатр

Имя входного файла: c1.in  
Имя выходного файла: c1.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

$X$  мальчиков и  $Y$  девочек пошли в кинотеатр и купили билеты на подряд идущие места в одном ряду. Напишите программу, которая выдаст, как нужно сесть мальчикам и девочкам, чтобы рядом с каждым мальчиком сидела хотя бы одна девочка, а рядом с каждой девочкой - хотя бы один мальчик.

### Формат входного файла

Два натуральных числа -  $X$  и  $Y$ , не превосходящие 100.

### Формат выходного файла

В выходной файл выведите какую-нибудь строку, в которой будет ровно  $X$  символов В (обозначающих мальчиков) и  $Y$  символов G (обозначающих девочек), удовлетворяющую условию задачи. Пробелы между символами выводить не нужно. Если рассадить мальчиков и девочек согласно условию задачи невозможно, в выходной файл должна быть записана строка NO SOLUTION.

### Примеры

c1.in	c1.out
5 5	BGBGBGBGBG
5 3	BGBGBBGB
100 1	NO SOLUTION

## Задача D. Количество слов

Имя входного файла: d1.in  
Имя выходного файла: d1.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Во входном файле записана строка текста, в которой могут встречаться:

- прописные и строчные (т.е. большие и маленькие) латинские буквы,
- пробелы,
- знаки препинания: точка, запятая, восклицательный и вопросительный знак,
- символ ”-”, обозначающий в некоторых случаях тире, а в некоторых - дефис.

Слово - это последовательность подряд идущих латинских букв и знаков дефис, ограниченная с обоих концов. В качестве ограничителей могут выступать начало строки, конец строки, пробел, знак препинания, тире. Тире отличается от дефиса тем, что слева и справа от знака дефис пишутся буквы, а хотя бы с одной стороны от тире

идет либо начало строки, либо конец строки, либо пробел, либо какой-либо знак препинания, либо еще одно тире.

Напишите программу, определяющую, сколько слов в данной строке текста.

### Формат входного файла

Строка длиной не более 200 символов.

### Формат выходного файла

Одно число - количество слов, которые содержатся в исходной строке.

### Примеры

d1.in	d1.out
Hello , world!	2
www.olympiads.ru	3
Gyro-compass - this is a ...	4

## Задача Е. Метро

Имя входного файла: e1.in  
Имя выходного файла: e1.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Метрополитен состоит из нескольких линий метро. Все станции метро в городе пронумерованы натуральными числами от 1 до  $N$ . На каждой линии расположено несколько станций. Если одна и та же станция расположена сразу на нескольких линиях, то она является станцией пересадки и на этой станции можно пересесть с любой линии, которая через нее проходит, на любую другую (опять же проходящую через нее).

Напишите программу, которая по данному вам описанию метрополитена определит, с каким минимальным числом пересадок можно добраться со станции  $A$  на станцию  $B$ . Если данный метрополитен не соединяет все линии в одну систему, то может так получиться, что со станции  $A$  на станцию  $B$  добраться невозможно, в этом случае ваша программа должна это определить.

### Формат входного файла

Во входном файле записано сначала число  $N$  - количество станций метро в городе ( $2 \leq N \leq 100$ ). Далее записано число  $M$  - количество линий метро ( $1 \leq M \leq 20$ ). Далее идет описание  $M$  линий. Описание каждой линии состоит из числа  $P_i$  - количество станций на этой линии ( $2 \leq P_i \leq 50$ ) и  $P_i$  чисел, задающих номера станций, через которые проходит линия (ни через какую станцию линия не проходит дважды).

В конце файла записаны два различных: числа  $A$  - номер начальной станции, и  $B$  - номер станции, на которую нам нужно попасть. При этом если через станцию  $A$  проходит несколько линий, то мы можем спуститься на любую из них. Так же если через станцию  $B$  проходит несколько линий, то нам не важно, по какой линии мы приедем.

### Формат выходного файла

В выходной файл выведите минимальное количество пересадок, которое нам понадобится. Если добраться со станции  $A$  на станцию  $B$  невозможно, выведите в выходной файл одно число -1 (минус один).

### Примеры

e1.in	e1.out
5 2 4 1 2 3 4 2 5 3 3 1	0
5 5 2 1 2 2 1 3 2 2 3 2 3 4 2 4 5 1 5	2
10 2 6 1 3 5 7 4 9 6 2 4 6 8 10 7 3 8	1
4 2 2 1 2 2 3 4 1 3	-1

## Задача F. A+B

Имя входного файла: f1.in  
Имя выходного файла: f1.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Даны два числа, требуется найти их сумму.

### Формат входного файла

Два целых числа  $A$  и  $B$  ( $0 \leq A, B \leq 2^{63} - 1$ ) разделённых пробелом или переводом строки.

### Формат выходного файла

Вывести  $A + B$  (гарантируется, что  $A + B \leq 2^{63} - 1$ ).

### Примеры

f1.in	f1.out
2 3	5
489231523 8234123	497465646
9223372036854775806 1	9223372036854775807

## Задача G. Переверните строку!

Имя входного файла: g1.in  
Имя выходного файла: g1.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Прочитать строку, перевернуть ее и вывести.

### Формат входного файла

Входной файл содержит одну строку  $S$ .

### Формат выходного файла

В первой строчке надо вывести перевернутую строку  $S$ .

### Примеры

g1.in	g1.out
Переверните строку!	!укортс етинревереп

## Задача Н. Списки

Имя входного файла: `h1.in`  
Имя выходного файла: `h1.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Дан список. Нужно эффективно реализовать операции добавления в него элемента и удаления элемента из него. Изначально список пустой.

### Формат входного файла

Входной файл содержит на каждой строке по одной из команд:

- **INSERT** «значение элемента» - Добавление в конец.
- **DELETE** «номер элемента с начала» - Удаление из начала.

### Формат выходного файла

В выходной файл нужно вывести удаленные элементы, по одному на каждой строке.

### Примеры

h1.in	h1.out
INSERT 10	2
INSERT 9	3
INSERT 8	9
INSERT 7	1
INSERT 6	10
INSERT 5	
INSERT 4	
INSERT 3	
INSERT 2	
INSERT 1	
DELETE 9	
DELETE 8	
DELETE 2	
DELETE 7	
DELETE 1	