Задача A. Кёрлинг

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

Кёрлинг (англ. curling, от англо-шотл. curr) — командная спортивная игра на ледяной площадке.

Википедия

Как известно, с 1998 года кёрлинг признан олимпийским видом спорта. Во время просмотра Зимних Олимпийских игр в Ванкувере премьер-министр Флатландии очень заинтересовался этим видом спорта и решил, что на олимпиаде 2014 года его страна обязательно должна участвовать в соревнованиях по кёрлингу, причем, как в женских, так и в мужских.

Позже выяснилось, что в стране пока что нет ни одной площадки для этой игры. Премьер-министр решил подойти к этому вопросу серьезно и приказал построить как минимум по одной площадке в каждом крупном городе. Первой начала строиться площадка в самом центре Сен-Флатбурга (столицы Флатландии). Она уже почти построена, для показа текущего счета уже куплено самое большое табло, но для него пока не написано программное обеспечение. Вам поручено написать программу, которая будет считать текущий счет партии в кёрлинг.

В кёрлинг играют две команды. Каждая игра состоит из десяти партий, который называют *эндами*. Если десять эндов заканчиваются вничью, то назначается дополнительный одиннадцатый экстра-энд. В каждом энде команды по очереди «играют» камни. В течение энда некоторые камни могут выйти из игры, в этом случае их убирают с площадки.

После того, как каждой командой сыграны все восемь камней, производится подсчет очков в энде. Команда, чей камень оказался ближе всего к центру дома, считается выигравшей энд. Она получает по одному очку за каждый камень, оказавшийся строго ближе к центру, чем любой из камней противника. Текущий счет определяется как сумма очков по каждому из завершенных эндов.

В игре по керлингу прошло несколько эндов. Для каждого из них задано конечное расположение камней, оставшихся в игре. Необходимо определить текущий счет.

Формат входного файла

Первая строка входного файла содержит единственое число n ($1 \le n \le 11$) — количество завершенных эндов. Далее следуют описания каждого из эндов в следующем формате. Первая строка описания содержит два числа m и k ($1 \le m, k \le 8$) — количество оставшихся в игре камней, принадлежащих соответственно первой и второй командам. Следующие m строк содержат по два целых числа, не превосходящих 1000 по модулю — координаты соответствующего камня первой команды. Следующие k строк также содержат по два целых числа, не превосходящих 1000 по модулю — координаты соответствующего камня второй команды. В одном энде никакие два камня не расположены в одной точке.

Центр дома находится в начале координат — точке с координатами $(0, 0)$.

Формат выходного файла

В выходной файл выведите текущий счет в партии в формате `a:b`, где `a` — текущее количество очков у первой команды, `b` — текущее количество очков у второй команды.

Примеры

<table>
<tr><th>curling.in</th><th>curling.out</th></tr>
<tr><td>2
2 2
1 1
2 2
0 1
-1 -1
2 2
0 0
1 0
-1 -1
100 100</td><td>2:1</td></tr>
<tr><td>1
1 1
10 0
0 10</td><td>0:0</td></tr>
</table>

Задача B. Текст

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

Давным-давно, в далекой-далекой галактике учились в одной школе два закадычных друга — Вася и Петя. Они были верными товарищами, не раз выручавшими друг друга в трудную минуту из самых безвыходных ситуаций. Однако речь сейчас пойдет не об этом, а о редчайшем случае — ссоре между двумя героями нашего повествования.

В конце седьмого класса Вася увлекся программированием и написал свой текстовый редактор. Естественно, Петя тут же захотел его испытать. Несложно представить насколько велико было его разочарование, когда обнаружилось, что Васина программа корректно работает только при использовании экрана с тем же разрешением, как и у него дома. Вася оправдывал это тем, что оптимальный вывод текста на экран — штука сложная, поэтому универсальным образом сделать это невозможно. Петя же заявил, что хоть программист он и никудышный, но легко решит эту задачу.

К сожалению, программирует Петя действительно из рук вон плохо, поэтому он просит вас помочь ему в написании решения.

На вход дан текст. Назовем *словом* последовательность символов, ограниченную пробелами, началом или концом текста. Обратите внимание, что в данной задаче знаки препинания считаются частью слова. Требуется разбить текст на строки так, чтобы длина каждой из них была не более k символов, при этом их общее количество было минимальным. Порядок слов и сами слова менять запрещено.

Формат входного файла

Первая строка входного файла содержит натуральное число k — максимально допустимая длина строки ($1 \le k \le 100$). Вторая строка входного файла содержит текст, который необходимо вывести. Текст состоит из латинских букв, цифр, пробелов и символов «,» (запятая), «.» (точка), «!» (восклицательный знак) и «?» (вопросительный знак). Размер входного файла не превышает 50000 байтов.

Формат выходного файла

Выведите заданный во входном файле текст так, чтобы длина каждой строки была не более k символов, а количество строк было минимально возможным. Гарантируется, что задача имеет решение. В случае если решение не единственно, выведите любое из них.

Слова в выходном файле должны быть отделены друг от друга пробелами и/или переводами строк.

Примеры

text.in	text.out
`22` `This.....is.a.sample.text!`	`This.is.a.sample.text!`
`12` `This.....is.a.sample.text!`	`This.is..a` `sample.text!`

Для наглядности в примерах пробелы заменены символом «.».

Задача C. Календарь

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

Однажды крупный марсианский магнат Аман Робрамович решил заказать в марсианской типографии «Флатланд-Печать» серию календарей с собственным профилем на каждой странице. Но так как Великий Финансовый Кризис не миновал и Марса, в типографии осталось недостаточное для качественной печати количество чернил, поэтому даты было решено печатать по старинке, на печатной машинке.

Достав из запасов печатную машинку «Лысач», работники типографии обнаружили, что некоторые головки сломаны. Напомним, что на печатных машинках, чтобы напечатать одну цифру требуется исправность соответствующей головки (например, чтобы напечатать цифру «4» необходимо, чтобы головка с цифрой «4» была исправна).

Работники типографии хотят узнать, в каком количестве дат им придется дописывать некоторые цифры карандашом.

Формат входного файла

В первой строке входного файла находится два числа n — количество месяцев на марсе и k — число сломанных головок в печатной машинке ($1 \le n \le 100000$, $0 \le k \le 10$). Во второй строке находится n чисел a_1, a_2, …, a_n — количество дней в месяцах ($1 \le a_i \le 100000$). В следующей строке находится k различных чисел — цифры которые нельзя напечатать.

Формат выходного файла

В выходной файл выведите количество дат, цифры в которых придется дописывать работникам типографии.

Примеры

calendar.in	calendar.out
12 2 31 28 31 30 31 30 31 31 30 31 30 31 0 3	78
2 2 5 15 0 1	8

Вид календаря до исправления мастерами типографии для первого примера показан на следующей странице.

Январь

Пн	Вт	Ср	Чт	Пт	Сб	Вс
				1	2	
4	5	6	7	8	9	1
11	12	1	14	15	16	17
18	19	2	21	22	2	24
25	26	27	28	29		1

Февраль

Пн	Вт	Ср	Чт	Пт	Сб	Вс
1	2		4	5	6	7
8	9	1	11	12	1	14
15	16	17	18	19	2	21
22	2	24	25	26	27	28

Март

Пн	Вт	Ср	Чт	Пт	Сб	Вс
1	2		4	5	6	7
8	9	1	11	12	1	14
15	16	17	18	19	2	21
22	2	24	25	26	27	28
29		1				

Апрель

Пн	Вт	Ср	Чт	Пт	Сб	Вс
			1	2		4
5	6	7	8	9	1	11
12	1	14	15	16	17	18
19	2	21	22	2	24	25
26	27	28	29			

Май

Пн	Вт	Ср	Чт	Пт	Сб	Вс
					1	2
	4	5	6	7	8	9
1	11	12	1	14	15	16
17	18	19	2	21	22	2
24	25	26	27	28	29	
1						

Июнь

Пн	Вт	Ср	Чт	Пт	Сб	Вс
	1	2		4	5	6
7	8	9	1	11	12	1
14	15	16	17	18	19	2
21	22	2	24	25	26	27
28	29					

Июль

Пн	Вт	Ср	Чт	Пт	Сб	Вс
			1	2		4
5	6	7	8	9	1	11
12	1	14	15	16	17	18
19	2	21	22	2	24	25
26	27	28	29		1	

Август

Пн	Вт	Ср	Чт	Пт	Сб	Вс
						1
2		4	5	6	7	8
9	1	11	12	1	14	15
16	17	18	19	2	21	22
2	24	25	26	27	28	29
	1					

Сентябрь

Пн	Вт	Ср	Чт	Пт	Сб	Вс
		1	2		4	5
6	7	8	9	1	11	12
1	14	15	16	17	18	19
2	21	22	2	24	25	26
27	28	29				

Октябрь

Пн	Вт	Ср	Чт	Пт	Сб	Вс
				1	2	
4	5	6	7	8	9	1
11	12	1	14	15	16	17
18	19	2	21	22	2	24
25	26	27	28	29		1

Ноябрь

Пн	Вт	Ср	Чт	Пт	Сб	Вс
1	2		4	5	6	7
8	9	1	11	12	1	14
15	16	17	18	19	2	21
22	2	24	25	26	27	28
29						

Декабрь

Пн	Вт	Ср	Чт	Пт	Сб	Вс
		1	2		4	5
6	7	8	9	1	11	12
1	14	15	16	17	18	19
2	21	22	2	24	25	26
27	28	29		1		

Задача D. Поддеревья

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

Дерево — это связный неориентированный граф без циклов. Поддеревом дерева T называется его связный подграф.

В зависимости от структуры дерева у него может быть несколько поддеревьев. Например, у дерева-цепочки из n вершин $n(n+1)/2$ поддеревьев, а у дерева-звезды ($n-1$ вершина связаны с одной центральной) количество поддеревьев равно $2^{n-1}+n-1$.

Аналогично может быть различное количество способов выделить в дереве два непересекающихся по вершинам поддерева, три непересекающихся поддерева, и т. д.

Наконец, вне зависимости от структуры дерева есть $2n-1$ способ выбрать $n-1$ непересекающееся поддерево и ровно один способ выбрать n непересекающихся поддеревьев (просто n изолированных вершин).

По заданному дереву с n вершинами выведите для каждого k от 1 до n количество способов выбрать k непересекающихся по вершинам поддеревьев данного дерева.

Формат входного файла

Первая строка входного файла содержит n ($2 \le n \le 100$). Следующие $n-1$ строка описывают ребра дерева.

Формат выходного файла

Выведите n целых чисел: для каждого k от 1 до n выведите количество способов выбрать k непересекающихся по вершинам поддеревьев заданного дерева. Выводите каждое число по модулю 10^9.

Примеры

subtrees.in	subtrees.out
5 1 2 2 3 3 4 4 5	15 35 28 9 1
5 1 2 1 3 1 4 1 5	20 38 28 9 1