

1. ПРОИЗВЕДЕНИЕ ЦИФР

Даны два числа A и B ($1 \leq A, B \leq 100000$). Напишите программу нахождения произведения числа A на все цифры числа B .

Входной файл `input.txt` содержит два числа A и B .

В выходной файл `output.txt` вывести одно число – произведение числа A на все цифры числа B .

Входные данные	Результат
3 33	27
111 10	0

2. ШАХМАТНАЯ ДОСКА

Даны координаты двух полей шахматной доски (координаты клетки - это 2 числа от 1 до 8: номер столбца и номер строки)

Одного ли цвета эти клетки на шахматной доске? Вывести в выходной файл сообщение YES, если они одного цвета, и NO иначе

Входные данные	Результат
1 1 2 2	YES
1 1 1 4	NO

3. ШАБЛОНЫ С ? И *

Шаблон называется строка, состоящая из английских букв (a, \dots, z, A, \dots, Z) и символов $?$ и $*$. Каждый из символов $?$ разрешается заменить на одну произвольную букву, а каждый из символов $*$ – на произвольную (возможно пустую) последовательность букв. Про любую строку из букв, которую можно получить из шаблона такими заменами, будем говорить, что она удовлетворяет этому шаблону.

Имеются два шаблона. Требуется найти строку минимальной длины, которая удовлетворяет обоим шаблонам, либо выдать сообщение, что такой строки не существует.

Заданные шаблоны записаны в первых двух строках входного файла. Длина каждого шаблона не превосходит 80 символов.

В выходной файл следует вывести строку минимальной длины, удовлетворяющую обоим шаблонам, либо число -1

Входные данные	Результат
A* *B	AB

4. КОРНИ КВАДРАТНОГО УРАВНЕНИЯ

Кто знает, как решается квадратное уравнение вида $ax^2+bx+c=0$, можете решить эту задачу.

Входной файл содержит три числа a, b, c ($-1000 \leq a, b, c \leq 1000$) – коэффициенты квадратного уравнения.

Выходной файл содержит в одной строке два числа с точностью до 3 знаков в дробной части, если у уравнения два корня (вначале вывести максимальный корень, а потом минимальный), одно число, если один корень и число -1, если нет корней.

Входные данные	Результат
1 2 3	-1
56 98 23	-0.279 -1.471

5. ПЕРЕВОД РИМСКИХ ЧИСЕЛ В ДЕСЯТИЧНЫЕ ЧИСЛА

Разработать программу перевода римских чисел в десятичную систему счисления. Кто забыл, какие бывают римские цифры напомним (I – это 1, II – это 2, III – понятно, что 3, IV – вы правы – это 4, V – уф, а это 5, X – кто не знает – это 10, L – а это кажется 50, C – точно знаю, что это 100, D – как пить дать 500, M – а это «штука», т.е. 1000).

Входной файл содержит одну строку **римское число S**, длиной не более 255 символов.

В выходной файл вам нужно вывести одно десятичное число. (Кто впервые знаком с программированием учтите, что число может перевалить за тип integer. Поэтому к «черту» этот тип, а лучше longint).

P.S.

А еще забыл сказать самое главное. Если перед большой римской цифрой стоит цифра, которая меньше него, то как вы понимаете нужно быть очень внимательным (что там такое было уж не помню).

Входные данные	Результат
XX	10
IVIXLXCCDCMIVXLCDM	2887

6. ВНИМАТЕЛЬНЫЙ ЧИТАТЕЛЬ КОЛЯ СЕРГЕЕВ ИЗ РЛИ

Ученик 10 класса РЛИ Сергеев Николай очень любит читать книги по программированию. При внимательном просмотре одной из страниц он обнаружил, что на одной странице помещается ровно A строк. Таким образом, на 1-й странице печатаются строки с 1-й по A -ю, на второй — с $(A+1)$ -й по $(A \cdot K)$ -ю и т.д. Коля подумал, а нельзя ли по номеру строки в тексте найти номер страницы, на которой будет напечатана эта строка, и порядковый номер этой строки на странице.

Входной файл input.txt содержит число A — количество строк, которое печатается на странице, и число N — номер строки ($1 \leq A \leq 200, 1 \leq N \leq 20000$).

В выходной файл output.txt выведите два числа — номер страницы, на которой будет напечатана эта строка и номер строки на странице

Примеры

Входные данные	Результат
50 1	1 1
15 43	3 13

7. ПОСЛЕДОВАТЕЛЬНОСТЬ ВЛАДИМИРА

Бутин Владимир, ученик 9 а класса Лицея №3 г.Чебоксары очень любит разбираться в последовательности чисел и давать им имена. Однажды просмотрев на последовательность, которую просматривал его друг он назвал ее «Вовкин». Итак, по его мнению последовательность является «Вовкин» последовательностью, если она одинаково читается как слева направо, так и справа налево. Например, следующие последовательности являются «Вовкиными»:

1 2 3 4 5 4 3 2 1

1 2 1 2 2 1 2 1

Вова предложил своему другу последовательность, в которой нужно определить, какое минимальное количество чисел и каких надо приписать в конец этой последовательности, чтобы она стала «Вовкин» последовательностью.

Во входном файле input.txt записано сначала число K — количество элементов исходной последовательности. Далее записано K чисел — элементы этой последовательности. $1 \leq K \leq 100$, элементы последовательности — натуральные числа от 1 до 9.

В выходной файл output.txt выведите сначала число M — минимальное количество элементов, которое надо дописать к последовательности, а потом M чисел (каждое — от 1 до 9) — числа, которые надо дописать к последовательности.

Примеры

Входные данные	Результат
9 1 2 3 4 5 4 3 2 1	0
5 1 2 3 4 5	4 4 3 2 1

8. ПРАВИЛО ПЕТРОВА АЛЕКСАНДРА

В кинотеатре будущего мэра г.Москвы Петрова Александра действует жесткое правило **рядом с каждым мальчиком сидела хотя бы одна девочка, а рядом с каждой девочкой — хотя бы один мальчик.** «А мальчиков и В девочек пошли в кинотеатр и купили билеты на подряд идущие места в одном ряду. Вам нужно написать программу, которая выдаст, как нужно сесть мальчикам и девочкам, чтобы действовало правило Петрова.

Во входном файле input.txt записаны два числа — A и B (оба числа натуральные, не превосходящие 100).

В выходной файл output.txt выведите какую-нибудь строку, в которой будет ровно A символов B (обозначающих мальчиков) и B символов G (обозначающих девочек), удовлетворяющую условию задачи. Пробелы между символами выводить не нужно.

Если рассадить мальчиков и девочек согласно условию задачи невозможно, в выходной файл должна быть записана строка NO SOLUTION.

Примеры

Входные данные	Результат
5 3	BGBGBBGB
100 1	NO SOLUTION

9. Ё МОЁ, СКОЛЬКО ЖЕ СЛОВ В СТРОКЕ?

Во входном файле input.txt записана строка текста, в которой могут встречаться:

- прописные и строчные (т.е. большие и маленькие) латинские буквы,
- пробелы,
- знаки препинания: точка, запятая, восклицательный и вопросительный знак,
- символ `-`, обозначающий в некоторых случаях тире, а в некоторых — дефис.

Слово — это последовательность подряд идущих латинских букв и знаков дефис, ограниченная с обоих концов. В качестве ограничителей могут выступать начало строки, конец строки, пробел, знак препинания, тире. Тире отличается от дефиса тем, что слева и справа от знака дефис пишутся буквы, а хотя бы с одной стороны от тире идет либо начало строки, либо конец строки, либо пробел, либо какой-либо знак препинания, либо еще одно тире.

Напишите программу, определяющую, сколько слов в данной строке текста.

Во входном файле input.txt записана строка длиной не более 200 символов.

В выходной файл output.txt выведите одно число — количество слов, которые содержатся в исходной строке.

Входные данные	Результат
www.liceum3cheb.ru	3
Gyro-compass – this is a ...	4

10. ДЛИНА ОБЩЕЙ ЧАСТИ

Даны N отрезков прямой. Найти длину общей части всех этих отрезков.

Из файла input.txt вводится сначала число N ($1 \leq N \leq 100$). Далее вводится N пар чисел,

задающих координаты левого и правого концов каждого отрезка. Все координаты - числа из диапазона от 0 до 30000. Левый конец отрезка всегда имеет координату строго меньшую, чем правый.

В выходной файл output.txt выведите длину общей части этих отрезков. Если у всех этих отрезков общей части нет, выведите 0.

Входные данные	Результат
3 1 10 3 15 2 6	3
3 1 10 2 20 11 20	0

11. ПРЕОБРАЗОВАНИЕ

Дан некоторый текст (длиной ≤ 255), слова которого состоят из 0 и 1 и разделены между собой произвольным количеством пробелов (количество пробелов ≥ 1), например:

110000111 000 001111 0000101 00001 1 0

Преобразовать текст следующим образом:

Если в слове количество 1 $>$ количества 0, заменить 0 в этом слове на 1, иначе заменить 1 в этом слове на 0.

Между словами оставить ровно по одному пробелу.

Для данного примера

000000000000 111111 0000000 00000 1 0

Входные данные	Результат
110000111 000 001111 0000101 00001 1 0	000000000000 111111 0000000 00000 1 0
000111 00 1110 111100000	000000 00 1111 0000000000

12. ПУЗЫРЬКОВАЯ СОРТИРОВКА

Дан целочисленный массив, размер которого ≤ 1000 , а значения элементов лежат в диапазоне $[-100;100]$.

Необходимо, применив пузырьковую сортировку, определить минимальное число просмотров (проходов) массива до полного его упорядочения по возрастанию значений элементов (считая и последнюю «контрольную» проверку).

Проходом будем считать один просмотр всех необходимых элементов массива с целью «всплытия» одного очередного «пузырька» на свой уровень.

Напомним, что при пузырьковой сортировке сравниваются пары соседних элементов массива $a[i]$ и $a[i+1]$.

Например, для массива:

1 2 3 4 5 6 8 7 9

требуется 2 (два) прохода: 1-й поменяет местами числа 7 и 8, 2 –й убедится в том, что массив после 1 –го прохода стал упорядоченным по возрастанию.

Во входном файле: в первой строке количество элементов, во второй значения элементов массива в одной строке через пробел.

В выходном файле – минимальное число проходов.

Входные данные	Результат
9 1 2 3 4 5 6 8 7 9	2
10 -3 -1 0 3 5 9 12 13 25 38	1

13. ЛОКАЛЬНАЯ СОРТИРОВКА

Дан целочисленный массив, размер которого ≤ 1000 , а значения элементов лежат в диапазоне $[-100;100]$.

Необходимо упорядочить по возрастанию элементы этого массива, лежащие между первым минимальным и последним максимальным, включив в сортировку и эти крайние элементы (т.е. минимальный и максимальный).

Например, для массива:

9 - 3 21 1 21 13 4 15 6 18 -7 9 -7 12

получим следующий результат:

9 -3 21 1 -7 4 6 13 15 18 21 9 -7 12

Во входном файле: в первой строке количество элементов, во второй значения элементов массива в одной строке через пробел.

В выходном файле – отсортированный указанным способом массив.

Входные данные	Результат
14 9 - 3 21 1 21 13 4 15 6 18 -7 9 -7 12	9 -3 21 1 -7 4 6 13 15 18 21 9 -7 12
10 -3 4 -3 0 3 4 5 9 12 13 -5	-3 -3 0 3 4 4 5 9 12 13 -5

14. КОРОТКАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ

Ограничение по времени тестирования: по 5 секунд на один тест.

Дано целое число n , $0 < n < 32768$. Пусть дана последовательность $P_1P_2P_3...P_k...$, где каждая группа цифр P_k состоит из записанных одно за другим чисел от 1 до k . Например, первые 84 цифры последовательности выглядят так: 11212312341234512345612345671234567812345678912345678910123456789101123456789101112.

Написать программу, которая определит, какая цифра находится на n -ой позиции в построенной последовательности.

Входной текстовый файл INPUT.TXT содержит одно число n .

Выходной файл OUTPUT.TXT должен содержать одну цифру, которая стоит на n -ой позиции в последовательности.

Пример файлов входных и выходных данных:

Входные данные	Результат
20	5

15. ЧЕТНОСТЬ БОЛЬШОГО ЧИСЛА

Ограничение по времени тестирования: по 5 секунд на один тест.

Рассмотрим длинное число, состоящее из N цифр ($1 \leq N \leq 32000$). Нужно определить, четное ли оно.

Написать программу, которая определит, четное данное число или нет. Если число четное, то в выходной файл вывести 1, если нечетное – 0.

Входной текстовый файл INPUT.TXT содержит одно число, состоящее из n цифр.

Выходной файл OUTPUT.TXT должен содержать одну цифру, 1 – если число четное и 0 – если число не четное.

Пример файлов входных и выходных данных:

Входные данные	Результат
4	1
1245135356256561256345656712453 1245	0

16. САМОЕ ДЛИННОЕ ЧИСЛО

Ограничение по времени тестирования: по 5 секунд на один тест.

Рассмотрим число, состоящее из N цифр ($1 \leq N \leq 1000$). Мальчик Вова решил получить из данного числа путем перестановок цифр самое большое число. Помогите ему составить программу для нахождения этого числа.

Входной файл input.txt в одной строке содержит искомое число, состоящее из N цифр.

Выходной файл output.txt должен содержать самое большое число.

Пример файлов входных и выходных данных:

Входные данные	Результат
12356	65321
11223344998855	99885544332211

17. КОЛИЧЕСТВО ДЕЛЯЩИХСЯ НА ТРИ

Ограничение по времени тестирования: по 5 секунд на один тест.

Рассмотрим последовательность 1, 12, 123, 1234, 12345, 123456, 1234567, 12345678, 123456789, 12345678910, 1234567891011,

Написать программу, которая определит количество из N первых элементов этой последовательности, делящихся на три.

Входной текстовый файл INPUT.TXT содержит одно число N – количество элементов последовательности ($1 \leq N \leq 2^{31} - 1$).

Выходной текстовый файл OUTPUT.TXT должен содержать одно найденное число.

Пример файлов входных и выходных данных:

Входные данные	Результат
4	2

18. СУММА ПРОТИВОПОЛОЖНЫХ ЧИСЕЛ

Ограничение по времени тестирования: по 5 секунд на один тест.

Пусть даны два любых целых числа A и B ($1 \leq A, B \leq 2000000000$). Нужно найти противоположное значение суммы их противоположных значений.

Входной файл input.txt в одной строке содержит два числа A и B.

В выходной файл output.txt в одной строке вывести противоположное значение суммы противоположных чисел.

Пример файлов входных и выходных данных:

Входные данные	Результат
4 5	9
-4 5	1

19. ПОДАРКИ

Ограничение по времени тестирования: по 5 секунд на один тест.

Приближался Новый год, и отец Вовы купил своим детям по подарку. Оказалось, что в них разное количество конфет. Тогда отец купил еще конфет и стал их раскладывать по подаркам следующим образом: брал подарок с наименьшим количеством конфет и добавлял в него одну конфету. Требуется написать программу, которая найдет наименьшее количество конфет, оказавшихся в одном из подарков после завершения их раскладывания.

Технические требования:

Входной текстовый файл INPUT.TXT содержит в первой строке N – количество детей и M – количество купленных конфет. Числа записаны через пробел, $1 \leq N \leq 10000$, $1 \leq M \leq 1000000$. Далее в N строках записаны числа в диапазоне от 1 до 30000 – количество конфет в подарках.

Выходной файл OUTPUT.TXT должен содержать одно найденное число.

Пример файлов входных и выходных данных:

Входные данные	Результат
2 4 1 2	3

20. УРАВНЕНИЕ $X\#A=B$

Ограничение по времени тестирования: по 5 секунд на один тест.

Операция $\#$ действует с целыми положительными числами побитового, т.е. надо представить числа в двоичном виде и для каждой пары бит с одинаковыми номерами выполнить действие по следующим правилам: $0\#0=0$, $1\#0=1$, $0\#1=1$, $1\#1=0$.

Написать программу, которая по заданным a и b решит уравнение $x\#a=b$.

Входной текстовый файл INPUT.TXT содержит два целых положительных числа, записанных через пробел. Числа не превышают значения 32767

Выходной файл OUTPUT.TXT должен содержать одно найденное число.

Пример файлов входных и выходных данных:

Входные данные	Результат
1 1	0

21. ПОСЛЕДОВАТЕЛЬНОСТЬ КЕАНЕ

Ограничение по времени тестирования: по 5 секунд на один тест.

Бесконечная последовательность битов, предложенная Кеане, равна $001001110001001110110110001\dots$ и формируется следующим алгоритмом: вначале записывается 0, потом 001, далее 001001110, то есть для получения следующего члена предыдущий записывается дважды, а справа приписывается его отрицание. Элементы этого ряда являются начальными подпоследовательностями Кеане.

Требуется написать программу, которая по заданному n определит n -й бит этой последовательности.

Входной текстовый файл INPUT.TXT содержит число n ($n \leq 10^{200}$).

Выходной файл OUTPUT.TXT должен содержать найденный бит.

Пример файлов входных и выходных данных:

Входные данные	Результат
18	0

22. УРАВНЕНИЕ ПРЯМОЙ ВИДА $Y=KX+B$

Ограничение по времени тестирования: по 5 секунд на один тест.

На плоскости даны две точки с соответствующими координатами. Через них проходит прямая, уравнение которой описывается при помощи уравнения $y=kx+b$.

Написать программу нахождения коэффициента k , учитывая, что точки одновременно не могут лежать на прямой, параллельной осям x или y .

Входной файл input.txt в одной строке содержит 4 любых числа x_1, y_1, x_2, y_2 , координаты соответствующих точек данной прямой ($-100 \leq x_1, y_1, x_2, y_2 \leq 100$).

Выходной файл output.txt одно число k с точностью до 4 знаков в дробной части.

Пример файлов входных и выходных данных:

Входные данные	Результат
1 1 2 2	1.000

23. ИГРА В МОНЕТЫ

Ограничение по времени тестирования: 5 секунд на каждый тест

Гриша и Дима играют в следующую игру: они разложили однокопеечные монетки в стопки (в разных стопках может быть различное количество монет), а стопки расположили на столе перед собой в ряд слева направо. Затем Гриша и Дима по очереди делают ходы. На каждом ходе один из игроков берет слева несколько стопок, не меньше одной, но и не больше, чем перед этим взял его соперник. Первый игрок своим первым ходом берет не более K стопок. Игра заканчивается, когда стопок не остается.

Написать программу, позволяющую вычислить, какое максимальное число монет может получить первый участник после окончания игры, если второй — тоже старается ходить так, чтобы получить как можно больше монет.

Входной файл INPUT.TXT состоит из одной строки, в которой записаны: число стопок N ($1 \leq N \leq 180$), за ним идут N чисел, задающих количество монет в стопках слева направо (количество монет в стопке — не менее 1 и не более 20000), а затем число K , ограничивающее количество стопок, которые первый игрок может взять на первом ходе ($1 \leq K \leq 80$). Все числа в строке разделены пробелом.

В выходной файл OUTPUT.TXT необходимо вывести одно число — максимальное количество монет, которое заведомо может получить первый игрок, как бы ни играл второй.

Пример файлов входных и выходных данных:

Входные данные	Результат
3 4 9 1 3	14
4 1 2 2 7 3	5
5 3 4 8 1 7 2	18

24. ХАНОЙСКАЯ БАШНЯ

Есть три стержня. На первом из них расположено N колец (1-е, верхнее, самое маленькое, N -ое, нижнее, самое большое). За один ход разрешается с любого стержня снять верхнее кольцо и надеть его на любой другой стержень. При этом запрещается класть большее кольцо на меньшее. Требуется, чтобы все кольца оказались на стержне номер 2.

Входные данные

Во входном файле записано одно число N ($1 \leq N \leq 10$)

Выходные данные

В выходной файл выведите последовательность команд.

Каждая команда задается двумя числами - с какого стержня снимаем кольцо, и на какой надеваем

Входные данные	Результат
2	1 3 1 2 3 2

25. ТРОЕЧНЫЕ ПОСЛЕДОВАТЕЛЬНОСТИ

Вводится число N . Сгенерируйте в анти-лексикографическом порядке все последовательности длины N , состоящие из чисел 3, 4, 5, в которых количество троек не превосходит двух.

В «анти-лексикографическом» обозначает «в порядке, обратном к лексикографическому».

В «лексикографическом порядке» обозначает, что если на первых местах две последовательности совпадают, а на месте $X+1$ - различаются, то раньше должна идти та из них, в которой число на месте $X+1$ меньше.

В анти-лексикографическом, соответственно, наоборот. $1 \leq N \leq 10$.

Входные данные	Результат
3	5 5 5 5 5 4 5 5 3 5 4 5 5 4 4 5 4 3 5 3 5 5 3 4 5 3 3 4 5 5 4 5 4 4 5 3 4 4 5 4 4 4 4 4 3 4 3 5 4 3 4 4 3 3 3 5 5 3 5 4 3 5 3 3 4 5 3 4 4 3 4 3

	3 3 5
	3 3 4

26. ЦИКЛ

Дан граф. Определить, есть ли в нем цикл отрицательного веса, и если да, то вывести его.

Входные данные. Во входном файле в первой строке число N ($1 \leq N \leq 100$) - количество вершин графа. В следующих N строках находится по N чисел - матрица смежности графа. Все веса ребер не превышают по модулю 10000. Если ребра нет, то соответствующее число равно 100000.

Выходные данные. В первой строке выходного файла выведите "YES", если цикл существует или "NO" в противном случае. При его наличии выведите во второй строке количество вершин в искомом цикле (считая одинаковые первую и последнюю) и в третьей строке - вершины, входящие в этот цикл в порядке обхода.

Входные данные	Результат
2	YES
0 -1	3
-1 0	1 2 1

27. ЛАБИРИНТ ЗНАНИЙ

В ЛКШ построили аттракцион "Лабиринт знаний". Лабиринт представляет собой N комнат, занумерованных от 1 до N , между некоторыми из которых есть двери. Когда человек проходит через дверь, показатель его знаний изменяется на определенную величину, фиксированную для данной двери. Вход в лабиринт находится в комнате 1, выход - в комнате N . Каждый ЛКШонок проходит лабиринт ровно один раз и попадает в группу в зависимости от набранных знаний (при входе в лабиринт этот показатель равен нулю). Ваша задача показать наилучший результат.

Входные данные. Первая строка входного файла содержит целые числа N ($1 \leq N \leq 2000$) - количество комнат и M ($1 \leq M \leq 10000$) - количество дверей. В каждой из следующих M строк содержится описание двери - номера комнат, из которой она ведет и в которую она ведет, а также целое число, которое прибавляется к количеству знаний при прохождении через дверь (это число по модулю не превышает 10000). Двери могут вести из комнаты в нее саму, между двумя комнатами может быть более одной двери. Выходные данные. В выходной файл выведите ":" - если можно получить неограниченно большой запас знаний, "(" - если лабиринт пройти нельзя, и максимальное количество набранных знаний в противном случае.

Входные данные	Результат
2 2	2 2
1 2 5	1 2 5
1 2 -5	1 2 -5

28. УРАВНЕНИЕ С ПРОПУЩЕННЫМИ ЦИФРАМИ

Задано уравнение вида $A + B = C$, где A , B и C - неотрицательные целые числа, в десятичной записи которых некоторые цифры заменены знаками вопроса (?). Примером такого уравнения является $?2+34=4?$. Требуется так подставить вместо знаков вопроса цифры, чтобы это равенство стало верным, либо определить, что это невозможно. Найти только одно из возможных решений.

Входные данные. Заданное уравнение содержится в первой строке входного файла. Длина уравнения не превышает 80 символов. Входной файл не содержит пробелов.

Выходные данные. В выходной файл требуется вывести верное равенство, полученное из исходного уравнения заменой знаков вопроса цифрами, либо сообщение "NO SOLUTION".

Входные данные	Результат
??2?4+9?=355	00264+91=355

29. ПОКУПКА БИЛЕТОВ. ?

За билетами на премьеру нового мюзикла выстроилась очередь из N человек, каждый из которых хочет купить 1 билет. На всю очередь работала только одна касса, поэтому продажа билетов шла очень медленно, приводя "постояльцев" очереди в отчаяние. Самые сообразительные быстро заметили, что, как правило, несколько билетов в одни руки кассир продаёт быстрее, чем когда эти же билеты продаются по одному. Поэтому они предложили нескольким подряд стоящим людям отдавать деньги первому из них, чтобы он купил билеты на всех.

Однако для борьбы со спекулянтами кассир продавала не более 3-х билетов в одни руки, поэтому договориться таким образом между собой могли лишь 2 или 3 подряд стоящих человека.

Известно, что на продажу i -му человеку из очереди одного билета кассир тратит A_i секунд, на продажу двух билетов - B_i секунд, трех билетов - C_i секунд. Напишите программу, которая подсчитает минимальное время, за которое могли быть обслужены все покупатели.

Обратите внимание, что билеты на группу объединившихся людей всегда покупает первый из них. Также никто в целях ускорения не покупает лишних билетов (то есть билетов, которые никому не нужны).

Формат входных данных

Во входном файле записано сначала число N - количество покупателей в очереди ($1 \leq N \leq 5000$). Далее идет N троек натуральных чисел A_i, B_i, C_i . Каждое из этих чисел не превышает 3600. Люди в очереди нумеруются начиная от кассы.

Формат выходных данных

В выходной файл выведите одно число - минимальное время в секундах, за которое могли быть обслужены все покупатели.

Входные данные	Результат
5 5 10 15 2 10 15 5 5 5 20 20 1 20 1 1	12
2 3 4 5 1 1 1	4

30. ВОССТАНОВЛЕНИЕ СКОБОК

Задан шаблон, состоящий из круглых скобок и знаков вопроса. Требуется определить, сколькими способами можно заменить знаки вопроса круглыми скобками так, чтобы получилось правильное скобочное выражение.

Входные данные. Первая строка входного файла содержит заданный шаблон длиной не более 80 символов.

Выходные данные. Выведите в выходной файл искомое количество способов. Исходные данные будут таковы, что это количество не превзойдет $2 \cdot 10^9$.

Входные данные	Результат
????(?)	2