

Содержание

Задачи на перебор	2
Задача 2A. Функция [1, 256]	2
Задача 2B. Скобки [1, 256]	3
Задача 2C. Перестановки 2 [1, 256]	4
Задача 2D. Поколение комбинаторов [1, 256]	5
Задача 2E. Сочетания [1, 256]	6
Задача 2F. Сочетания и ничего лишнего [1, 256]	7
Задача 2G. Анти-Фибоначчи [1, 256]	8
Задача 2H. Не похоже ни на кого [1, 256]	9
Задача 2I. Различные разбиения [1, 256]	10
Задача 2J. Продавец аквариумов [1, 256]	11
Задача 2K. Запрещённые подстроки [1, 256]	12
Задача 2L. Таблицы Юнга [1, 256]	13
Задача 2M. Пути на доске [1, 256]	14
Задача 2N. Покрытие доминошками [1, 256]	15
Задача 2O. Длинные домино [1, 256]	16

Задачи на перебор

Задача 2А. Функция [1, 256]

Вычислите функцию: $f(n) = \begin{cases} 1 & \text{если } n \leq 2 \\ f(\lfloor 6 * n/7 \rfloor) + f(\lfloor 2 * n/3 \rfloor) & \text{если } n \bmod 2 = 1 \\ f(n - 1) + f(n - 3) & \text{если } n \bmod 2 = 0 \end{cases}$

Формат входных данных

Входные данные содержат натуральное число n ($1 \leq n \leq 10^{12}$).

Формат выходных данных

Выведите значение функции по модулю 2^{32} .

Пример

stdin	stdout
7	10

Задача 2В. Скобки [1, 256]

Правильная скобочная последовательность (ПСП) определяется следующим образом:

1. Пустая строка является ПСП.
2. Если S — ПСП, то (S) также является ПСП.
3. Если A и B являются ПСП, то и AB (конкатенация A и B) также является ПСП.

Напишите программу, выводящую все ПСП заданной длины.

Формат входных данных

В единственной строке входного файла записано целое число n . $1 \leq n \leq 10$.

Формат выходных данных

Выведите по одной строке для каждой ПСП длины $2n$ (из n пар скобок). Порядок не имеет значения.

Пример

stdin	stdout
1	()
2	()() (())

Задача 2С. Перестановки 2 [1, 256]

Во входном файле задано число n ($1 \leq n \leq 9$). Выведите в выходной файл в лексикографическом порядке все перестановки чисел от 1 до n .

Пример

stdin	stdout
3	1 2 3 1 3 2 2 1 3 2 3 1 3 1 2 3 2 1

Задача 2D. Поколение комбинаторов [1, 256]

Сочетанием из n элементов по k называется убывающая последовательность из k чисел из диапазона от 1 до n .

Сгенерируйте все сочетания из n элементов по k в антилексикографическом порядке, т.е. так, что для любых двух выведенных сочетаний первые l чисел равны, а $l+1$ -е в предыдущем больше, чем в следующем.

Формат входных данных

Во входном файле содержатся два целых числа n и k . $1 \leq k \leq n \leq 15$.

Формат выходных данных

В выходной файл выведите все сочетания из n элементов по k в антилексикографическом порядке, по одному сочетанию на строку.

Пример

stdin	stdout
3 2	3 2 3 1 2 1

Задача 2Е. Сочетания [1, 256]

Сочетанием из n элементов по k называется возрастающая последовательность из k чисел из диапазона от 1 до n .

Во входном файле заданы числа n и k . Выведите в выходной файл все сочетания по k из чисел от 1 до n в лексикографическом порядке. $1 \leq k \leq n \leq 16$.

Пример

stdin	stdout
4 2	1 2 1 3 1 4 2 3 2 4 3 4

Задача 2F. Сочетания и ничего лишнего [1, 256]

По данным натуральным x и y ($x \leq y \leq 1000$) выведите все возрастающие последовательности длины x , состоящие из чисел $1 \dots y$.

Формат входных данных

Во входном файле два числа — x и y .

Формат выходных данных

Каждая последовательность должна выводиться в отдельной строке.

Последовательности должны быть выведены в лексикографическом порядке.

Примеры

stdin	stdout
3 5	1 2 3 1 2 4 1 2 5 1 3 4 1 3 5 1 4 5 2 3 4 2 3 5 2 4 5 3 4 5

Задача 2G. Анти-Фибоначчи [1, 256]

Лёше надоели числа Фибоначчи. Всю последнюю неделю, когда он приходил на урок математики или информатики, учителя рассказывали что-то про числа Фибоначчи и задавали на дом задачи про них.

На этой неделе домашнее задание у Лёши — написать программу, которая по заданному целому положительному числу N находит количество способов разбить N на положительные целые слагаемые. Способы, отличающиеся лишь порядком слагаемых, считаются одинаковыми. К примеру, для $N = 4$ это количество способов — 5:

$$\begin{aligned} N &= 4 \\ &= 3 + 1 \\ &= 2 + 2 \\ &= 2 + 1 + 1 \\ &= 1 + 1 + 1 + 1 \end{aligned}$$

Поскольку Лёше не нравятся числа Фибоначчи, он решил написать программу, которая считает только такие разбиения, в которых среди слагаемых нет чисел Фибоначчи. Более того, разбиения, в которых количество слагаемых является числом Фибоначчи, Лёша тоже решил не считать.

Помогите Лёше написать такую программу.

Формат входных данных

В первой строке входного файла записано целое число N ($1 \leq N \leq 50$).

Формат выходных данных

Выведите в выходной файл одно число — количество таких разбиений N на положительные целые слагаемые, что ни какое-либо из слагаемых, ни их количество не являются числами Фибоначчи.

Пример

stdin	stdout
4	0

Задача 2Н. Не похоже ни на кого [1, 256]

Дана строка длины не более 30, состоящая из букв «a» и «b». Найдите число способов разбить её на подстроки так, чтобы все подстроки в разбиении были различны, например для строки «abab» есть следующие разбиения: $\{abab, aba + b, ab + a + b, a + bab, a + ba + b, a + b + ab\}$

Формат входных данных

Строка.

Формат выходных данных

Число.

Примеры

stdin	stdout
abab	6

Задача 21. Различные разбиения [1, 256]

Найдите количество различных разбиений натурального числа n на натуральные слагаемые таких, что для любых двух различных чисел $a \neq b$, входящих в разбиение, верно, что количества чисел a и b в разбиении различны. Разбиения, отличающиеся только порядком слагаемых, различными не считаются.

Например, если $n = 4$, то из пяти возможных разбиений этому условию удовлетворяют все, кроме разбиения на слагаемые 1 и 3: в этом разбиении количество единиц равно количеству троек.

$$\begin{array}{ll} 4 = 1 + 1 + 1 + 1 & 4 \text{ единицы} \\ 4 = 1 + 1 + 2 & 2 \text{ единицы, 1 двойка} \\ 4 = 1 + 3 & 1 \text{ единица и 1 тройка!} \\ 4 = 2 + 2 & 2 \text{ двойки} \\ 4 = 4 & 1 \text{ четвёрка} \end{array}$$

Формат входных данных

В первой строке входного файла записано натуральное число n ($1 \leq n \leq 100$).

Формат выходных данных

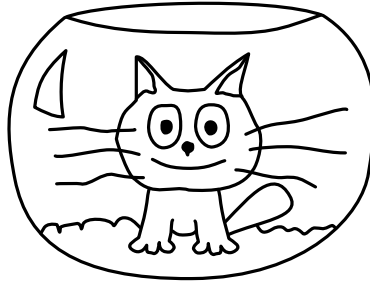
В первой строке выходного файла выведите количество разбиений числа n , удовлетворяющих заданным ограничениям.

Примеры

stdin	stdout
4	4
6	7

Задача 2J. Продавец аквариумов [1, 256]

Продавец аквариумов для кошек хочет объехать n городов, посетив каждый из них ровно один раз. Помогите ему найти кратчайший путь.



Формат входных данных

Первая строка входного файла содержит натуральное число n ($1 \leq n \leq 13$) — количество городов. Следующие n строк содержат по n чисел — длины путей между городами.

В i -й строке j -е число — $a_{i,j}$ — это расстояние между городами i и j ($0 \leq a_{i,j} \leq 10^6$; $a_{i,j} = a_{j,i}$; $a_{i,i} = 0$).

Формат выходных данных

В первой строке выходного файла выведите длину кратчайшего пути. Во второй строке выведите n чисел — порядок, в котором нужно посетить города.

Начинать и заканчивать путь можно в любом городе.

Пример

stdin	stdout
5	666
0 183 163 173 181	1 3 2 5 4
183 0 165 172 171	
163 165 0 189 302	
173 172 189 0 167	
181 171 302 167 0	

Задача 2К. Запрещённые подстроки [1, 256]

Сколько существует строк длины ровно n , состоящих из символов 'a', 'b', 'c', и не содержащих некоторых запрещённых слов?

Формат входных данных

Число n ($1 \leq n \leq 30$) и количество запрещённых подстрок k ($0 \leq k \leq 5$).

Следующие k строк содержат сами строки, их длины не более 5

Формат выходных данных

Число строк по модулю 23917.

Примеры

stdin	stdout
3 1 aa	22
20 2 aa abc	19979

Подсказка по решению

Эту задачу удобно решать рекурсией. А из рекурсии, конечно, можно сделать ленивую динамику. Не бойтесь делать запоминание map-ом, например, `map<pair<int,string>,int>`.

Заметьте, что ответ рекурсии зависит только от последних 5 символов.

Задача 2L. Таблицы Юнга [1, 256]

Нужно найти число способов расставить числа от 1 до N внутри диаграммы Юнга площади N так, чтобы числа внутри каждой строки и каждого столбца возрастали. Каждое число от 1 до N нужно использовать ровно один раз.

$$1 \leq N \leq 50$$

Формат входных данных

Число строк диаграммы Юнга k . Далее k длин строк: $a_1 \geq a_2 \geq \dots \geq a_k$.

Число 1 должно оказаться в первой клетке первой строки.

Первый столбец диаграммы имеет высоту k . Высоты столбцов, также как и длины строк, убывают.

Формат выходных данных

Одно целое число — количество способов расставить числа от 1 до N , где $N = \sum_{i=1}^k a_i$.

Пример

stdin	stdout
2 2 2	2
4 4 3 2 1	768

Задача 2М. Пути на доске [1, 256]

Рассмотрим бесконечную клетчатую доску.

Назовём *путём* из одной клетки в другую последовательность клеток, в которой каждые две идущие подряд клетки являются соседними по стороне. Длина пути — это количество клеток в нём, не считая начальную.

Назовём путь *простым*, если в нём не встречается двух одинаковых клеток.

Зафиксируем какую-то клетку на доске. Сколько существует простых путей заданной длины, начинающихся в этой клетке?

Формат входных данных

В первой строке входного файла задано целое число n ($0 \leq n \leq 22$).

Формат выходных данных

В первой строке выходного файла выведите одно число — количество путей длины n из этой клетки.

Примеры

stdin	stdout
0	1
1	4
2	12

Задача 2N. Покрытие доминошками [1, 256]

Сколько способов покрыть прямоугольник $W \times H$ доминошками?
Каждая клетка должна быть покрыта ровно одной доминошкой.

Формат входных данных

Числа W и H . Ограничения: $W \cdot H \leq 100$, кроме этого $1 \leq W, H$.

Формат выходных данных

Число способов покрыть прямоугольник доминошками.

Пример

stdin	stdout
3 2	3

Задача 20. Длинные домино [1, 256]

Найдите количество способов замостить прямоугольник размера $m \times n$ длинными домино — прямоугольниками размера 3×1 .

Каждое домино должно полностью находиться внутри прямоугольника, домино не должны накладываться.

Формат входных данных

Входной файл содержит m и n ($1 \leq m \leq 9$, $1 \leq n \leq 30$).

Формат выходных данных

Выведите количество способов замостить прямоугольник $m \times n$ длинными домино.

Пример

stdin	stdout
3 3	2
3 10	28