

Содержание

Must have	2
Задача 8А. Скобки [0.1 sec, 256 mb]	2
Задача 8В. Поколение комбинаторов [0.1 sec, 256 mb]	3
Задача 8С. Сочетания [0.1 sec, 256 mb]	4
Обязательные задачи	5
Задача 8D. Перестановки [0.4 sec, 256 mb]	5
Задача 8Е. Сочетания и ничего лишнего [0.1 sec, 256 mb]	6
Задача 8F. Анти-Фибоначчи [0.1 sec, 256 mb]	7
Задача 8G. Не похожие ни на кого [4.0 sec, 256 mb]	8
Задача 8H. Не похожие ни на кого слагаемые [4.0 sec, 256 mb]	9
Задача 8I. Различные разбиения [0.4 sec, 256 mb]	10
Дополнительные задачи	11
Задача 8J. Пути на доске [3.0 sec, 256 mb]	11
Задача 8K. Перестановки [1.5 sec, 256 mb]	12
Задача 8L. Расстановка 8 ферзей [5.0 sec, 256 mb]	13

У вас не получается читать/выводить данные?

Воспользуйтесь примерами (**c++**) (**python**).

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же stdin), вывести ответ нужно в **стандартный поток вывода** (он же stdout).

Обратите внимание на GNU C++ компиляторы с суффиксом inc.

Подни можно пользоваться **дополнительной библиотекой** (optimization.h).

То есть, использовать быстрый ввод-вывод: **пример про числа и строки**.

И быструю аллокацию памяти (ускоряет vector-set-map-весь-STL): **пример**.

Для тех, кто хочет разобраться, как всё это работает.

Короткая версия быстрого ввода-вывода (**тык**) и короткая версия аллокатора (**тык**).

Must have

Задача 8А. Скобки [0.1 сек, 256 mb]

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	()() (())

Подсказка по решению

Это базовая задача на тему «рекурсия».

Задача 8В. Поколение комбинаторов [0.1 сек, 256 mb]

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

Подсказка по решению

Это базовая задача на тему «рекурсия».

Задача 8С. Сочетания [0.1 sec, 256 mb]

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

Подсказка по решению

Возьмите предыдущее решение и подкрутите...

Обязательные задачи

Задача 8D. Перестановки [0.4 сек, 256 mb]

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

Замечание

Мы изучаем рекурсивный перебор.

В этой задаче запрещены нерекурсивные решения.

Но безусловно хорошо, что вы их знаете.

Подсказка по решению

Не забывайте про быстрый вывод.

Не нужно всё это хранить, не тратьте лишнее время, сразу выводите.

Помните давным-давно говорили, что часто создавать новые вектора — плохо?

Задача 8Е. Сочетания и ничего лишнего [0.1 сек, 256 mb]

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

Подсказка по решению

Видите связь с цэшками?

Вообще говоря, таких последовательностей может быть очень много, $C_{1000}^{500} \approx 2^{1000}$, но вас просят их вывести \Rightarrow для данных вам конкретных x и y не так уж много.

Решайте за \mathcal{O} (размера вывода).

Задача 8F. Анти-Фибоначчи [0.1 sec, 256 mb]

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

Подсказка по решению

Всё ещё обычная рекурсия, ничего сложного.

Проверяйте в рекурсии «число ли Фибоначчи» за $O(1)$.

Задача 8G. Не похоже ни на кого [4.0 sec, 256 mb]

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

Подсказка по решению

Обычная рекурсия, не мудрите. Нужна структура данных, чтобы быстро проверять, была ли уже такая подстрока: стандартные подойдут.

p.s. Кстати, помните про быструю аллокацию? Здесь она ускоряет. Чтобы в рекурсивной функции освобождать память, можно трогать `allocator_pos`. Читайте исходники.

Задача 8Н. Не похожие ни на кого слагаемые [4.0 сек, 256 mb]

Сколько способов представить число n в виде суммы различных положительных слагаемых?

Количество слагаемых любое, порядок важен.

Например, для $n = 4$ есть три способа: $4 = 1 + 3$, $4 = 3 + 1$, $4 = 4$.

Формат входных данных

Число n ($1 \leq n \leq 70$).

Формат выходных данных

Число способов.

Примеры

stdin	stdout
4	3
6	11
10	57

Подсказка по решению

Первая идейная задача.

Тут проверяют, поняли ли вы тему «запоминание»?

Напомним прямо здесь: если ответ на задачу возвращает рекурсивная функция, её результат зависит только от параметров, нет смысла запускаться второй раз от тех же параметров, шар в помощь. Не бойтесь получающихся структур.

Задача 81. Различные разбиения [0.4 сек, 256 mb]

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.

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

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

Подсказка по решению

Не мудрите. Тема та же. Действуйте по инструкции.

Если перебирать слагаемые по убыванию, то выйти можно, как только остались только единички.

Дополнительные задачи

Задача 8J. Пути на доске [3.0 sec, 256 mb]

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

Подсказка по решению

В этой задаче всего 22 теста...

Если ваше решение чуть-чуть не проходит ТЛ, это не страшно.

Тем не менее, у жюри есть оптимизация, позволяющая сдать задачу без прекалка.

Задача 8К. Перестановки [1.5 сек, 256 mb]

Саша и Федя играют в интересную игру. У них есть n кубиков, на которых написаны различные числа от 1 до n . Ребята нарисовали на бумаге n клеточек в ряд и играют по следующим правилам.

Сначала первый игрок выставляет некоторые кубики на клеточки, затем второй игрок выставляет на свободные клетки оставшиеся кубики. После этого первый игрок делает следующие действия: он смотрит, какое число написано на последнем кубике (пусть это число a) и после этого переставляет последние a кубиков в обратном порядке. Эти действия первый игрок повторяет до тех пор, пока последним не станет кубик с числом 1.

Например, пусть у ребят пять кубиков. Если первый игрок поставил второй и третий кубик на третье и пятое место: «. . 3 . 2», то второй игрок может расставить оставшиеся кубики так: «41352». В этом случае первому игроку потребуется сделать пять действий: «41325», «52314», «54132», «54123», «54321», после чего игра закончится.

Сейчас первым ходил Саша. Помогите Феде расставить кубики так, чтобы Саша сделал максимально возможное количество действий.

Формат входных данных

Во входном файле содержится число n ($1 \leq n \leq 25$). Следующие n чисел задают расположение кубиков после хода Саши. Число 0 означает, что клетка свободна, число от 1 до n — номер кубика, который стоит в этой клетке. Во входном файле не более 10 нулей.

Формат выходных данных

На первой строке выходного файла выведите максимальное количество действий, которое придется сделать Саше.

На второй строке выведите n чисел от 1 до n , где i -е число означает номер кубика, стоящего в i -ой клетке после хода Феде. Если оптимальных решений несколько, выведите любое.

Пример

stdin	stdout
5 0 0 3 0 2	5 4 1 3 5 2
2 0 0	1 1 2

Подсказка по решению

Попробуйте моделировать. И перебирать по ходу, но максимально лениво.

Задача 8L. Расстановка 8 ферзей [5.0 sec, 256 mb]

Сколько способов на шахматной доске $n \times n$ расставить n ферзей так, чтобы никакие два не били друг друга? Напомним, ферзь бьёт всех по горизонтали, по вертикали, по обеим диагоналям.

Формат входных данных

На первой строке n ($4 \leq n \leq 16$).

Формат выходных данных

Единственное число, количество способов. Способы считаются различными, если множества клеток, занятые ферзями, различаются.

Пример

stdin	stdout
4	2

Замечание

Есть честное решение, укладывающееся в TL.

Пожалуйста, не шлите в эту задачу прекалк. Прекалк нужно было слать в другую.