

Содержание

Задачи здорового человека	2
Задача 7А. Спуск с горы [0.2 sec, 256 mb]	2
Задача 7В. Палиндром [0.2 sec, 256 mb]	3
Задача 7С. Запрещённые подстроки [0.2 sec, 256 mb]	4
Для искателей острых ощущений	5
Задача 7D. Гномы и число топсортот [0.5 sec, 256 mb]	5
Задача 7Е. Коммивояжёр возвращается! [3 sec, 256 mb]	6
Задача 7F. Крышки [1 sec, 256 mb]	7

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

В некоторых задачах большой ввод и вывод. Пользуйтесь **быстрым вводом-выводом**.

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) **переопределение стандартного аллокатора** ускорит вашу программу.

Обратите внимание на GNU C++ компиляторы с суффиксом `inc`, они позволяют пользоваться **дополнительной библиотекой**. Под ними можно сдать **вот это**.

Задачи здорового человека

Задача 7А. Спуск с горы [0.2 сек, 256 mb]

В одном из горнолыжных курортов Италии проводятся соревнования по горнолыжному спуску. Каждому спортсмену предстоит скатиться с горы на лыжах. На любом этапе спуска участник получает определенное число очков. После прохождения трассы очки суммируются. Участник, набирающий наибольшее количество очков, выигрывает. Гора представляет собой треугольник, в качестве элементов которого выступают целые числа — очки за прохождение этапа. На каждом уровне спортсмену предоставляется выбор — двигаться вниз влево или вниз вправо. Начало спуска — в самой высокой точке горы, конец в одной из самых низких.

$$\begin{array}{c} \underline{1} \\ \underline{4} \ 3 \\ 5 \ \underline{6} \ 7 \\ 8 \ \underline{9} \ 0 \ 9 \end{array}$$

Требуется найти максимальное количество очков, которое может набрать спортсмен.

Формат входных данных

Во входном файле содержится целое число n — число этапов ($1 \leq n \leq 100$), далее n строк, каждая из которых характеризует свой уровень. В строке с номером i содержится ровно i целых чисел: a_1, a_2, \dots, a_i ($-100 \leq a_k \leq 100, 1 \leq k \leq i$) — количество очков в каждой из позиций.

Формат выходных данных

В результирующем файле должно находиться искомое целое число.

Пример

stdin	stdout
4	20
1	
4 3	
5 6 7	
8 9 0 9	

Подсказка по решению

Обычная простая динамика.

Задача 7В. Палиндром [0.2 sec, 256 mb]

Палиндромом называется строка, которая читается одинаково как слева направо, так и справа налево. Требуется найти самый длинный палиндром P , получающийся из данной строки S удалением любого (возможно, нулевого) количества символов.

Формат входных данных

Входной файл содержит строчку S , состоящую из строчных латинских букв (a-z). Длина S не превышает 1 000 символов.

Формат выходных данных

Выходной файл должен содержать искомый палиндром. Если таких палиндромов несколько, выведите любой из них.

Примеры

stdin	stdout
anna	anna
perevorot	ror

Подсказка по решению

По подотрезкам. $O(n^2)$.

Задача 7С. Запрещённые подстроки [0.2 sec, 256 mb]

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.

Для искателей острых ощущений

Задача 7D. Гномы и число топсортов [0.5 сек, 256 mb]

Каждый вечер семья гномов ужинает за длинным столом. Все гномы сидят за одной стороной стола в определенном порядке. Разные места ценятся гномами по-разному, поскольку один конец стола находится ближе к кухне, и всем известно, что сидящий ближе к кухне получит еду раньше. Более того, некоторые гномы считают себя более важными, чем некоторые другие, и поэтому должны сидеть ближе к кухне.

Каждый вечер порядок, в котором гномы сидят за столом, должен быть новым. Посчитайте, сколько дней гномы смогут ужинать вместе.

Формат входных данных

Первая строка входного файла содержит два числа n и m ($1 \leq n \leq 19$), n — количество гномов, m — количество зависимостей. Каждая из следующих m строк содержит два целых числа x_i, y_i ($1 \leq x_i, y_i \leq n$); это означает, что гном x_i считает себя более важным, чем гном y_i . Каждая такая зависимость дается не более одного раза. Гномы пронумерованы целыми числами от 1 до n .

Формат выходных данных

Выведите единственное число — количество возможных рассадок гномов за столом.

Примеры

stdin	stdout
3 1 1 2	3

Подсказка по решению

Динамика по масочкам. Простая.

Задача 7Е. Коммивояжёр возвращается! [3 сек, 256 mb]

Коммивояжёр возвращается в систему Альфы Центавра! Население системы с нетерпением ждёт его прибытия — каждый хочет приобрести что-нибудь с далёких планет!

Как обычно, коммивояжёр хочет минимизировать транспортные расходы. Он выбирает начальную планету, прилетает туда на межгалактическом корабле, после чего посещает все остальные планеты системы в порядке, минимизирующем суммарную стоимость посещения, и на другом межгалактическом корабле улетает обратно. Естественно, коммивояжёр не хочет летать ни на какую планету дважды.

Найдите оптимальный маршрут для коммивояжёра. Массы больше не могут ждать!

Формат входных данных

В системе Альфы Центавра n планет. Это число записано в первой строке входного файла ($1 \leq n \leq 19$). Следующие n строк содержат по n чисел каждая: j -ое число на i -ой из этих строк — стоимость перемещения a_{ij} от i -ой планеты до j -ой. Числа в каждой строке разделены пробелами. Числа a_{ii} не несут полезной информации. Все числа во входном файле положительны и не превосходят 10^8 .

Формат выходных данных

В первой строке выходного файла выведите минимальную суммарную стоимость посещения всех планет. Во второй строке выведите n чисел через пробел — номера планет системы в порядке их посещения. Если оптимальных маршрутов несколько, можно вывести любой из них.

Пример

stdin	stdout
3	5
8 1 6	3 1 2
3 5 7	
4 9 2	

Подсказка по решению

Разобрана и на практике, и на лекции. $\mathcal{O}(2^n n^2)$.

Задача 7F. Крышки [1 sec, 256 mb]

У программиста Петрова есть хобби — собирать крышки от пивных бутылок. Ничего странного, он знает еще с сотню программистов, которые очень уважают пиво. Да, они тоже собирают крышки. Не все из них, конечно, но некоторые. Если честно, то часть своих крышек он просто купил, уже без бутылок. Да, это не совсем спортивно, зато коллекция теперь более солидная. Одна вот беда — не хватает ему для полноты коллекции еще нескольких редких крышек. Он даже нашел в Интернете программистов, которые согласны продать их ему. Некоторые даже продают крышки сразу наборами, с большой скидкой. Почему продают, да еще со скидкой? А вы попробуйте объяснить жене, какая польза от пивных крышек. Она же не программист. Осталось выбрать оптимальные предложения. Если объяснить жене зачем надо хранить крышки еще возможно, то почему на них надо тратить деньги — точно не поймет. Поэтому надо купить как можно дешевле. Петров выписал на бумажку все варианты и задумался. Купить сразу все не получится, никаких денег не хватит. Тогда надо купить самые необходимые, но подешевле. Да уж, без программы тут не обойдешься...

Формат входных данных

В первой строке записано число N — количество недостающих Петрову крышек ($1 \leq N \leq 20$). Далее идет N строк — цена, за которую можно купить эту крышку, если покупать ее отдельно. В следующей строке стоит число M ($0 \leq M \leq 100$) — количество предложений по продаже наборов, содержащих нужные ему крышки. Далее идет M строк описывающих эти наборы. В каждой строке первое число — цена набора, второе — количество крышек в наборе, далее перечислены номера крышек (каждый номер от 1 до N), которые в этот набор входят. Номера в наборе не повторяются. Все цены — положительные числа, не превосходящие 2000. В последней строке перечислен минимальный набор крышек, который Петров намерен купить в любом случае.

Формат выходных данных

Выведите минимальную сумму, необходимую Петрову, чтобы купить все крышки из приведенного в последней строке набора.

Примеры

stdin	stdout
4	25
10	
11	
12	
13	
3	
17 2 1 3	
25 3 2 3 4	
15 2 3 4	
3 1 3 4	

Подсказка по решению

Спариваем идеи про маски и про рюкзак. «Рюкзак на множествах».