

Задача А. Битовые операции

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

В данной задаче даны целые числа A и k . В выходной файл, по одному в строке, выведите целые числа, являющиеся ответами на следующие запросы:

1. Значение k -го бита числа A , т.е 0 или 1
2. Число, которое получается из A установкой k -го бита, равным 1
3. Число, которое получается из A установкой k -го бита, равным 0
4. Число, которое получается из A инвертированием k -го бита
5. Число, которое получается из A , если обнулить k его последних бит.
6. Число, которое состоит из последних k бит числа A

Формат входного файла

Во входном файле записаны через пробел два числа — A ($0 \leq A < 2^{31}$) и k ($0 \leq k \leq 31$).

Формат выходного файла

В выходной файл выведите ответы на запросы, указанные в условии задачи.

Примеры

<code>bits.in</code>	<code>bits.out</code>
5 1	0 7 5 7 4 1

Задача В. Министры его величества

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

Его Величество Король Бубей Второй пожелал назначить новый кабинет министров (информация о том, что случилось со старым — строго засекречена). К составу кабинета министров есть следующие пожелания:

1. Министров должно быть как можно меньше (так ими легче управлять, да и на зарплате можно сэкономить);

2. Для каждой области (строительство, финансы и т.д.) должен быть хотя бы один министр, который в ней разбирается.

На рассмотрение Его Величества поступило N кандидатур. Определите, сколько и каких людей должны получить министерские посты.

Формат входного файла

В первой строке входного файла даны два натуральных числа N и K ($1 \leq N \leq 20$, $1 \leq K \leq 30$) — количество кандидатов в списке и количество областей, в которых министры должны разбираться. Далее идут N строк следующего формата: в начале строки вводится число P_i ($1 \leq P_i \leq K$) — количество областей, в которых разбирается i -й кандидат, затем вводятся номера этих областей (натуральные числа, не превышающие K).

Формат выходного файла

Сначала выведите количество министров, которое планируется назначить, исходя из требований задачи, затем перечислите номера подходящих кандидатов в порядке возрастания. Если решений несколько, выберите то, в котором участвуют кандидаты, идущие раньше по списку (то есть минимальное лексикографически). Гарантируется, что решение всегда существует.

Примеры

<code>ministers.in</code>	<code>ministers.out</code>
3 2 2 1 2 2 1 2 2 1 2	1 1
4 3 1 1 1 2 1 3 2 1 2	2 3 4

Задача С. Шашки

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

Как известно, в ЛКШ каждую смену проходит турнир по шашкам. В ЛКШ.Зима организаторы тоже хотели провести такой турнир, но проводить его по стандартным правилам было бы банально. Культурги полезли в Википедию и выяснили, что шашки имеют много разновидностей. Одна из них, «Новогодняя», абсолютно не похожа на привычные нам шашки. Игра происходит на прямоугольной доске $n \times m$. В каждой из n строк расположено ровно 2 шашки (одна белая, другая черная, при этом белая всегда стоит левее черной). За один ход игрок может взять любую шашку своего цвета и передвинуть ее на любую клетку

этой же строки, находящуюся ближе к шашке соперника, но не перепрыгивая через нее. Проигрывает тот, кто не может сделать ход. Найдите, кто выиграет при правильной игре.

Формат входного файла

Во входном файле записаны через пробел два числа — n и m ($1 \leq n \leq 10^5$, $2 \leq m \leq 10^9$). Далее в n строках записаны по два целых числа — l и r ($1 \leq l < r \leq m$) — расположение белой и черной шашки в i -ой строке соответственно.

Формат выходного файла

В выходной файл выведите «First», если у первого игрока есть выигрышная стратегия и «Second» в противном случае.

Примеры

checkers.in	checkers.out
1 10 1 5	First
2 10 1 5 1 5	Second

Задача D. Пересечение моста

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

Известная математическая загадка о мосте и фонаре звучит следующим образом. Четыре человека хотят пересечь мост. Он настолько старый, что не может выдержать больше двух человек одновременно. Кроме того, на улице уже темно, поэтому люди не могут ходить по мосту без фонаря. Для каждого человека известно время, за которое он пересекает мост в одиночестве. Пусть это будет 1, 2, 5 и 10 минут соответственно. Однако, когда люди пересекают мост вдвоем, то они идут со скоростью самого медленного из них. У них имеется всего один фонарик. Его нельзя оставлять на мосту — необходимо, чтобы фонарик постоянно находился у какого-то человека. Требуется посчитать минимальное время, необходимое для прохождения всех людей через мост.

В данном случае ответ равен 17. Первый и второй проходят по мосту, потратив 2 минуты. Первый возвращается, получив дополнительную минуту. После чего мост пересекают третий и четвертый за 10 минут. Второй возвращается с фонариком за 2 минуты. Наконец, первый и второй пересекают мост за 2 минуты. Это дает в общей сложности $2 + 1 + 10 + 2 + 2 = 17$ минут.

В этой задаче к мосту подошло N человек, для каждого из которых известно время, которое человек тратит на переход по мосту. Вычислите минимальное время, необходимое для того, чтобы все N человек оказались на другом берегу.

Формат входного файла

В первой строке входного файла дано натуральное число N ($1 \leq N \leq 6$). Во второй строке через пробел заданы N натуральных чисел, не превосходящих 100, где i -е число означает, сколько времени i -й человек тратит на пересечение моста в одиночестве.

Формат выходного файла

Выходной файл должен содержать единственное число — ответ на задачу.

Примеры

bridge.in	bridge.out
4 1 2 5 10	17
5 1 2 3 4 5	16