

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 входных данных

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

Формат выходных данных

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. Выращивание бактерий

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вы — большой любитель бактерий. Вам хочется вырастить немного бактерий в коробочке.

Изначально коробочка пуста. Каждое утро можно положить любое количество бактерий в коробочку. Каждую ночь каждая бактерия делится на две бактерии. Когда-нибудь вы надеетесь увидеть ровно x бактерий в коробочке.

Какое минимальное количество бактерий вам суммарно надо положить в коробочку для достижения этой цели?

Формат входных данных

В единственной строке записано одно целое число x ($1 \leq x \leq 10^9$) — количество бактерий.

Формат выходных данных

Единственная строка, содержащая одно целое число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
5	2
8	1

Замечание

В первом примере мы можем положить одну бактерию в коробочку утром первого дня, а утром третьего дня в коробочке будет 4 бактерий. Теперь надо положить в коробочку ещё одну бактерию, в результате чего в коробочке окажется 5 бактерий. Мы суммарно добавили 2 бактерии в коробочку, так что ответ равен 2.

Во втором примере мы можем положить одну бактерию утром первого дня, тогда утром четвёртого дня в коробочке будет 8 бактерий. Таким образом, ответ равен 1.

Задача С. 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 входных данных

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 выходных данных

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 D. Сумма «случайных» чисел

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

На столе лежат n шариков, на каждом шарике написано натуральное число. Настя и Сева используют эти шарики, чтобы получать «случайные» числа. Процедура получения «случайного» числа следующая. Сначала Настя берёт со стола некоторое непустое подмножество шариков; при этом необходимо, чтобы на столе остался хотя бы один шарик. Затем Сева также берёт какое-то непустое подмножество шариков, оставшихся на столе; после этого шариков на столе может не остаться. Наконец, Настя и Сева вычисляют «случайное» число $r = a \bmod b$, где a — это сумма чисел на шариках Насти, а b — сумма чисел на шариках Севы; $a \bmod b$ — это остаток от деления a на b . После этого все шарики возвращаются на стол.

Предположим, что Настя выбрала каждое допустимое подмножество шариков, и для каждого из них Сева выбрал по одному разу все допустимые подмножества оставшихся шариков. Найдите сумму всех «случайных» чисел, которые получились при этом.

Формат входных данных

В первой строке задано целое число n ($2 \leq n \leq 16$). Вторая строка содержит n целых чисел s_1, s_2, \dots, s_n через пробел; s_i — это число, написанное на i -м шарике ($1 \leq s_i \leq 1\,000\,000$).

Формат выходных данных

В первой строке выведите одно число — сумму всех получившихся у Насти и Севы «случайных» чисел.

Примеры

стандартный ввод	стандартный вывод
2 1 1	0
3 1 1 1	3
3 3 1 2	8

Задача Е. Забавная игра

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Легендарный учитель математики Юрий Петрович придумал забавную игру с числами. А именно, взяв произвольное целое число, он переводит его в двоичную систему счисления, получая некоторую последовательность из нулей и единиц, начинающуюся с единицы. (Например, десятичное число $19 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$ в двоичной системе запишется как 10011_2). Затем учитель начинает сдвигать цифры полученного двоичного числа по циклу (так, что последняя цифра становится первой, а все остальные сдвигаются на одну позицию вправо), выписывая образующиеся при этом последовательности из нулей и единиц в столбик — он подметил, что независимо от выбора исходного числа получающиеся последовательности начинают с некоторого момента повторяться. И, наконец, Юрий Петрович отыскивает максимальное из выписанных чисел и переводит его обратно в десятичную систему счисления, считая это число результатом проделанных манипуляций. Так, для числа 19 список последовательностей будет таким:

10011
11001
11100
01110
00111
10011
...

и результатом игры, следовательно, окажется число $1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 28$

Поскольку придуманная игра с числами все больше занимает воображение учителя, отвлекая тем самым его от работы с ну очень одаренными школьниками, Вас просят написать программу, которая бы помогла Юрию Петровичу получать результат игры без утомительных ручных вычислений.

Формат входных данных

Входной файл содержит одно целое число N ($0 \leq N \leq 32\,767$).

Формат выходных данных

Ваша программа должна вывести в выходной файл одно целое число, равное результату игры.

Примеры

стандартный ввод	стандартный вывод
1	1
23	30

Задача F. Очень много ИЛИ

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1.5 секунд
Ограничение по памяти: 256 мегабайт

Давным давно, на великом заводе "Два Станка" жил-был Андрей. Он очень любил битовые операции, жить без них не мог. И вот, в очередной раз он решил выполнить как можно больше операций, не тратя сил ни на что, кроме вычислений. Попробуйте и вы!

Есть стартовое число X и три типа операций, заданные двумя числами. Первое число - номер типа операции T , второе - вспомогательный аргумент A .

1) Если $T = 1$, то мы должны применить **битовое И** между аргументом A и текущим значением X .

2) Если $T = 2$, то мы должны применить **битовое ИЛИ** между аргументом A и текущим значением X .

3) Если $T = 3$, то мы должны применить **исключающее ИЛИ** между аргументом A и текущим значением X .

Андрей выписал значения для целых N операций, после чего, делает следующее:

Сначала он применит к X группу операций состоящую только из операции под номером 1;

Потом группу операций состоящую из операции под номерами 1 и 2 в порядке следования;

И так далее...

В конце выполнит группу из всех операций под номерами 1, 2, ..., N .

Андрей хочет знать, чему будет равно значение X после каждой группы операций. Помогите ему с этим!

Формат входных данных

В первой строке заданы числа N и X ($1 \leq N \leq 10^5, 1 \leq X \leq 2^{30}$), которые соответственно равны количеству операций и стартовому значению переменной X .

В последующих N строках находится информация о всех запросах, по два числа T и A в каждой строке, где $1 \leq T \leq 3$ - тип запроса и $1 \leq A \leq 2^{30}$ - вспомогательный аргумент.

Формат выходных данных

Выведите N чисел, значение X после выполнение первых k групп операций.

Пример

стандартный ввод	стандартный вывод
3 10	9
3 3	15
2 5	12
1 12	