



## Лото (30 мин., 10 баллов)

Ленни любит играть в лото. В игре он выбирает список из  $N$  уникальных целых чисел в пределах от 1 до  $M$ . Если его список совпадает со списком выпавших чисел, то он выигрывает большой приз.

У Ленни есть схема, которая, по его мнению, является удачной. Ему нравится выбирать список чисел так, что каждое число в нем хотя бы в два раза больше предыдущего. Например, если  $N = 4$  и  $M = 10$ , тогда списки Ленни – это:

1 2 4 8  
1 2 4 9  
1 2 4 10  
1 2 5 10

В таком случае у Ленни есть четыре списка на выбор.

Ваша программа для заданных значений  $N$  и  $M$  должна определить количество списков Ленни.

### Входной файл

Входной файл *input.txt* состоит из строк. Первая строка содержит десятичное целое число  $C$  ( $0 < C \leq 50$ ). Следующие  $C$  строк представляют пары десятичных целых чисел  $N$  и  $M$  ( $1 \leq N \leq 10$ ,  $1 \leq M \leq 2000$  и  $N \leq M$ ), разделитель – пробел.

Последняя строка не заканчивается символом перевода строки.

### Выходной файл

Выходной файл *output.txt* состоит из  $C$  строк, в каждой строке записано десятичное целое число равное количеству найденных списков для соответствующей пары  $N$  и  $M$  из входного файла.

Последняя строка не должна заканчиваться символом перевода строки.

### Пример

<i>input.txt</i>	<i>output.txt</i>
3	4
4 10	100
2 20	10000
2 200	

## Салон (30 мин., 10 баллов)

Кампания владеет несколькими салонами красоты. Если в салон приходит клиент и есть свободное место, то он обслуживается немедленно. Если свободного места нет, то клиент немедленно уходит из салона. Каждый клиент посещает один салон не более одного раза в день вне зависимости от результата посещения.

Для того, чтобы определить количество необслуженных клиентов, в каждом салоне установили компьютер, который регистрирует приход и уход клиентов. Это происходит следующим образом. Каждому клиенту назначается уникальная заглавная буква (верхний регистр) английского алфавита. Когда клиент приходит или уходит, тогда его буква добавляется в конец статистики. Сбор статистики заканчивается после того, когда последний клиент покинет салон.

Необходимо определить, сколько клиентов не было обслужено в каждом из салонов.

### Входной файл

Входной файл *input.txt* состоит из  $n$  (не больше 20) строк. Каждая строка описывает работу одного салона за один день. Это описание состоит из двух полей, разделённых пробелом. Первое – количество мест в салоне (целое десятичное число от 1 до 20 включительно), второе – статистика посещения этого салона.

Последняя строка не заканчивается символом перевода строки.

### Выходной файл

Выходной файл *output.txt* состоит из  $n$  строк. Каждая строка содержит целое десятичное число, равное количеству клиентов, которые не были обслужены в соответствующем салоне.

Последняя строка не должна заканчиваться символом перевода строки.

### Пример

<i>input.txt</i>	<i>output.txt</i>
2 ABBAJJKZKZ	0
3 GACCBDDBAGEE	1
3 GACCBGDDBAEE	0
1 ABVCCA	2



## Слова (30 мин., 10 баллов)

Дано три слова A, B и C. Необходимо определить, можно ли сформировать слово C путём комбинирования букв слов A и B, используя следующие правила:

1. Слово C состоит из всех букв слов A и B.
2. Буквы слова C, взятые из A (B), находятся в том же порядке, что и в слове A (B).
3. Буквы, взятые из слова A (B), не обязательно идут подряд в слове C.

Например, из букв слов “cat” и “tree” можно сформировать слово “tcræete” и слово “catrtee”, но нельзя сформировать слово “cttaree”.

### Входной файл

Первая строка входного файла *input.txt* содержит целое десятичное число  $n$  ( $1 \leq n \leq 1000$ ). Каждая из следующих  $n$  строк состоит из трёх слов A, B и C, разделитель – пробел.

Последняя строка не заканчивается символом перевода строки.

Слова состоят только из строчных (нижний регистр) букв английского алфавита. Длина слова C равна сумме длин слов A и B. Длина слова A (B) не превышает 200.

### Выходной файл

Выходной файл *output.txt* состоит из  $n$  строк. В каждой строке записано число 0 или 1. Каждое число определяет возможность (1) или невозможность (0) формирования слова C из букв слов A и B из соответствующей строки входного файла.

Последняя строка не должна заканчиваться символом перевода строки.

### Пример

<i>input.txt</i>	<i>output.txt</i>
3	1
cat tree tcræete	1
cat tree catrtee	0
cat tree cttaree	

## Сейф (30 мин., 10 баллов)

Необходимо открыть сейф. У сейфа есть кодовый замок, который использует буквы вместо чисел. Для того, чтобы открыть сейф, надо использовать специальное слово. Это слово содержит от пяти до двенадцати не повторяющихся заглавных (верхний регистр) букв английского алфавита. Каждой букве соответствует число, равное её порядковому номеру в английском алфавите (букве А соответствует 1, букве В – 2, ..., букве Z – 26).

Пять букв  $\alpha$ ,  $\beta$ ,  $\delta$ ,  $\gamma$  и  $\varepsilon$  из слова образуют комбинацию для кодового замка. Механизм кодового замка вычисляет выражение  $(\alpha^1 - \beta^2 + \delta^3 - \gamma^4 + \varepsilon^5)$ . Если результат выражения равен секретному числу и если эта комбинация лексикографически больше, чем все другие, сформированные из букв того же слова, которые дают такой же результат, то сейф открывается.

Необходимо разработать программу, которая находит буквенный код, если известны секретное число и специальное слово.

### Входной файл

Входной файл *input.txt* состоит из  $n$  ( $1 \leq n \leq 20$ ) строк. Каждая строка состоит из двух полей, разделённых пробелом. Первое – секретное число (положительное целое десятичное, не более 12000000). Второе – специальное слово.

Последняя строка не заканчивается символом перевода строки.

### Выходной файл

Выходной файл *output.txt* состоит из  $n$  строк. Каждая строка выходного файла соответствует строке входного файла и содержит буквенный код, если его можно получить из секретного числа и специального слова, в противном случае строка содержит символ #.

Последняя строка не должна заканчиваться символом перевода строки.

### Пример

<i>input.txt</i>	<i>output.txt</i>
1 ABCDEFGHIJKL	LKEBA
11700519 ZAYEXIWOVU	YOXUZ
3072997 SOUGHT	GHOST
1234567 THEQUICKFROG	#