

## Задача А. 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 В. 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 без них не может. И вот, короткая 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 поочередного 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 С. 17 стульев

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

Остап Бендер снова пытается получить причитающиеся драгоценности, но на этот раз они были заперты в шкатулке, для открытия которой необходимо иметь  $N$  ключей. По закономерной случайности каждый из ключей был спрятан в одном из  $N$  стульев, распроданных на недавнем аукционе. После аукциона эти стулья были развезены в  $N$  городов.

И вот теперь Остап решил на новую безумную затею: заехать в каждый из городов и, провернув в каждом из них аферу, выкрасть необходимые ключи. Чтобы избежать конфликтов с недоброжелателями, Остап не хочет больше одного раза появляться в каком-либо городе. Также у Остапа есть список цен за проезд между каждой парой городов. Изначально Остап находится в городе под номером 1 и после посещения всех городов может незаметно скрыться из этой страны.

Помогите Остапу найти порядок посещения городов, при котором ему потребуется потратить как можно меньше средств на странствия, и тогда, возможно, он поделится с Вами добытыми бриллиантами.

### Формат входных данных

Первая строка содержит единственное число  $N$  — количество городов ( $1 \leq N \leq 17$ ).

Следующие  $N$  строк содержат по  $N$  целых неотрицательных чисел.  $j$ -тое число в  $i$ -той строке означает стоимость проезда из города  $i$  в город  $j$  ( $0 \leq a_{ij} \leq 100$ ). Если  $a_{ij} > 0$ , то проезд стоит  $a_{ij}$  рублей, иначе — это означает, что из города  $i$  в  $j$  невозможно проехать напрямую.

### Формат выходных данных

В первой строке выведите минимальную сумму денег, необходимую для посещения всех городов Остапом. В следующей строке выведите  $N$  чисел — порядок посещения городов, при котором эта сумма достигается. Если затею Остапа невозможно вывести, то в единственной строке выходного файла выведите число -1.

### Примеры

<code>trader.in</code>	<code>trader.out</code>
3 0 3 2 3 0 6 2 6 0	8 1 3 2
5 0 6 4 0 0 6 0 7 0 7 4 7 0 0 0 0 0 0 0 2 0 7 0 2 0	20 1 3 2 5 4

## Задача D. Раскраска графа

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

Дан граф из  $n$  вершин, раскрасьте его в минимально возможное число цветов так, чтобы никакие две вершины, соединенные ребром, не были одного цвета.

### Формат входных данных

В первой строке содержится число  $t$  — количество тестовых примеров ( $1 \leq t \leq 5$ ).

Далее содержится  $t$  тестовых случаев, заданных в следующем формате:

В первой строке записаны числа  $n$  и  $m$  — количество вершин и ребер соответственно ( $1 \leq n \leq 17$ ,  $0 \leq m \leq \frac{n \cdot (n-1)}{2}$ ).

Затем идет  $m$  строк, в которых содержится по два числа  $v_i u_i$ , что означает, что вершины  $v_i$  и  $u_i$  соединены ребром ( $1 \leq v_i, u_i \leq n, v_i \neq u_i$ ).

Гарантируется, что все ребра в каждом тестовом случае различны.

### Формат выходных данных

Для каждого тестового случая в первой строке выведите минимальное число цветов  $k$ .

Во второй строке выведите  $n$  чисел  $a_i$  — цвета вершин ( $1 \leq a_i \leq k$ ).

### Примеры

стандартный ввод	стандартный вывод
3	3
3 3	3 2 1
1 2	2
2 3	1 2 2 1 1
3 1	3
5 3	1 3 1 1 2 1
2 1	
3 1	
4 2	
6 7	
1 2	
1 5	
2 5	
2 3	
2 4	
5 6	
5 4	

## Задача Е. Транзитивное замыкание

Имя входного файла: floyd32.in  
Имя выходного файла: floyd32.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Дан ориентированный граф. Найдите его транзитивное замыкание, то есть для каждой пары вершин  $a, b$  определите, есть ли путь из  $a$  в  $b$ .

### Формат входных данных

На первой строке число вершин  $n$  ( $1 \leq n \leq 1\,000$ ). Следующие  $n$  строк имеют длину  $n$ , состоят из нулей и единиц и задают матрицу смежности графа. Единица в  $i$ -й строке,  $j$ -м столбце обозначает ребро из  $i$  в  $j$ .

### Формат выходных данных

Выведите матрицу смежности транзитивного замыкания данного графа.

### Примеры

floyd32.in	floyd32.out
3	011
010	001
001	000
000	

## Задача F. Поиск подстроки

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

Найти позиции всех вхождений строки  $T$  в строку  $S$ .

### Формат входных данных

Первые две строки входных данных содержат строки  $S$  и  $T$ , соответственно. Длины строк больше 0 и меньше 50000, строки содержат только латинские буквы.

### Формат выходных данных

Выведите номера символов, начиная с которых строка  $T$  входит в строку  $S$ , в порядке возрастания.

### Примеры

стандартный ввод	стандартный вывод
ababbababa aba	0 5 7

## Задача G. Фонари

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

В детском лагере «Берендеевы поляны» вряд ли найдётся кто-нибудь, кто не побоялся бы сходить ночью на хутор. В связи с этим администрация лагеря установила вдоль дороги  $N$  фонарей. Лампочки в фонарях могут перегорать, и, хотя в «Берендеевых полянах» есть электрик, он может заболеть и не успеть заменить перегоревшую лампочку на новую до приезда очередной комиссии.

Каждая посещающая лагерь комиссия считает своим долгом проинспектировать состояние фонарей, освещающих дорогу на хутор, но дорога эта очень длинная, поэтому каждый раз членов комиссии подвозят на машине к какому-нибудь фонарю, они выходят, идут пешком до другого фонаря, осматривая по пути все фонари, и там садятся в машину и уезжают.

Если они замечают хотя бы один перегоревший фонарь, в «Берендеевых полянах» устраивается штрафное посвящение для преподавателей.

По информации о происходящих событиях для каждого приезда комиссии выведите результат осмотра фонарей.

### Формат входных данных

В первой строке входного файла записаны два числа  $N$  и  $k$  ( $1 \leq N, k \leq 100\,000$ ) — количество фонарей и количество событий. Далее следуют  $k$  строк, описывающих события. События описываются следующим образом:

- перегорание фонаря задаётся двумя числами, первое из которых равно  $-1$ , а второе задаёт номер перегоревшего фонаря (от 1 до  $N$ );
- замена лампочки электриком задаётся двумя числами, первое из которых равно 1, а второе задаёт номер заменённой лампочки (от 1 до  $N$ );
- приезд комиссии задается тремя числами, первое из которых равно 0, а два других ( $a$  и  $b$ ) задают отрезок дороги, по которому проходит комиссия. А именно, члены комиссии видят все фонари с номерами от  $a$  до  $b$  включительно и только их. Гарантируется, что  $1 \leq a \leq b \leq N$ .

Изначально все фонари исправны.

### Формат выходных данных

Для каждого приезда комиссии выведите в выходной файл одну строку. А именно, если члены комиссии не заметят ни одного перегоревшего фонаря, то выведите «PASSED», иначе — «PENALTY».

### Пример

streetlamps.in	streetlamps.out
5 6	PASSED
-1 2	PENALTY
0 1 1	PASSED
0 2 5	
-1 4	
1 2	
0 2 3	

## Задача Н. Перестановки strike back

Имя входного файла: permutation2.in  
Имя выходного файла: permutation2.out  
Ограничение по времени: 5 секунд  
Ограничение по памяти: 256 мегабайт

Вася выписал на доске в каком-то порядке все числа от 1 по  $N$ , каждое число ровно по одному разу. Иногда он стирает какое-то число и записывает на его место другое. Количество чисел, выписанных Васей, оказалось довольно большим, поэтому Вася не может окинуть взглядом все числа. Однако ему надо всё-таки представлять эту последовательность, поэтому он написал программу, которая в любой момент отвечает на вопрос — сколько среди чисел, стоящих на позициях с  $x$  по  $y$ , по величине лежат в интервале от  $k$  до  $l$ . Сделайте то же самое.

### Формат входных данных

В первой строке лежит два натуральных числа —  $1 \leq N \leq 100\,000$  — количество чисел, которые выписал Вася и  $1 \leq M \leq 100\,000$  — суммарное количество вопросов и изменений сделанных Васей. Во второй строке дано  $N$  чисел — последовательность чисел, выписанных Васей. Далее в  $M$  строках находятся описания вопросов. Каждый запрос на изменение числа в некоторой позиции начинается со слова SET и имеет вид SET a b ( $1 \leq a \leq N$ ,  $1 \leq b \leq N$ ). Это означает, что Вася изменил число, записанное в позиции  $a$  на число  $b$ . Каждый Васин вопрос начинается со слова GET и имеет вид GET x y k l ( $1 \leq x \leq y \leq N$ ,  $1 \leq k \leq l \leq N$ ).

### Формат выходных данных

Для каждого Васиного вопроса выведите единственное число — ответ на Васиный вопрос.

### Примеры

permutation2.in	permutation2.out
4 4	1
1 2 3 4	3
GET 1 2 2 3	2
GET 1 3 1 3	
SET 1 4	
GET 1 3 1 3	

## Задача I. 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 J. 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 К. 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 L. 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