

Задача А. Перестановки

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

Выведите все перестановки чисел от 1 до N в лексикографическом порядке.

Формат входных данных

Во входном файле задано число N ($1 \leq N \leq 8$).

Формат выходных данных

Выведите все перестановки, разделяя числа пробелами, по одной в каждой строке.

Примеры

<code>permutations.in</code>	<code>permutations.out</code>
3	1 2 3 1 3 2 2 1 3 2 3 1 3 1 2 3 2 1

Задача В. Сочетания

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

Даны числа n и k . Выведите в выходной файл все сочетания по k из чисел от 1 до n в лексикографическом порядке.

Формат входных данных

Во входном файле заданы числа n и k , $1 \leq k \leq n \leq 16$.

Формат выходных данных

Выведите все сочетания, разделяя числа пробелами, по одному в каждой строке.

Примеры

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

Задача С. 30 кресел

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

Потерпев фиаско в погоне за стульями Остап не пал духом, а ввязался в новую авантюру. Услышав, что неподалёку на аукционе распродают n старинных кресел, не долго думая он решил попытать удачу и проверить: не скрываются ли сокровища в одном из них. Придя на торги, Остап понял, что денег у него хватит на выкуп ровно k из n кресел. Своим самым счастливым числом Остап считает число m , поэтому он снова обращается к Вам за помощью и просит выбрать m -е сочетание k из n кресел.

Формат входных данных

Во входном файле заданы числа n , k и m . $1 \leq k \leq n \leq 30$, $0 \leq m \leq \binom{n}{k} - 1$.

Формат выходных данных

Выведите в выходной файл в возрастающем порядке номера кресел, входящие в m -е в лексикографическом порядке сочетание по k из чисел от 1 до n . Сочетания занумерованы, начиная с 0.

Примеры

<code>choose.in</code>	<code>choose.out</code>
4 2 3	2 3

Задача D. Как убить время

Имя входного файла: num2part.in
Имя выходного файла: num2part.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Некто Симон уже давно дорешал все задачи из констестов и теперь убивает время на практике тем, что раскладывает найденные у комповника n камешков в кучки разного размера. Изначально он разложил все камешки по одному. В каждую следующую минуту он выкладывает из них следующее лексикографическое разбиение на кучки. Порядок разбиений будем определять следующим образом: рассмотрим все разбиения n камешком на кучки, в каждом разбиении упорядочим их в порядке не убывания. Будем считать, что разбиение $a_1 + a_2 + \dots + a_n$ лексикографически меньше $b_1 + b_2 + \dots + b_m$, если для некоторого $k \forall j \leq k : a_j = b_j$ и либо $k = n$, либо $a_{k+1} < b_{k+1}$.

Формат входных данных

Во входном файле заданы числа n и r . $1 \leq n \leq 100$, разбиение с номером r — существует.

Формат выходных данных

Выведите r -ое разбиение n камешков на кучки, разбиения нумеруются с 0.

Примеры

num2part.in	num2part.out
4 3	2+2
5 5	2+3

Задача Е. Номер по правильной скобочной последовательности

Имя входного файла: `brackets2num.in`
Имя выходного файла: `brackets2num.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Дана правильная скобочная последовательность, содержащая N открывающих скобок. Найдите её номер в лексикографическом порядке среди всех правильных скобочных последовательностей с таким же количеством открывающихся скобок, учитывая, что «(» < «)». Последовательности нумеруются с 0.

Формат входных данных

Во входном файле задана строка, являющаяся правильной скобочной последовательностью, $1 \leq N \leq 20$.

Формат выходных данных

Выведите единственное число — номер последовательности.

Примеры

<code>brackets2num.in</code>	<code>brackets2num.out</code>
<code>((()))()</code>	3

Задача F. Задобрите Инти

Имя входного файла: `makeintihappy.in`
Имя выходного файла: `makeintihappy.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Как вы помните, племя Майя вымерло, племя Йюля вынуждено было распастся и покинуть обжитые территории, и вот теперь племени Авгуйя тоже угрожает опасность. Задоблив бога дождя Тлалока, они напроць забыли про бога солнца Инти, который будучи обделённым вниманием грозиться ниспослать испепеляющую жару, если жители племени не порадуют его божественный взор новой коллекцией симпатичных узоров, выложенных из огромных гранитных плит, покрашенных в чёрный и белый цвета. Напомним, что симпатичным узором называется прямоугольник шириной W и высотой H ни какой квадрат 2 на 2 которого не покрашен в один цвет.

В данной задаче таскать гранитные плиты Вам не придётся, однако потребуется определить K -й лексикографически симпатичный узор. Один узор будем считать лексикографически меньше другого, если при просмотре узоров по столбцам сверху вниз найдётся такая позиция, что все плиты рассмотренные ранее совпадают попарно по цвету, а плита на данной позиции у первого узора белая, а у второго — чёрная.

Формат входных данных

Во входном файле заданы числа W , H и K . ($1 \leq W \leq 1000$, $1 \leq H \leq 10$, $1 \leq K \leq 10^{18}$)

Формат выходных данных

Выведите требуемый симпатичный узор, обозначая белые плиты символом $'w'$, а чёрные — символом $'b'$. Если не существует узора с таким номером — выведите "Impossible" (без кавычек).

Примеры

<code>makeintihappy.in</code>	<code>makeintihappy.out</code>
5 1 17	bwww
7 4 1234567	wbwbwb wbbwwb wwwbbwb wbwbwb