

## Задача А Москва-сортировочная

Имя входного файла: a.in  
Имя выходного файла: a.out  
Максимальное время работы на одном тесте: 2 секунды  
Максимальный объем используемой памяти: 64 мегабайта

Ежедневно диспетчеру железнодорожной станции "Москва-Сортировочная" приходится переставлять вагоны во многих поездах, чтобы они шли в заданном порядке. Для этого диспетчер может расцепить пришедший на станцию состав в произвольных местах и переставить образовавшиеся сцепки из одного или нескольких вагонов в произвольном порядке. Порядок вагонов в одной сцепке менять нельзя, также нельзя развернуть всю сцепку так, чтобы последний вагон в сцепке оказался первым в ней.

Диспетчер просит вашей помощи в определении того, какое минимальное число соединений между вагонами необходимо расцепить, чтобы переставить вагоны в составе в требуемом порядке.

### Формат входных данных

В первой строке входного файла содержится целое число  $N$ , ( $1 \leq N \leq 100$ ). Во второй строке содержится перестановка натуральных чисел от 1 до  $N$  (то есть все натуральные числа от 1 до  $N$  в некотором порядке). Числа разделяются одним пробелом. Эта перестановка задает номера вагонов в приходящем на станцию составе. Требуется, чтобы в уходящем со станции составе вагоны шли в порядке их номеров.

### Формат выходных данных

Программа должна записать в выходной файл единственное целое число, равное минимальному количеству соединений между вагонами, которое нужно расцепить в данном составе, чтобы их можно было переставить по порядку.

### Примеры

a.in	a.out
4 3 1 2 4	2
5 5 4 3 2 1	4
2 1 2	0

## Задача В Кафе

Имя входного файла: `b.in`  
Имя выходного файла: `b.out`  
Максимальное время работы на одном тесте: 2 секунды  
Максимальный объем используемой памяти: 64 мегабайта

Около Петинского университета недавно открылось новое кафе, в котором действует следующая система скидок: при каждой покупке более чем на 100 рублей покупатель получает купон, дающий право на один бесплатный обед (при покупке на сумму 100 рублей и меньше такой купон покупатель не получает).

Однажды Пете на глаза попался преискусант на ближайшие  $N$  дней. Внимательно его изучив, он решил, что будет обедать в этом кафе все  $N$  дней, причем каждый день он будет покупать в кафе ровно один обед. Однако стипендия у Пети небольшая, и поэтому он хочет по максимуму использовать предоставляемую систему скидок так, чтобы его суммарные затраты были минимальны. Требуется найти минимально возможную суммарную стоимость обедов и номера дней, в которые Пете следует воспользоваться купонами.

### Формат входных данных

В первой строке входного файла записано целое число  $N$  ( $0 \leq N \leq 100$ ). В каждой из последующих  $N$  строк записано одно целое число, обозначающее стоимость обеда в рублях на соответствующий день. Стоимость — неотрицательное целое число, не превосходящее 300.

### Формат выходных данных

В первой строке выведите минимальную возможную суммарную стоимость обедов. Во второй строке выведите два числа  $K_1$  и  $K_2$  — количество купонов, которые останутся неиспользованными у Пети после этих  $N$  дней и количество использованных им купонов соответственно.

В последующих  $K_2$  строках выведите в возрастающем порядке номера дней, когда Пете следует воспользоваться купонами. Если существует несколько решений с минимальной суммарной стоимостью, то выведите то из них, в котором значение  $K_1$  максимально (на случай, если Петя когда-нибудь ещё решит заглянуть в это кафе). Если таких решений несколько, выведите любое из них.

### Пример

<code>b.in</code>	<code>b.out</code>
5	235
35	0 1
40	5
101	
59	
63	

## Задача С EuroEnglish

Имя входного файла:	c.in
Имя выходного файла:	c.out
Максимальное время работы на одном тесте:	2 секунды
Максимальный объем используемой памяти:	64 мегабайта

Европейская комиссия планирует принять решение о том, что официальным языком Евросоюза станет английский. Был также разработан план упрощения английской письменности, который планируется реализовать за четыре года.

Первоочередной задачей будет избавление от буквы **c**, которая в сочетаниях **ci** и **ce** будет изменяться на **s**, в сочетании **ck** — опускаться, а в остальных случаях заменяться на **k**. При этом все замены будут производиться в строгом порядке слева направо. То есть, например, в слове *success* сначала первая из двух букв *c* заменится на *k*, а затем вторая — на *s*, то есть получится *suksess*. А слово *сск* превратится в *kk*.

На второй год из английских слов изымут все удвоенные буквы: **ee** изменят на **i**, **oo** - на **u**, а в остальных комбинациях будут просто писать одну букву вместо двух одинаковых. Такие замены также будут делать строго в порядке слева направо. Так, слово *ooo* превратится в *uo*, а *oou* — просто в *u* (в нем сначала *oo* заменится на *u*, а затем *uu* — на *u*), слово *iee* превратится в *i* (в нем сначала *ee* заменится на *i*, а затем *ii* — на *i*).

На третий год на конце слова станут опускать букву **e**, если эта буква не является единственной буквой в слове.

Наконец, завершением реформы станет отмена артиклей (в английском языке три артикля: **a**, **an** и **the**). При этом удаляться эти артикли будут только тогда, когда они в исходном тексте были словами *a*, *an*, *the*. То есть, например, текст *the table* после реформ первых трех лет превратиться в *th tabl*, а после реформы четвертого года — просто в *tabl*. А слово *aaaaa* после реформы первых лет станет словом *a*, но поскольку изначально оно не было словом *a* (артиклем), то оно в итоге так и останется словом *a*.

Напишите программу, которая будет переводить классический английский текст на Евроинглиш.

### Формат входных данных

Во входном файле записана одна строка текста, состоящая не более чем из 200 символов: латинских строчных и заглавных букв, пробелов и знаков препинания (в тексте могут встречаться: точка, запятая, вопросительный и восклицательный знаки, двоеточие, тире, точка с запятой, открывающаяся и закрывающаяся скобки, апострофы, кавычки). Заглавные буквы могут встречаться только в начале слова. Нигде подряд не могут стоять два пробела. В начале и в конце строки не может стоять пробел. Слова отделяются друг от друга пробелами и/или знаками препинания.

### Формат выходных данных

В выходной файл нужно выдать преобразованную строку с учетом следующих требований:

- начинаться с заглавной буквы должны те и только те слова, которые начинались с заглавной буквы в исходном тексте;
- не должно встречаться двух пробелов подряд;
- пробелы между словами и знаками препинания должны остаться там и только там, где они были в исходной строке, в начале и в конце строки пробелов быть не должно.

### Примеры

c.in	c.out
cacao and coffee	kakao and kofi
Cinderella! Where Is The Dress???	Sinderela! Wher Is Dres???
'A' is a letter	' ' is leter
!!!Hello!!!A-the-"word"	!!!Helo!!!--"word"
Aaaa then the ckckck	A then k
"A"-the an	" "_
A the an	
success	sukses

## Задача D D++

Имя входного файла: d.in  
Имя выходного файла: d.out  
Максимальное время работы на одном тесте: 2 секунды  
Максимальный объем используемой памяти: 64 мегабайта

Мальчик Кирилл недавно придумал свой язык программирования. Он назвал его D++.

В D++ существует только один массив и 26 переменных. Переменные называются маленькими латинскими буквами от a до z. Есть 4 различных варианта операции присваивания (здесь и далее *переменная* — это маленькая латинская буква, задающая имя переменной, а *индекс* — натуральное число, задающее индекс элемента в массиве):

$переменная1 \leftarrow переменная2$

Значение переменной *переменная1* становится равным значению переменной *переменная2*;

$переменная \leftarrow индекс$

Значение переменной *переменная* становится равным значению элемента массива с индексом *индекс*;

$индекс \leftarrow переменная$

Значение элемента массива с индексом *индекс* становится равным значению переменной *переменная*;

$индекс1 \leftarrow индекс2$

Значение элемента массива с индексом *индекс1* становится равным значению элемента массива с индексом *индекс2*.

Других операций в языке D++ нет. В одной строке программы на D++ может быть записана только одна операция. В программе не может быть пустых строк, но могут быть незначащие пробелы.

Дана последовательность из  $N$  натуральных чисел. Все числа различны и не превосходят  $N$ . Будем считать, что эта последовательность записана в массиве языка D++ начиная с элемента с номером 1. Требуется написать программу на D++, которая упорядочит массив за наименьшее количество операций.

После выполнения программы последовательность также должна быть записана в массиве начиная с элемента с номером 1. Программа не должна использовать элементы массива с номерами меньше 1, а также с номерами больше  $N$ .

### Формат входных данных

В первой строке входного файла содержится единственное число  $N$  ( $1 \leq N \leq 10000$ ). Во второй строке содержится  $N$  натуральных чисел — последовательность, записанная в массиве.

### Формат выходных данных

Выходной файл должен содержать программу на D++, которая сортирует данную последовательность, выполняя наименьшее число операций присваивания.

### Пример

d.in	d.out
2	x<-1
2 1	1<-2
	2<-x

## Задача E Скобки

Имя входного файла: e.in  
Имя выходного файла: e.out  
Максимальное время работы на одном тесте: 2 секунды  
Максимальный объем используемой памяти: 64 мегабайта

Назовем строку  $S$  правильной скобочной последовательностью, если она состоит только из символов '{', '}', '[', ']', '(', ')' и выполнено хотя бы одно из следующих трех условий:

- 1)  $S$  — пустая строка;
- 2)  $S$  можно представить в виде  $S=S_1+S_2+S_3+\dots+S_N$  ( $N>1$ ), где  $S_i$  — непустые правильные скобочные последовательности, а знак "+" обозначает конкатенацию (приписывание) строк;
- 3)  $S$  можно представить в виде  $S='{'+C+'}'$  или  $S='['+C+'']'$  или  $S='('+C+')'$ , где  $C$  является правильной скобочной последовательностью.

Дана строка, состоящая только из символов '{', '}', '[', ']', '(', ')'. Требуется определить, какое минимальное количество символов надо вставить в эту строку для того, чтобы она стала правильной скобочной последовательностью.

### Формат входных данных

В первой строке входного файла записана строка, состоящая только из символов '{', '}', '[', ']', '(', ')'. Длина строки не превосходит 100 символов.

### Формат выходных данных

Вывести в первую строку выходного файла единственное неотрицательное целое число — ответ на поставленную задачу.

### Примеры

e.in	e.out
{ ( ) }	2
( [ { } ] )	0

## Задача F Двойкие числа

Имя входного файла: f.in  
Имя выходного файла: f.out  
Максимальное время работы на одном тесте: 2 секунды  
Максимальный объем используемой памяти: 64 мегабайта

Натуральное число называется *двойким*, если в его десятичной записи встречается не более двух различных цифр. Например, числа 3, 23, 33, 100, 12121 — двойкие, а числа 123 и 9980 — нет.

Для заданного натурального числа  $N$  требуется найти ближайшее к нему двойкое число (если таких чисел два — любое из них).

### Формат входных данных

Во входном файле записано одно натуральное число  $N$ , не превосходящее 30 000.

### Формат выходных данных

В выходной файл требуется выдать единственное число — ближайшее двойкое к числу  $N$ .

### Примеры

f.in	f.out
123	122
2012	2020
11111	11111

## Задача G ООО

Имя входного файла: g.in  
Имя выходного файла: g.out  
Максимальное время работы на одном тесте: 2 секунды  
Максимальный объем используемой памяти: 64 мегабайта

Одна серьезная организация (ОСО) решила построить очень охраняемый объект (ООО). Для этого она нашла на пустыре два одиноко стоящих прожектора. Каждый из прожекторов освещает какой-то угол, меньший 180 градусов. ОСО хочет, чтобы весь ООО был освещен обоими прожекторами.

Положения прожекторов и освещаемые ими углы заданы. Требуется найти максимальную площадь ООО, который удастся построить.

Возможна ситуация, что ООО может иметь сколь угодно большую площадь. Однако если это не так, то гарантируется, что в этом случае максимальная возможная площадь не будет превышать  $10^{15}$ .

### Формат входных данных

Во входном файле последовательно заданы описания двух прожекторов. Каждый из прожекторов описывается следующим образом: сначала идут координаты прожектора, а затем — координаты двух точек: по одной точке на каждой из сторон угла, освещаемого им. Все числа целые, не превосходящие по модулю 10000.

### Формат выходных данных

Выведите максимально возможную площадь ООО с точностью не менее 3-х знаков после десятичной точки. Если ООО построить не удастся, выведите 0. Если ООО может иметь любую сколь угодно большую площадь, выведите число  $-1$ .

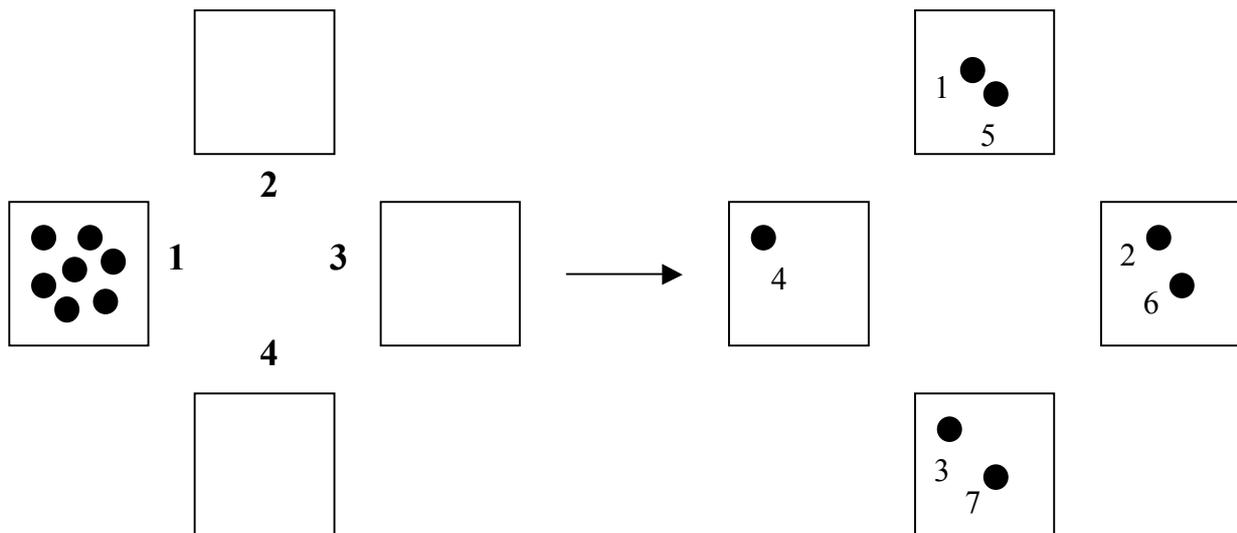
### Примеры

g.in	g.out
0 1 2 3 3 2 3 0 3 3 5 2	-1
0 1 2 3 3 2 3 0 3 3 -3 2	3.000
0 1 2 3 3 2 3 0 3 -2 -3 2	0

## Задача Н Калах

Имя входного файла: h.in  
Имя выходного файла: h.out  
Максимальное время работы на одном тесте: 2 секунды  
Максимальный объем используемой памяти: 64 мегабайта

Для игры в калах используют несколько коробочек, расставленных по кругу, в которых лежат шарики. Ход осуществляется следующим образом. Берутся все шарики из одной коробочки, и начинают раскладываться по одному в коробочки подряд начиная со следующей по часовой стрелке. Если шариков больше, чем коробочек, то процесс продолжается (шарики раскладываются по второму кругу, по третьему и т.д.), пока не будут разложены все шарики. В коробочку, из которой взяли шарики, их тоже кладут. Пример одного хода приведен на рисунке. Справа шарики пронумерованы в том порядке, в котором они раскладывались по коробочкам.



Петя, тренируясь перед соревнованиями, разложил шарики по коробочкам произвольным образом, и стал делать произвольные ходы. После каждого хода он записывал номер коробочки, в которую попадал последний шарик. В некоторый момент он решил восстановить начальную конфигурацию по конечной и по тем записям, которые он делал. Напишите программу, которая поможет ему в этом.

### Формат входных данных

В первой строке входного файла записано два натуральных числа:  $N \leq 100$  — количество коробочек и  $M \leq 100$  — количество сделанных Петей ходов. Коробочки пронумерованы последовательно по часовой стрелке числами от 1 до  $N$ . В следующих  $N$  строках записано количество шариков в первой, второй, ...,  $N$ -ой коробочках в конечной конфигурации (по одному числу в каждой строке). В следующих  $M$  строках записаны номера коробочек, в которые был положен последний шарик на первом, втором, ...,  $M$ -ом ходу соответственно (по одному числу в каждой строке). Общее количество шариков не превосходит  $10^9$ .

### Формат выходных данных

В выходной файл требуется вывести  $N$  чисел: первоначальное количество шариков в первой, второй, ...,  $N$ -ой коробочках.

### Примеры

Первый пример описывает ситуацию, изображенную на рисунке.

h.in	h.out
4 1	7
1	0
2	0
2	0
2	
4	

h.in	h.out
2 2	1
1	1
1	
2	
2	

## Задача I Сортировка масс

Имя входного файла: `i.in`  
Имя выходного файла: `i.out`  
Максимальное время работы на одном тесте: 2 секунды  
Максимальный объем используемой памяти: 64 мегабайта

Как известно, Россия является одним из ведущих экспортеров нефти. Разные страны мира, от достаточно больших до сравнительно маленьких, нуждаются в этой нефти как в воздухе. В ее состав в больших количествах входят ароматические углеводороды, которые обуславливают ее высокое качество. Доставка нефти в пункт назначения осуществляется с помощью нефтепровода. Считается, что количество нефти, отправленное в страну назначения, равно количеству полученной нефти. На самом деле это, конечно, не так. Как и многое другое, нефть воруют некоторые несознательные личности. Причем неофициально считается, что больше нефти воруют в нефтепроводах тех стран, куда нефти посылается больше (может быть, несознательные личности считают, что приносят, таким образом, меньше ущерба, кто знает...). Официальное руководство компании РусскаяНефть решило узнать, правдивый это слух или нет, чтобы усилить (а может просто установить) охрану на тех нефтепроводах, где больше всего воруют нефть.

Для этого им нужно отсортировать нефтепроводы по количеству нефти, которая протекает в направлении какой-то страны за сутки. У компании РусскаяНефть, как и у любой уважающей себя компании, есть несколько штатных программистов, и руководство предложило им решить эту, в сущности, нетрудную задачу. Но программистов поставило в тупик то, что данные о количестве нефти представлены в разных единицах измерения (начиная от граммов и заканчивая тоннами).

Поэтому они решили найти человека, который был бы в силах решить эту задачу за них, и обещают взять его на работу в эту перспективную и процветающую компанию. Решите задачу, и, кто знает, может, повезет именно Вам?

### Формат входных данных

В первой строке входного файла находится целое число  $N$  ( $1 \leq N \leq 1000$ ) — количество нефтепроводов. В каждой из следующих  $N$  строк находится количество (точнее — масса) нефти, транспортированной по соответствующему нефтепроводу за сутки, по одному в строке. Масса нефти задана целым числом от 1 до 10000 с указанием соответствующей единицы измерения. Число и единица измерения разделены ровно одним пробелом. Единица измерения задается одной из трех букв: **g** (граммы), **p** (пуды), **t** (тонны), причем перед этой буквой может стоять одна из приставок: **m** (милли-), **k** (кило-), **M** (мега-), **G** (гига-). Напомним, что эти приставки обозначают умножение единицы измерения на  $10^{-3}$ ,  $10^3$ ,  $10^6$  и  $10^9$  соответственно. 1 пуд = 16380 граммов, 1 тонна =  $10^6$  граммов.

### Формат выходных данных

В выходной файл выведите  $N$  строк, в которых должны быть записаны массы нефти в порядке неубывания. Каждая строка должна описывать массу нефти в одном из нефтепроводов.

Масса должна быть описана согласно формату входного файла. Единицы измерения масс в выходном файле не обязаны соответствовать единицам измерения масс во входном файле, главное, чтобы массы были равны исходным. Все числа в выходном файле, также как и во входном, должны быть натуральными и не превышать 10000.

### Пример

<code>i.in</code>	<code>i.out</code>
5	32 mg
234 g	234 g
4576 mp	4576 mp
2 t	2 t
32 mg	2 t
2 Mg	