

## Задача А. Шифрование

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

Для передачи закодированных сообщений был предложен следующий шифр. Из первоначального текста, состоящего из слов (последовательностей прописных латинских букв), записанных через пробел, сначала убираются все пробелы, затем каждой из 26 букв латинского алфавита ставится в соответствие одна из 10 цифр, и каждая буква заменяется на соответствующую ей цифру.

Также известен словарь, в который входят все слова в первоначальном тексте. По заданной таблице соответствия цифр буквам, словарю и закодированному фрагменту требуется узнать, сколько существует упорядоченных наборов слов из словаря, при зашифровке которых получится данный фрагмент. Так, например, для заданных в первом фрагменте таблицы соответствия, словаря и закодированного фрагмента корректными расшифровками являются тексты «АВ» и «А Z».

### Формат входного файла

В первой строке входного файла содержится количество тестовых примеров  $T$ ,  $1 \leq T \leq 20$ .

Далее следуют  $T$  тестовых примеров. В первой строке каждого примера задана строка из 26 цифр — таблица соответствия, в которой первая цифра соответствует букве «А», вторая — «В» и так далее. В следующей строке идёт  $1 \leq N \leq 2000$  — количество слов в словаре, в следующих  $N$  строках идут сами слова, по одному на строку. Слова состоят из прописных латинских букв, длина каждого слова не превышает 20000 знаков. Завершает тестовый пример закодированный фрагмент — строка, составленная из не более, чем 2000 цифр 0...9.

### Формат выходного файла

Для каждого тестового примера в отдельной строке выведите ответ — количество способов, которым может быть расшифрован соответствующий фрагмент. Гарантируется, что ответ не превосходит  $10^9$ .

### Пример

<code>crypt.in</code>	<code>crypt.out</code>
2	2
0111111111111111111111111111	1
3	
AB	
A	
Z	
01	
0144444444444444444444444444	
2	
REJECTED	
ACCEPTED	
44444444	

## Задача В. Игра

Имя входного файла: `game.in`  
Имя выходного файла: `game.out`  
Ограничение по времени: 12 секунд  
Ограничение по памяти: 64 Мб

По инициативе ректора Тбилисского государственного университета академика Векуа была организована археологическая экспедиция, целью которой было исследование небольшого поселения, обнаруженного на берегу Чёрного моря. Кроме обычных для поселений того времени остатков построек и глиняной посуды археологами были обнаружены странные рисунки, напоминавшие головоломку или какую-то игру.

Изучение рисунков и сопровождавших их надписей позволило выяснить правила игры. На прямоугольном клетчатом поле  $M \times N$  расставлено  $K$  фишек так, что в каждой клетке расположено не более одной фишки. Задача игрока — снять как можно больше фишек с доски. Первым ходом игрок может снять любую фишку, каждая последующая снятая фишка должна быть соседней с предыдущей снятой (две фишки считаются соседними, если они расположены в клетках, имеющих общую сторону). В случае, если ход в соответствии с правилами сделать невозможно, игра заканчивается и подсчитываются снятые фишки.

Ваша задача — по заданной исходной позиции определить, какой максимальный результат можно получить в данной игре.

### Формат входного файла

В первой строке входного файла содержится количество тестовых примеров  $T$ ,  $1 \leq T \leq 20$ .

Далее следуют  $T$  тестовых примеров. В первой строке каждого примера заданы два целых положительных числа  $N$  и  $M$ , не превосходящие 100 — количество строк и столбцов игрового поля. Далее следуют  $N$  строк по  $M$  символов — описание поля, при этом точками обозначены поля, не занятые фишками, а строчными латинскими буквами  $o$  — поля, занятые фишками.

### Формат выходного файла

В каждой из  $T$  строк выходного файла выведите максимальное число фишек, которое можно снять в соответствующем тестовом примере, следуя правилам игры.

### Пример

<code>game.in</code>	<code>game.out</code>
2	3
3 4	1
.o..	
ooo.	
.o.o	
2 2	
..	
.o	

## Задача С. Лингвистика

Имя входного файла:	lang.in
Имя выходного файла:	lang.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 Mb

Группа лингвистов решила составить карту распространения различных языков и на территории Южного Кавказа. Оказалось, что территории распространения каждого языка можно представить на карте в виде многоугольника, при этом никакие две территории не пересекаются иначе, чем по общей вершине или общей стороне. Для изучения взаимного влияния языков лингвисты поручили Вам для каждого языка вычислить количество соседних с территорией его распространения территорий распространения других языков. При этом соседними считаются две территории, имеющие хотя бы одну общую сторону. Известно, что «анклавов» (то есть территорий, полностью окружённых некоторой другой территорией) на карте нет.

### Формат входного файла

В первой строке входного файла содержится количество тестовых примеров  $T$ ,  $1 \leq T \leq 20$ .

В первой строке каждого тестового примера задано количество различных языков  $N$ ,  $1 \leq N \leq 10000$ . Далее идут описания территорий распространения языков. Для территории распространения  $i$ -го языка первая строка содержит целое число  $m_i$ ,  $3 \leq m_i \leq 10^5$  — количество вершин соответствующего многоугольника, в следующих  $m_i$  строках перечислены в порядке обхода координаты его вершин — по два целых неотрицательных числа  $x_{ij}$  и  $y_{ij}$ , не превосходящих  $10^9$ . При этом суммарное количество вершин на всей карте не превосходит  $10^5$ , а для любых двух отрезков, являющихся сторонами каких-либо многоугольников, верно следующее утверждение: они или не пересекаются, или имеют общую вершину, или один из них полностью содержит другой.

### Формат выходного файла

Для каждого тестового примера выведите в порядке, заданном входным файлом, количество территорий, соседних с территорией распространения данного языка.

## Пример

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

## Задача D. Очень простая задача

Имя входного файла: `prime.in`  
Имя выходного файла: `prime.out`  
Ограничение по времени: 20 секунд  
Ограничение по памяти: 64 Мб

В каждом наборе задач должна быть очень простая задача. Например, такая. По заданному числу  $N$  вычислите  $N$ -е простое число.

### Формат входного файла

В первой строке входного файла задано целое положительное число  $M$ , не превосходящее 10000 - количество тестовых примеров. Далее в последующих  $M$  строках идут целые числа  $N_i$ , ( $1 \leq N_i \leq 5 \times 10^7$ ).

### Формат выходного файла

Выходной файл должен состоять из  $M$  строк. В  $i$ -й строке выходного файла выведите  $N_i$ -е простое число.

### Пример

<code>prime.in</code>	<code>prime.out</code>
4	2
1	3
2	5
3	7
4	

## Задача Е. Канатная дорога

Имя входного файла:            road.in  
Имя выходного файла:           road.out  
Ограничение по времени:       2 секунды  
Ограничение по памяти:         64 Мб

На построенной для туристов в горах Кавказа канатной дороге станции были занумерованы следующим образом. Номером станции может служить только  $K$ -значное целое число с суммой цифр  $N$ , при этом десятичная запись этого числа не должна содержать нулей. Номера станциям присваиваются в порядке возрастания, например, при  $K = 3$  и  $N = 6$  станция у подножия горы получает номер 114, вторая — номер 123, третья — номер 132, четвёртая — номер 141 и так далее, до последнего номера 411.

Вы находитесь на станции с номером  $X$ . Ваша задача — узнать, сколько станций Вам придётся проехать по канатной дороге, чтобы попасть на станцию с номером  $Y > X$ .

### Формат входного файла

В первой строке входного файла содержится количество тестовых примеров  $T$ ,  $1 \leq T \leq 20$ .

В первой строке каждого тестового примера задано целое положительное число  $K$ ,  $2 \leq K \leq 100$  — количество знаков, используемых в номере станции. Во второй строке задаются два  $K$ -значных числа с одинаковой суммой цифр — номера станции, на которой Вы находитесь, и станции с большим номером, на которую Вам необходимо попасть.

### Формат выходного файла

Для каждого тестового примера выведите одно целое положительное число — количество станций (включая исходную станцию и станцию назначения), которые Вам придётся проехать по канатной дороге.

### Пример

road.in	road.out
2	2
2	4
12 21	
3	
222 321	

## Задача F. Прогноз сейсмической активности

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

На сейсмографической станции ведутся замеры сейсмической активности в определённых точках Южного Кавказа. Результат каждого замера оценивается по специальной шкале и представляет собой целое число в диапазоне от 0 до  $10^6$ . Замеры проводятся ежедневно и заносятся в специальный журнал.

Один из самых простых методов прогнозирования сейсмической активности заключается в следующем: в журнале ищется наиболее длинная последовательность замеров, совпадающая с последними сделанными замерами, и в качестве прогноза берётся значение сейсмической активности, идущее в журнале за этой последовательностью. Если таких последовательностей несколько, то берётся та из них, которая находится ближе к настоящему времени (то есть к концу последовательности). Ваша задача — реализовать подобный метод.

### Формат входного файла

В первой строке входного файла содержится количество тестовых примеров  $T$ ,  $1 \leq T \leq 20$ .

В первой строке каждого тестового примера задано целое положительное число  $N$ , не превосходящее 100000 — количество замеров в журнале. В следующей строке идут  $N$  неотрицательных целых чисел, каждое из которых не превосходит  $10^6$  — замеры в порядке их проведения.

### Формат выходного файла

Для каждого тестового примера выведите прогноз значения следующего замера сейсмической активности, полученный указанным в задаче методом, или  $-1$ , если такой прогноз сделать невозможно.

### Пример

<code>seism.in</code>	<code>seism.out</code>
2	-1
4	3
1 2 3 4	
10	
1 2 1 3 2 1 4 1 2 1	