

## Задача А. Термоконтроль

Имя входного файла:	standard input
Имя выходного файла:	standard output
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

В современных компьютерных играх требования к видеокарте довольно велики. Поэтому проблема охлаждения видеокарт стоит довольно остро. Разработчики одной из игр решили встроить в игровой движок блок контроля температуры видеокарты. Температура не должна быть слишком высокой или слишком низкой. Однако в процессе отладки обнаружилась ошибка в алгоритмах контроля температуры, и теперь блок работает следующим образом: каждую минуту блок может дать команду увеличить или уменьшить температуру на целое число градусов, не превосходящее трёх. Поправка выбирается блоком с некоторой данной вероятностью в зависимости от текущей температуры. Кроме того, каждую минуту температура уменьшается на один градус (эффект остывания).

Вам даны параметры изменения температуры. Посчитайте вероятность того, что температура во время процесса отладки блока будет находиться в заданном интервале.

### Формат входного файла

Первая строка ввода содержит количество тестов ( $1 \leq T \leq 20$ ). Первая строка каждого теста содержит четыре целых числа  $A$ ,  $B$ ,  $C$ ,  $N$ , где  $A$  — минимальная разрешённая температура,  $B$  — максимальная разрешённая температура,  $C$  — начальная температура,  $N$  — время процесса отладки в минутах ( $0 \leq A \leq B \leq 30, A \leq C \leq B, 0 \leq N \leq 30$ ). Каждая из следующих  $B - A + 1$  строк содержит 7 неотрицательных целых чисел, в сумме дающих 100 — вероятность в процентах изменения температуры на  $-3$ ,  $-2$ ,  $-1$ ,  $0$ ,  $1$ ,  $2$  и  $3$  градуса соответственно.

### Формат выходного файла

Выведите  $T$  строк вида Case #A: B, где  $A$  — номер теста (начиная с 1),  $B$  — искомая вероятность для данного теста.

### Пример

standard input	standard output
2	Case #1: 0.45
1 2 1 2	Case #2: 0.59049
0 0 0 10 50 40 0	
0 0 50 0 0 30 20	
3 5 4 5	
0 0 0 10 20 30 40	
0 0 10 20 30 40 0	
0 10 20 30 40 0 0	

## Задача В. Числа Белла

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

При разработке «искусственного интеллекта», управляющего поведением компьютерных оппонентов, возникла следующая задача.

Число Белла  $B_n$  равно количеству разбиений множества из  $n$  элементов на произвольное количество непустых подмножеств.  $B_0 = 1$ , так как существует только одно разбиение пустого множества.  $B_3 = 5$ , так как существует 5 возможных разбиений множества  $\{a, b, c\}$  из трех элементов:  $\{\{a\}, \{b\}, \{c\}\}, \{\{a, b\}, \{c\}\}, \{\{a, c\}, \{b\}\}, \{\{a\}, \{b, c\}\}, \{\{a, b, c\}\}$ . Рассмотрим определитель  $D_n$ :

$$\begin{vmatrix} B_0 & B_1 & B_2 & \dots & B_n \\ B_1 & B_2 & B_3 & \dots & B_{n+1} \\ \dots & \dots & \dots & \dots & \dots \\ B_n & B_{n+1} & B_{n+2} & \dots & B_{2n} \end{vmatrix}$$

По заданному простому числу  $p$  найти наибольшее целое  $k$ , для которого  $D_n$  делится на  $p^k$ .

### Формат входного файла

Каждая строка содержит два целых числа  $n$  и  $p$  ( $0 \leq n, p \leq 10000$ ). Известно, что  $p$  — простое число. Всего в файле не более 23917 строк.

### Формат выходного файла

Для каждой пары входных значений  $n$  и  $p$  в отдельной строке вывести наибольшее целое  $k$ , для которого  $D_n$  делится на  $p^k$ .

### Пример

standard input	standard output
1 5	0
3 2	2
4 2	5
4 3	2
10000 3	24962375

## Задача С. Запоминаемость

Имя входного файла:            **standard input**  
Имя выходного файла:           **standard output**  
Ограничение по времени:       4 секунды  
Ограничение по памяти:         64 мегабайта

При подборе значений для параметров различных объектов в компьютерных играх часто бывает важен не только баланс, но и «запоминаемость» параметров игроками. Часто для повышения запоминаемости оказывается достаточно выбрать и расположить объекты в таком порядке, чтобы игрок мог заметить закономерность.

Вам дан список целых положительных чисел — ударная сила различных типов оружия в игре. Найдите количество чисел в максимальном подмножестве данного множества, обладающем тем свойством, что все числа из этого подмножества могут быть расположены в цепочку так, что каждое число либо делит, либо делится на своего соседа. По мнению консультанта-психолога, такая цепочка запоминается довольно легко.

### Формат входного файла

Первая строка ввода содержит количество тестов  $T$  ( $1 \leq T \leq 35$ ). Каждая из следующих  $T$  строк содержит количество элементов множества  $N$  ( $1 \leq N \leq 17$ ) и  $N$  целых положительных чисел, каждое — от 1 до  $10^9$  включительно.

### Формат выходного файла

Выведите  $T$  строк вида **Case #A: B**, где  $A$  — номер теста (начиная с 1),  $B$  — искомая величина для данного теста.

### Пример

standard input	standard output
2	Case #1: 3
3 1 2 3	Case #2: 4
5 2 3 4 5 6	

## Задача D. Пересечения

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

«Базовые» карты, на которых происходит игра, строятся следующим образом.

Пусть дано положительное целое  $N$  — номер этапа. Пусть  $A$ ,  $B$  и  $C$  — такие неотрицательные целые числа, что  $A + B + C = N$ . Пусть на координатной оси отмечено  $N$  точек с одинаковым интервалом между каждыми двумя соседними — «стартовые» комнаты для игроков. Система переходов представляет собой прямые, проведённые под углом 45 градусов к координатной оси через  $A$  левых точек, под углом 90 градусов к координатной оси через  $B$  следующих точек и под углом 135 градусов к координатной оси — через оставшиеся  $C$  точек. Эти прямые пересекутся в некотором количестве перекрёстков. Например, для  $N = 5$ ,  $A = 1$ ,  $B = 2$ ,  $C = 2$  получается 6 перекрёстков. Чем больше перекрёстков возможно при разных выборах  $A$ ,  $B$ ,  $C$ , тем более острой может быть игровая ситуация. Ваша задача довольно проста — для заданного  $N$  посчитать сумму количеств перекрёстков для всех возможных троек  $A$ ,  $B$ ,  $C$ .

### Формат входного файла

Первая строка ввода содержит количество тестов ( $1 \leq T \leq 1000$ ). Каждая из следующих  $T$  строк содержит число  $N$ , ( $2 \leq N \leq 10^6$ ) — количество точек на прямой в данном тесте.

### Формат выходного файла

Выведите  $T$  строк вида `Case #A: B`, где  $A$  — номер теста (начиная с 1),  $B$  — сумма количеств точек пересечения для заданного  $N$ .

### Пример

standard input	standard output
3	Case #1: 3
2	Case #2: 13
3	Case #3: 91
5	

## Задача Е. Баланс

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 10 секунд  
Ограничение по памяти: 64 мегабайта

При подборе параметров для разных классов персонажей в компьютерных играх важным является игровой баланс. Пусть класс полностью определяется  $m$  параметрами. Считаем, что класс  $A$  однозначно превосходит класс  $B$ , если для каждого из  $m$  параметров  $A_i \geq B_i$  (то есть значение этого параметра для класса  $A$  не менее, чем значение этого параметра для класса  $B$ , при этом хотя бы для одного  $i$   $A_i > B_i$ ). При наличии однозначного превосходства одного класса над другим игровой баланс существенно нарушается.

Вам предложено  $N$  различных классов. Требуется найти количество классов, для каждого из которых среди оставшихся  $N - 1$  не существует ни одного однозначно превосходящего класса.

### Формат входного файла

Первая строка ввода содержит количество тестов  $T$  ( $1 \leq T \leq 10$ ). Первая строка каждого теста содержит 2 числа:  $N$  ( $1 \leq N \leq 50000$ ) — количество предложенных классов и  $M$  ( $1 \leq M \leq 4$ ) — количество параметров, определяющих класс. Далее следуют  $N$  строк, каждая из которых содержит  $M$  целых чисел — параметры классов, разделённые пробелами (каждый параметр меньше  $1.1 \times 10^9$  по модулю). При этом не существует двух различных классов с совпадающими параметрами.

### Формат выходного файла

Выведите  $T$  строк вида `Case #A: B`, где  $A$  — номер теста (начиная с 1),  $B$  — требуемое количество классов.

### Пример

standard input	standard output
2	Case #1: 1
4 1	Case #2: 3
1	
2	
3	
4	
4 2	
0 0	
1 1	
2 0	
0 2	

## Задача F. Сюжет игры

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Серия фантастических романов, которая легла в основу сюжета игры, состоит из  $n$  томов. Тома пронумерованы от 1 до  $n$  и выстроены в ряд на книжной полке руководителя проекта в соответствии с запланированными в игре событиями. Он не любит строгого порядка (так как тогда тем, кто читал книгу, будет менее интересно играть), но также он не любит и полнейшего хаоса (тогда просто теряется логика игрового мира).

Руководитель проекта использует следующее правило: он рассчитывает «хаотичность» для каждой расстановки томов как сумму абсолютных величин разностей номера тома и его позиции на полке по всем  $N$  томам и считает допустимой только расстановку, «хаотичность» которой равна заданному числу  $d$ . Сколько различных расстановок с этим свойством существует?

### Формат входного файла

Первая строка ввода содержит количество тестов  $T$  ( $1 \leq T \leq 100$ ). Каждая из следующих  $T$  строк содержит два целых числа — данные для одного теста: количество томов  $n$  ( $1 \leq n \leq 50$ ) и требуемую хаотичность  $d$  ( $0 \leq d \leq 10000$ ), которые разделены пробелом.

### Формат выходного файла

Выведите  $T$  строк вида `Case #A: B`, где  $A$  — номер теста (начиная с 1),  $B$  — количество перестановок  $n$  томов с «хаотичностью»  $d$ , взятое по модулю 100007.

### Пример

standard input	standard output
5	Case #1: 1
2 0	Case #2: 1
2 2	Case #3: 0
4 1	Case #4: 3
4 2	Case #5: 9
4 6	

## Задача G. Трёхмерные модели

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

При разработке системы 3d-моделей объектов, встречающихся в игре, возникла следующая задача: Дано множество точек в трёхмерном пространстве. Требуется для всех возможных  $k$  посчитать количество  $k$ -вершинных граней выпуклого многогранника минимального объёма, содержащего все данные точки.

### Формат входного файла

Первая строка ввода содержит количество тестов  $T$  ( $1 \leq T \leq 100$ ). Первая строка каждого теста содержит количество точек в множестве —  $N$  ( $4 \leq N \leq 30$ ). Далее следуют  $N$  строк, каждая из которых содержит 3 целых числа  $X, Y, Z$  ( $-1000 \leq X, Y, Z \leq 1000$ ) — координаты точек. Гарантируется, что любой выпуклый многогранник, содержащий все данные точки, будет иметь положительный объём.

### Формат выходного файла

Для каждого из  $T$  тестов выведите строку вида `Case #A:`, где  $A$  — номер теста (начиная с 1), а затем — ещё  $M$  строк, где  $M$  — количество разных типов граней (под типом грани понимается количество её вершин). В каждой из следующих  $M$  строк нужно вывести два числа:  $k$  — количество вершин в грани и  $q_k$  — количество  $k$ -вершинных граней в многограннике. После  $k$  нужно напечатать двоеточие и затем — пробел. Результат нужно выводить в порядке возрастания  $k$ .

### Пример

standard input	standard output
2	Case #1:
6	3: 5
0 0 0	5: 1
2 0 0	Case #2:
0 2 0	3: 4
2 2 0	
3 1 0	
1 1 1	
5	
0 0 0	
1 1 0	
0 2 0	
2 1 0	
1 1 1	

## Задача Н. Многопользовательский режим

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Современную компьютерную игру сложно представить без многопользовательского режима. При проработке такого режима часто возникают проблемы при организации турниров, особенно если игра ведётся «команда на команду».

Например, на сервере собралось чётное количество  $n$  игроков. Они играют турнир из нескольких дуэлей «команда из  $n/2$  человек на команду из  $n/2$  человек», меняя составы команд. Для того, чтобы турнир состоялся, любые два игрока должны находиться в противоположных командах как минимум дважды (дабы исключить флуктуации). Каково минимальное количество раундов может содержать состоявшийся турнир?

Например, если у нас есть 8 игроков, пронумерованных от 1 до 8, то мы можем организовать деление на команды следующим образом: (1, 2, 3, 4) – (5, 6, 7, 8), (1, 2, 5, 6) – (3, 4, 7, 8), (1, 3, 5, 7) – (2, 4, 6, 8), (1, 4, 6, 7) – (2, 3, 5, 8).

### Формат входного файла

Первая строка ввода содержит количество тестов  $T$  ( $1 \leq T \leq 50$ ). Каждая из следующих  $T$  строк содержит чётное число  $N$  ( $2 \leq N \leq 300$ ) — количество игроков на сервере в данном тесте.

### Формат выходного файла

Выведите  $T$  строк вида `Case #A: B`, где  $A$  — номер теста (начиная с 1),  $B$  — нужное количество раундов для заданного  $N$ .

### Пример

<code>standard input</code>	<code>standard output</code>
4	Case #1: 2
2	Case #2: 3
4	Case #3: 5
12	Case #4: 5
16	



## Задача I. Торговые альянсы

Имя входного файла:	standard input
Имя выходного файла:	standard output
Ограничение по времени:	4 секунды
Ограничение по памяти:	64 мегабайта

Схема одного из миров, в которых проходит действие игры, выглядит так. Есть города, принадлежащие различным расам. Каждый город может принадлежать только одной расе. Расы пронумерованы числами от 1 до 4. При этом в силу сложных взаимоотношений между расами дороги могут соединять только города рас 1 и 2, 2 и 3, 3 и 4, 4 и 1.

Вам нужно посчитать максимальное число потенциальных торговых альянсов в данном мире — то есть непересекающихся множеств из четырёх городов (по одному городу каждой расы), таких, что город  $A$  расы 1 соединён с городом  $B$  расы 2, тот соединён с городом  $C$  расы 3, тот, в свою очередь, с городом  $D$  расы 4, а тот — с первоначальным городом  $A$ .

### Формат входного файла

Первая строка ввода содержит количество тестов  $T$  ( $1 \leq T \leq 10$ ). Первая строка каждого теста содержит 4 числа:  $N_1, N_2, N_3, N_4$  — количества городов, населённых расами 1, 2, 3 и 4 соответственно ( $1 \leq N_1, N_2, N_4 \leq 10, 1 \leq N_3 \leq 7$ ). Далее следуют  $2 \times N_1$  строк,  $2 \times i$ -я строка содержит количество городов расы 2, соединённых с  $i$ -м городом расы 1 и номера этих городов,  $(2i + 1)$ -я строка содержит количество городов расы 4, соединённых с  $i$ -м городом расы 1, а затем — номера этих городов (города каждой расы занумерованы от 0 до  $N_P - 1$ , где  $P$  — номер расы). Далее следуют  $2 \times N_3$  строк,  $2 \times i$ -я строка содержит количество городов расы 2, соединённых с  $i$ -м городом расы 3 и номера этих городов,  $(2i + 1)$ -я строка содержит количество городов расы 4, соединённых с  $i$ -м городом расы 3, а затем — номера этих городов.

### Формат выходного файла

Выведите  $T$  строк вида **Case #A: B**, где  $A$  — номер теста (начиная с 1),  $B$  — количество торговых альянсов для данного теста.

## Пример

standard input	standard output
2	Case #1: 1
1 3 2 4	Case #2: 3
3 0 1 2	
4 0 1 2 3	
2 0 1	
1 0	
2 1 2	
2 1 3	
4 7 4 7	
3 0 1 5	
3 0 1 5	
3 0 2 6	
3 0 2 6	
3 1 2 6	
3 1 2 6	
3 0 1 2	
3 0 1 2	
2 0 2	
2 0 2	
1 0	
1 0	
2 1 0	
2 1 0	
2 0 1	
2 0 1	

## Задача J. Невыполнимое задание

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Иногда при разработке сценариев игры бывает необходимо придумать невыполнимое задание для игрока (например, чтобы слишком простая тактика «брать все задания» не срабатывала, или же просто «чтобы жизнь малиной не казалась»).

Одно из заданий выглядит так. Игроку дано дерево. В каждый узел дерева можно поместить одну или несколько фишек. После расположения фишек можно выбрать узел дерева, в котором есть две или больше фишек, и убрать из него две фишки, при этом поместив одну фишку в любой из соседних узлов. Такую операцию можно повторять несколько раз. Ваша задача — найти максимальное количество фишек (по модулю  $M$ ), которое можно разместить на дереве так, чтобы выполнялось условие: найдётся хотя бы один узел дерева, в который будет невозможно поместить ни одной фишки с помощью указанных операций. Собственно, в помещении фишки в данный узел и будет состоять искомое невыполнимое задание.

### Формат входного файла

Первая строка ввода содержит количество тестов  $T$  ( $1 \leq T \leq 20$ ). Первая строка каждого теста содержит два числа:  $N$  ( $2 \leq N \leq 30000$ ) — количество узлов в дереве и  $M$  ( $2 \leq M \leq 2^{31} - 1$ ). Далее следует  $N - 1$  строка, каждая из которых содержит номера двух смежных узлов дерева (узлы занумерованы от 1 до  $N$ ), разделённых пробелом.

### Формат выходного файла

Выведите  $T$  строк вида `Case #A: B`, где  $A$  — номер теста (начиная с 1),  $B$  — искомая величина для данного теста.

### Пример

standard input	standard output
2	Case #1: 16
6 997	Case #2: 5
1 2	
1 4	
3 4	
5 3	
3 6	
7 13	
1 2	
1 3	
1 4	
2 5	
3 6	
4 7	