

Задача А. К-ичные числа

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

По данным натуральным n и k выведите все последовательности длины n , составленные из символов $0 \dots k - 1$, в лексикографическом порядке.

Формат входных данных

Во входном файле заданы два числа — n и k ($2 \leq k, n \leq 10$).

Формат выходных данных

Каждая последовательность должна выводиться в отдельной строке, вывод должен завершаться символом новой строки. Числа, входящие в последовательность, должны быть разделены одним пробелом.

Пример

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

Задача В. Сочетания-1

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

По данным натуральным n и k выведите все двоичные последовательности длины n , содержащие ровно k единиц в лексикографическом порядке.

Формат входных данных

Входной файл содержит два числа, n и k ($1 \leq n \leq 100, 0 \leq k \leq n$).

Формат выходных данных

Каждая последовательность должна выводиться в отдельной строке, вывод должен завершаться символом новой строки. Числа, входящие в последовательность, должны быть разделены одним пробелом. Гарантируется, что количество чисел в выходном файле не превосходит 200000

Пример

стандартный ввод	стандартный вывод
4 2	0 0 1 1 0 1 0 1 0 1 1 0 1 0 0 1 1 0 1 0 1 1 0 0

Задача С. Сочетания-2

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

По данным натуральным k и n ($1 \leq k \leq n$) выведите все **убывающие** последовательности длины k состоящие из чисел $1 \dots n$ в лексикографическом порядке.

Формат входных данных

Во входном файле два числа — k и n ($1 \leq k \leq n \leq 1000$).

Формат выходных данных

Каждая последовательность должна выводиться в отдельной строке, вывод должен завершаться символом новой строки. Числа, входящие в последовательность, должны быть разделены одним пробелом. Гарантируется, что количество чисел в выходном файле не превосходит 500 000.

Пример

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

Задача D. Перестановки

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

Дано натуральное число n . Выведите всевозможные перестановки чисел от 1 до n в **обратном** лексикографическом порядке.

Формат входных данных

Во входном файле одно число — n ($1 \leq n \leq 9$).

Формат выходных данных

Каждая последовательность должна выводиться в отдельной строке, вывод должен завершаться символом новой строки. Числа, входящие в последовательность, должны быть разделены одним пробелом.

Пример

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

Задача Е. Без двух единиц подряд

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

По данному натуральному числу n выведите все двоичные последовательности длины n , не содержащие двух единиц подряд, в лексикографическом порядке.

Формат входных данных

Одно натуральное число n ($n \leq 26$).

Формат выходных данных

Каждая последовательность должна выводиться в отдельной строке, вывод должен завершаться символом новой строки. Числа, входящие в последовательность, должны быть разделены одним пробелом.

Пример

стандартный ввод	стандартный вывод
3	0 0 0 0 0 1 0 1 0 1 0 0 1 0 1

Замечание

Реализовать рекурсивно.

Задача F. Правильные скобочные последовательности

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

Дано натуральное число n . Выведите каждую k -ю правильную скобочную последовательность, состоящую из n открывающих круглых скобок и n закрывающих скобок в лексикографическом порядке.

Формат входных данных

Во входном файле два числа — n и k ($1 \leq n \leq 15$, $1 \leq k \leq 100$).

Формат выходных данных

Каждая последовательность должна выводиться в отдельной строке, вывод должен завершаться символом новой строки.

Примеры

стандартный ввод	стандартный вывод
3 1	((())) (()()) (())() ()(()) ()()()
3 2	(()()) ()(())

Задача G. Разбиения на слагаемые

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

Перечислите все разбиения целого положительного числа N на целые положительные слагаемые. Разбиения должны обладать следующими свойствами:

- Слагаемые в разбиениях идут в невозрастающем порядке.
- Разбиения перечисляются в лексикографическом порядке.

Формат входных данных

Во входном файле находится единственное число N ($1 \leq N \leq 40$).

Формат выходных данных

В выходной файл выведите искомые разбиения по одному на строку.

Пример

стандартный ввод	стандартный вывод
4	1 1 1 1 2 1 1 2 2 3 1 4

Задача Н. Монетки

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

64 мегабайта

В Волшебной стране используются монетки достоинством $A_1, A_2, \dots, A_M$. Волшебный человечек пришел в магазин и обнаружил, что у него есть ровно по две монетки каждого достоинства. Ему нужно заплатить сумму N . Напишите программу, определяющую, сможет ли он расплатиться без сдачи.

Формат входных данных

Сначала вводится целое число N ($1 \leq N \leq 10^9$), затем — целое число M ($1 \leq M \leq 16$) и далее M попарно различных целых чисел $A_1, A_2, \dots, A_M$ ($1 \leq A_i \leq 10^9$).

Формат выходных данных

Выведите сначала K — количество монет, которое придется отдать Волшебному человечку, если он сможет заплатить указанную сумму без сдачи. Далее выведите K чисел, задающих достоинства монет. Если решений несколько, выведите вариант, в котором Волшебный человек отдаст наименьшее возможное количество монет. Если таких вариантов несколько, выведите любой из них.

Если без сдачи не обойтись, то выведите одно число 0. Если же у Волшебного человечка не хватит денег, чтобы заплатить указанную сумму, выведите одно число -1 (минус один).

Примеры

стандартный ввод	стандартный вывод
5 2 1 2	3 2 2 1
7 2 1 2	-1
5 2 3 4	0

Задача I. Мирные ферзи

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

Известно, что на шахматной доске размером 8×8 можно расставить 8 ферзей не бьющих друг друга, причем сделать это можно 92 способами.

Дано натуральное n . Определите сколькими способами на доске $n \times n$ можно расставить n мирных ферзей.

Формат входных данных

Во входном файле содержится одно число n ($1 \leq n \leq 12$).

Формат выходных данных

Выведите единственное число — ответ на задачу.

Пример

стандартный ввод	стандартный вывод
8	92

Задача J. Старый компьютер

Имя входного файла: `oldcomputer.in`
Имя выходного файла: `oldcomputer.out`
Ограничение по времени: 5 секунд
Ограничение по памяти: 64 мегабайта

У Темы очень старый компьютер с очень странной операционной системой. Она настолько странная, что окна на экране не могут пересекаться, но могут быть вложены.

Будем говорить, что окно A вложено в окно B , если все клеточки (пиксели) окна B находятся внутри окна A .

Также будем говорить, что глубина вложенности окон на экране равна X , если существует последовательность окон $a_1, a_2, a_3 \dots a_x$, такая что a_1 вложено в a_2 , a_2 вложено в a_3 , и так далее, а в конце a_{x-1} вложено в a_x , ну и a_x никуда не вложено. И, конечно, не существует более длинной последовательности, которая удовлетворяет таким же требованиям.

Однажды сестра Темы открыла на его компьютере очень много игр в оконном режиме. Все бы было хорошо, но, когда Тема захотел поиграть в свою новую любимую игру, он вспомнил, что, если максимальная вложенность окон больше K и он запустит игру, то компьютер взорвется. Теперь он хочет узнать, какова же текущая глубина вложенности, чтобы закрыть лишние окна.

Формат входных данных

Вам задана карта экрана. Карта представляет из себя табличку $N \times N$ ($N \leq 500$), и про каждый пиксель известны два числа: a и b . Если этот пиксель является верхней левой точкой какого-то окна, то a это высота этого окна, а b это его ширина. a и b заданы в пикселях. Если же пиксель не является левой верхней клеткой, какого-то окна, то $a = b = -1$

В первой строке входного файла содержится число N . В следующих N строках содержится N пар записанных через слэш чисел $a_1/b_1 \ a_2/b_2 \dots a_n/b_n$

Формат выходных данных

Выведите одно число — глубину вложенности окон на экране.

Пример

<code>oldcomputer.in</code>	<code>oldcomputer.out</code>
3 2/2 -1/-1 -1/-1 -1/-1 1/1 -1/-1 -1/-1 -1/-1 -1/-1	2

Задача К. Шнурки

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

На ЛКШ обрушился очередной дождь. Вся обувь намокла и испачкалась, и дети устроили массовую чистку своей обуви. Перед тем, как погладить шнурки от всех кроссовок, ЛКШата постирали их и развесили сушиться в один ряд. Какие-то шнурки при этом унесло ветром, так что осталось ровно 3^n шнурков.

Девочка Марина, увидевшая висющие шнурки, задумала маленькую шалость. Она разделила шнурки на группы по три штуки (первые три шнурка, вторые три шнурка и т.д.), и в каждой группе сплела косичку. Марина заплетает косичку из трёх шнурков следующим классическим способом: сначала она меняет местами левый шнурок со средним, затем правый со средним, потом опять левый со средним, потом правый со средним и так далее, проводя эти операции суммарно k раз.

На достигнутом девочка Марина решила не останавливаться. Уже сплетённые косички она вновь разделила на группы по три штуки, таким образом в каждой группе оказалось по девять шнурков — три косички, заплетённые на первом этапе. Внутри каждой группы она заплела косичку из косичек по такому же алгоритму, что и прежде: сначала меня левую косичку со средней, затем правую со средней, потом левую со средней и так далее, тоже ровно k раз. Когда Марина меняла две косички местами, положение шнурков внутри косичек не менялось.

Как можно догадаться, и этого девочке Марине не хватило. На следующем шаге она сплела ещё большие косички из косичек поменьше, полученных на предыдущем шаге. Эту процедуру она повторила n раз, пока у неё не получилась одна большая супермегакосичка.

Владельцы кроссовок почему-то разозлились на Марину за запутанные шнурки, хотя получившаяся супермегакосичка им очень понравилась. Они отвлеклись от своих кроссовок и стали изучать, какие последовательности шнурков вообще можно было получить таким образом.

Помогите ребятам. Для заданной последовательности цветов шнурков перед шалостью девочки Марины укажите, в каком порядке шнурки окажутся в конце описанной процедуры.

Формат входных данных

В первой строке содержится целое число n ($1 \leq n \leq 10$) — число повторений. Во второй строке содержится целое число k ($0 \leq k \leq 10^9$) — количество обменов, которое делает Марина, заплетая очередную косичку. Затем идёт последняя строка из маленьких букв латинского алфавита длины 3^n . В этой строке i -й символ означает цвет i -го шнурка перед началом описанной процедуры.

Формат выходных данных

Выведите строку, соответствующую цветам шнурков после того, как Марина заплела супермегакосичку.

Примеры

стандартный ввод	стандартный вывод
1 2 rgb	gbr
2 5 aaaaaaaab	aaaabaaaa

Задача L. Доказательство в HOL

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

Студент Дима учится на втором курсе университета. На втором курсе в его университете читается курс математической логики. В этом курсе особое внимание акцентируется на *автоматических доказательствах* — программах, позволяющих с их помощью доказывать различные сложные теоремы.

Курсовая работа по математической логике у Димы такая: необходимо доказать с помощью автоматического доказателя *HOL*, что шахматную доску размером $2^N \times 2^N$, из которой вырезана одна клетка, можно покрыть в один слой уголками из трех клеток.

Дима не верит в то, что это правда, и пытается составить контрпример.

Ваша задача — доказать Диме, что он неправ, и решить задачу для Диминых входных данных.

Формат входных данных

В первой и единственной строке находятся три числа N , X , Y , $1 \leq N \leq 6$, $1 \leq X \leq 2^N$, $1 \leq Y \leq 2^N$.

Этими числами задана доска $2^N \times 2^N$, из которой вырезана клетка с координатами (X, Y) . X — координата по горизонтали, Y — по вертикали, $(1, 1)$ — верхний левый угол доски.

Формат выходных данных

Каждый уголок характеризуется своим уникальным номером. Уголки пронумерованы начиная с единицы, без пропусков.

В выходной файл выведите 2^N строк по 2^N чисел — номера уголков, покрывающих соответствующие клетки.

Вырезанную клетку обозначьте нулем.

Система оценки

Тесты, в которых $N = 2$ стоят 12 баллов.

Тесты, в которых $N = 3$ стоят 12 баллов.

Тесты, в которых $N = 4$ стоят 28 баллов.

Тесты, в которых $N = 5$ стоят 28 баллов.

Тесты, в которых $N = 6$ стоят 20 баллов.

Примеры

стандартный ввод	стандартный вывод
1 1 1	0 1 1 1
2 2 2	2 2 4 4 2 0 1 4 3 1 1 5 3 3 5 5