

## 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

bits.in	bits.out
5 1	0 7 5 7 4 1
29 3	1 29 21 21 24 5

### Замечание

Запрещается использовать циклы, условия, сложение, вычитание (кроме получения константы -1), умножение и деление.

Биты нумеруются с нуля, то есть младший бит имеет номер 0.

## 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 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 С. 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 D. Снусмумрик и Клипдассы

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

Однажды Снусмумрик набрел на необычную полянку, на которой живут Клипдассы. Всего есть 30 норок, в каждой из которых живёт по одному Клипдассу. Норки пронумерованы от 1 до 30.

Клипдассы решили сыграть со Снусмумриком в игру, которая длится  $q$  секунд. Каждую секунду каждый из 30 Клипдассов либо запрыгивает в свою норку, либо выпрыгивает из своей норки, либо остаётся на месте. Когда Клипдасс из  $i$ -й норки запрыгивает в неё или выпрыгивает из неё, Снусмумрик получает  $2^i$  очков. После каждой секунды Снусмумрик записывает, сколько очков он получил.

Известно, что после окончания игры все Клипдассы оказались в своих норках. Но Снусмумрик забыл, сколько Клипдассов до игры не сидели в своих норках. Помогите Снусмумрику!

### Формат входных данных

В первой строке задано одно целое число  $q$  ( $1 \leq q \leq 100\,000$ ) — продолжительность игры в секундах.

Затем следует  $q$  строк. В  $i$ -й из них записано одно целое число  $n$  ( $0 \leq n \leq 2^{31} - 2$ ) — количество очков, полученных Снусмумриком на  $i$ -й секунде. Гарантируется что все данные о количестве полученных очков записаны корректно.

### Формат выходных данных

Выведите одно целое число — количество Клипдассов, которые перед игрой не сидели в своих норках.

### Примеры

стандартный ввод	стандартный вывод
1 0	0
2 6 2	1

## Задача Е. Очень много ИЛИ

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

Андрей очень любит битовые операции, жить бз них не может. И вот, коротая очередно вечер, он решил выполнить как можно больше операций, не тратя сил ни на что, кроме вычислений. И вот, что он придумал. У Андрея есть стартовое число  $X$  и три типа операций, заданные двумя числами. Номером типа  $T$  и вспомогательным аргументом  $A$ .

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. И по его плану, он хочет максимизировать количество вычислений.

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;

И так далее...

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

Однако Андрей не рассчитал время, а результат после выполнения очередной группы операций ему очень интересен. Помогите Андрюше посчитать значение  $X$  после поочерёдного выполнения каждой группы преобразований.

### 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 наодится 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	

## Задача F. Симпатичные узоры

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

Компания BrokenTiles планирует заняться выкладыванием во дворах у состоятельных клиентов узор из черных и белых плиток, каждая из которых имеет размер  $1 \times 1$  метр. Известно, что дворы всех состоятельных людей имеют наиболее модную на сегодня форму прямоугольника  $M \times N$  метров.

Однако при составлении финансового плана у директора этой организации появилось целых две серьезных проблемы: во первых, каждый новый клиент очевидно захочет, чтобы узор, выложенный у него во дворе, отличался от узоров всех остальных клиентов этой фирмы, а во вторых, этот узор должен быть симпатичным.

Как показало исследование, узор является симпатичным, если в нем нигде не встречается квадрата  $2 \times 2$  метра, полностью покрытого плитками одного цвета.

Для составления финансового плана директору необходимо узнать, сколько клиентов он сможет обслужить, прежде чем симпатичные узоры данного размера закончатся. Помогите ему!

### Формат входных данных

На первой строке входного файла находятся два положительных целых числа, разделенные пробелом —  $M$  и  $N$  ( $1 \leq M \times N \leq 30$ ).

### Формат выходных данных

Выведите в выходной файл единственное число — количество различных симпатичных узоров, которые можно выложить во дворе размера  $M \times N$ . Узоры, получающиеся друг из друга сдвигом, поворотом или отражением считаются различными.

### Примеры

<code>tilings.in</code>	<code>tilings.out</code>
1 1	2
1 2	4
4 1	16
2 3	50

## Задача G. Медианная сумма

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Вам дан массив из  $N$  чисел  $A_1, A_2, \dots, A_N$ .

Рассмотрим суммы всех непустых подмножеств этого массива. Это  $2^N - 1$  сумм, нечетное число.

Пусть  $S_1, S_2, \dots, S_{2^N-1}$  это список всех этих сумм в неубывающем порядке. Найдите медиану этого массива, то есть число  $S_{2^{N-1}}$ .

### Формат входных данных

В первой строке входных данных находится число  $N$  ( $1 \leq N \leq 2000$ ). В следующей строке находится элементы массива  $A_i$  ( $1 \leq A_i \leq 2000$ ).

### Формат выходных данных

Выведите единственное число — медиану отсортированного массива сумм.

### Примеры

стандартный ввод	стандартный вывод
3 1 2 1	2
1 58	58

## Задача Н. Поможем дикой природе

Имя входного файла: `grants.in`  
Имя выходного файла: `grants.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Фонд изучения дикой природы в течение  $t$  лет ежегодно выделяет денежные гранты в поддержку исследований северной фауны. На гранты претендуют три организации, одна из которых занимается изучением тюленей, вторая — оленей, третья — белых медведей.

Для упрощения бухгалтерского учёта фонд принял следующие решения:

- размер любого гранта в денежных единицах должен быть степенью числа 2, то есть равен  $2^k$  для некоторого целого  $k \geq 0$ ;
- все гранты, получаемые одной организацией в одном году, должны иметь различные размеры.

В  $i$ -м году фонд планирует полностью распределить  $n_i$  денежных единиц, выделенных на гранты. Сравнивать результативность использования средств возможно только для грантов одинакового размера, выделенных каждой из трёх организаций. Такие гранты называются целевыми. Распределение денежных единиц на гранты между тремя организациями считается оптимальным, если как можно большая часть общей суммы выделена на целевые гранты.

Например, если в текущем году на все гранты выделено 47 денежных единиц, то оптимальным вариантом распределения будет: выделить каждой из организаций целевые гранты размерами по 2 и 8 денежных единиц, что составит в сумме 30 единиц. Остальные 17 единиц можно распределить, например, выделив первой организации 16 денежных единиц, а третьей — 1 денежную единицу. Выделить более 30 денежных единиц на целевые гранты, распределяя 47 денежных единиц, нельзя.

Требуется написать программу, которая по заданной в  $i$ -м году общей сумме грантов  $n_i$  определяет, сколько денежных единиц следует выделить каждой из трёх организаций при оптимальном распределении грантов.

### Формат входных данных

В первой строке входных данных записано целое число  $t$  — количество лет ( $1 \leq t \leq 100$ ). В каждой из последующих  $t$  строк записано целое число  $n_i$  — общая сумма грантов, которую необходимо полностью распределить в  $i$ -м году.

### Формат выходных данных

Выходные данные должны содержать  $t$  строк по три целых числа в каждой — суммы грантов, которые следует выделить каждой из трёх организаций в соответствующий год. Если оптимальных вариантов распределения несколько, необходимо вывести любой из них.

### Пример

<code>grants.in</code>	<code>grants.out</code>
3	4 0 0
4	7 7 7
21	27 10 10
47	

## Задача I. Лосяш и дерево

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

В одной научной статье Лосяш увидел необычное дерево из  $N$  вершин, в каждой вершине записано одно число — 0 или 1.

Изучая интересные свойства этого дерева, Лосяш задался целью найти такой путь в этом дереве, что *xor* значений в вершинах, лежащих на этом пути равен *xor* значений в вершинах, которые являются соседними к какому-либо из концов пути, но не лежат на пути.

Помогите ему найти его жизненный путь!

Обратите внимание, что не запрещается выводить путь, концы которого совпадают.

### Формат входных данных

В первой строке вводится целое число  $N$  ( $2 \leq N \leq 200\,000$ ).

В следующей строке содержатся  $N$  целых чисел  $a_1, \dots, a_N$  — значения, записанные в вершинах ( $a_i = 0$  или 1)

В каждой из следующих  $N - 1$  строк вводится два числа  $u_i, v_i$  — очередное ребро дерева ( $1 \leq u_i, v_i \leq N$ ).

### Формат выходных данных

Если подходящего пути не существует, то выведите  $-1$ , иначе первая строка должна содержать одно целое число  $M$  ( $1 \leq M \leq N$ ) — количество вершин в пути.

В следующей строке должны быть записаны  $M$  целых чисел — вершины в пути в порядке обхода ( $1 \leq v_i \leq N$ )

### Примеры

стандартный ввод	стандартный вывод
4 0 1 1 1 1 2 1 3 1 4	3 2 1 3
2 0 1 1 2	-1
4 1 0 0 1 2 1 3 2 4 2	1 2

## Задача J. Большой рюкзак

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.75 секунд  
Ограничение по памяти: 4 мегабайта

Вам дано  $N$  предметов веса  $w_1, w_2, \dots, w_n$  и число  $S$ . Скажите, возможно ли набрать суммарный вес  $S$  взяв каждый из данных предметов не более одного раза.

### Формат входных данных

В первой строке даны два целых числа  $N$  и  $S$  ( $1 \leq N \leq 40\,000, 1 \leq S \leq 100\,000$ ) — количество весов и нужный вес соответственно.

В следующей строке через пробел вводятся сами веса  $w$  ( $1 \leq w_i \leq 20\,000$ ).

### Формат выходных данных

Если набрать нужный вес возможно, выведите «YES», иначе выведите «NO»

### Пример

стандартный ввод	стандартный вывод
8 25 7 13 15 2 4 6 9 14	YES