

Олимпиадные задачи. PASCAL

Язык программирования: PASCAL

Сергей Митрофанов

E-mail: infostar@mail.ru

<http://www.Best-Listing.ru/>

В сборнике 89 задач

Гимназия "Лаборатория Салахова"

Сургут

25 декабря 2015 года

Содержание

| | | |
|----|---|----|
| 1 | Стандартные файлы Input и Output [io_3] | 3 |
| 2 | Условный оператор [if_7] | 4 |
| 3 | Циклы [ts_16] | 9 |
| 4 | Одномерные массивы [om_12] | 17 |
| 5 | Целочисленная арифметика [tsa_3] | 24 |
| 6 | Двумерные массивы [dm_8] | 26 |
| 7 | Обработка строк [str_24] | 31 |
| 8 | Сортировки [s_4] | 46 |
| 9 | Записи [z_5] | 49 |
| 10 | Динамическая память [din_4] | 53 |
| 11 | Комбинаторика [cmb_1] | 57 |
| 12 | for MSP [msp_2] | 58 |

1 Стандартные файлы Input и Output [io_3]

io_1. Требуется сложить два целых числа A и B .

Время: 1 с. Память: 16 Мб Сложность: 2%

Входные данные:

В единственной строке `input.txt` записано 2 числа через пробел, не превышающих 10^9 .

Выходные данные:

В единственную строку выходного файла `output.txt` нужно вывести одно целое число – сумму чисел A и B .

Пример:

| | |
|-----------|------------|
| input.txt | output.txt |
| 2 3 | 5 |

io_2. Из текстового файла прочитать все цифры и найти их сумму.

io_3. Заданы первый и второй элементы арифметической прогрессии. Требуется написать программу, которая вычислит элемент прогрессии по его номеру.

Входной файл `input.txt` содержит три целых числа, разделенных пробелами – первый элемент прогрессии A_1 ($1 \leq A_1 \leq 1000$), второй элемент прогрессии A_2 ($1 \leq A_2 \leq 1000$), и номер требуемого элемента N ($1 \leq N \leq 1000$).

Выходной файл `output.txt` должен содержать одно целое число – N -й элемент арифметической прогрессии.

2 Условный оператор [if_7]

if_1.

В прошлом году Вася пошел в школу и научился считать. В этом году он изучил таблицу умножения и теперь умеет перемножать любые числа от 1 до 10 без ошибок. Друг Петя рассказал ему про системы счисления, отличные от десятичной. В частности, про двоичную, восьмеричную и даже шестнадцатеричную. Теперь Вася без труда (но уже с помощью листка и ручки) может перемножать числа от 1 до 10 и в этих системах, используя перевод из нестандартной системы в десятичную и обратно из десятичной. Например, если Васе нужно перемножить числа 101 и 1001 в двоичной системе, то он сначала эти числа переводит в десятичное представление следующим образом:

$$101_2 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 4 + 0 + 1 = 5$$

$$1001_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 8 + 0 + 0 + 1 = 9$$

После чего перемножение чисел 5 и 9 Вася с легкостью производит в десятичной системе счисления в уме и получает число 45. Далее производится перевод из десятичной системы счисления в двоичную. Для этого Вася делит число 45 на 2 (порядок системы счисления), запоминая остатки от деления, до тех пор пока в результате не останется число 0: ответ составляется из полученных остатков от деления путем их записи в обратном порядке. Таким образом Вася получает результат: $101_2 \cdot 1001_2 = 101101_2$. Но теперь Вася изучает таблицу умножения чисел от 1 до 100 в десятичной системе счисления, а поскольку запомнить такую таблицу очень сложно, то Васе придется очень долго ее зубрить. Составьте для Васи программу, которая поможет ему проверять свои знания.

Входные данные Во входном файле INPUT.TXT записаны три натуральных числа A , B и C через пробел. Числа A и $B \leq 10^2$, а $C \leq 10^6$.

Выходные данные В выходной файл нужно вывести YES в том случае, если $A \cdot B = C$ и вывести NO в противном случае.

if_2.

Пятью пять – двадцать пять!

(Время: 1 сек. Память: 16 Мб. Сложность: 8 %)

Вася и Петя учатся в школе в одном классе. Недавно Петя поведал Васе о хитром способе возведения в квадрат натуральных чисел, оканчивающихся на цифру 5. Теперь Вася может с легкостью

возводить в квадрат двузначные (и даже некоторые трехзначные) числа, оканчивающиеся на 5. Способ заключается в следующем: для возведения в квадрат числа, оканчивающегося на 5 достаточно умножить число, полученное из исходного вычеркиванием последней пятерки на следующее по порядку число, затем остается лишь приписать "25" к получившемуся результату справа. Например, для того, чтобы возвести число 125 в квадрат достаточно 12 умножить на 13 и приписать 25, т.е. приписывая к числу $12 \cdot 13 = 156$ число 25, получаем результат 15625, т.е. $125^2 = 15625$. Напишите программу, возводящую число, оканчивающееся на 5, в квадрат для того, чтобы Вася смог проверить свои навыки.

Входные данные

В единственной строке входного файла INPUT.TXT записано одно натуральное число A , оканчивающееся на цифру 5, не превышающее $4 \cdot 10^5$.

Выходные данные

В выходной файл OUTPUT.TXT выведите одно натуральное число – A^2 без лидирующих нулей.

Примеры

| N | INPUT.TXT | OUTPUT.TXT |
|---|-----------|------------|
| 1 | 5 | 25 |
| 2 | 75 | 5625 |
| 3 | 4255 | 18105025 |

if_2v2. Вася и Петя учатся в школе в одном классе. Недавно Петя поведал Васе о хитром способе возведения в квадрат натуральных чисел, оканчивающихся на цифру 5. Теперь Вася может с легкостью возводить в квадрат двузначные (и даже некоторые трехзначные) числа, оканчивающиеся на 5.

Способ заключается в следующем: для возведения в квадрат числа, оканчивающегося на 5 достаточно умножить число, полученное из исходного путем вычеркивания последней пятерки на следующее за ним по порядку число, затем остается лишь приписать "25" к получившемуся результату справа.

Время: 1 с. Память: 16 Мб Сложность: 8%

Входные данные:

В единственной строке `input.txt` записано единственное целое число N , не превышающее по абсолютной величине $4 \cdot 10^5$.

Выходные данные:

В единственную строку выходного файла `output.txt` нужно вывести одно натуральное число – A^2 без лидирующих нулей

Пример:

| input.txt | output.txt |
|-----------|------------|
| 5 | 25 |
| 75 | 5625 |
| 4255 | 18105025 |

if_3.

Прямоугольник состоит из $X \times Y$ квадратных клеток одинакового размера. Из него вырезан прямоугольник размером

$$(X - 2) \times (Y - 2)$$

так, что осталась рамка шириной в одну клетку. Определить, можно ли покрыть всю рамку плитками размером $A \times 1$. Запас плиток неограничен, плитки не накладываются одна на другую и за пределы рамки не выходят.

Входной текстовый файл `Input.txt` содержит в первой строке натуральное число K – количество тестов ($1 \leq K \leq 10$). В следующих K строках записаны по три натуральных числа: X, Y – размеры рамки, A – размер плитки

$$(3 \leq X, Y \leq 2 \cdot 10^9, 1 \leq A \leq 2 \cdot 10^9).$$

Числа разделены пробелами.

Выходной текстовый файл `Output.txt` должен содержать одну строку из K символов 0 или 1 (1 – если покрытие существует, 0 – иначе).

Пример

| Input.txt | Output.txt |
|---------------------|------------|
| 1 3 3 1 | 1 |
| 2 3 3 2 3 3 3 | 10 |

if_4.

Заданы две клетки шахматной доски. Требуется определить, возможно ли попасть из одной клетки в другую одним ходом шахматного коня, а если нет, то следует выяснить, возможно ли попасть с помощью двух ходов.

Входной файл *input.txt* содержит координаты двух клеток в общепринятом формате: каждая координата записывается как латинская буква и цифра, координаты отделены друг от друга запятой и пробелом.

Выходной файл *output.txt* должен содержать цифру "1", если возможно из одной клетки в другую попасть за 1 ход, либо "2", если попасть можно за 2 хода, либо "NO", если одна клетка недостижима из другой ни за 1, ни за 2 хода.

| Input.txt | Output.txt |
|-----------|------------|
| a1, h8 | NO |
| a1, b3 | 1 |
| a1, d4 | 2 |

if_5.

Два отрезка на плоскости заданы целочисленными координатами своих концов в декартовой системе координат. Требуется определить, существует ли у них общая точка.

Входной файл *input.txt* содержит координаты четырех точек, задающих отрезки. В первой строке содержатся координаты первого конца первого отрезка, во второй – второго конца первого отрезка, в третьей и четвертой – координаты концов второго отрезка. Все координаты – целые числа, не превосходящие 10000 по абсолютной величине.

В выходной файл *output.txt* следует вывести слово "Yes", если общая точка есть, или слово "No" – в противном случае.

Пример

| Input.txt | Output.txt |
|--------------------------|------------|
| 0 0 1 1 0 1 1 0 | Yes |

if_6.

Мише поручили решить следующую задачу: имеется дата начала и конца некоторого мероприятия; требуется определить его длительность. Он написал программу и попросил Машу ее проверить. Через некоторое время пришла Маша и расстроила Мишу: "Твоя программа работает неправильно. По-моему, ты забыл, что года бывают високосными". У Миши очень мало свободного времени и он не успевает исправить свою программу. Помогите ему. Год является високосным, тогда и только тогда, когда, выполнено одно из следующих условий:

год делится на 4, но не делится на 100; год делится на 400.

Например, года 400, 404, 496, 504, 2000, 2004 являются високосными, а года 100, 200, 300, 503, 1000, 2001, 2005 – нет.

Первая строка входного файла *input.txt* содержит дату начала мероприятия. Вторая строка входного файла содержит дату конца мероприятия. Гарантируется, что первая дата меньше второй. Даты заданы в формате DD.MM.YYYY.

В выходной файл *output.txt* выведите единственное число – длительность мероприятия (в днях).

Примеры

| Input.txt | Output.txt |
|------------|------------|
| 01.02.2004 | 30 |
| 01.03.2004 | |
| 01.09.0005 | 366 |
| 01.09.0006 | |
| 01.09.2005 | 36 |
| 06.10.2005 | |

if_7.

Переполнение

Производится арифметическая операция сложения двух заданных 32-разрядных целых чисел со знаком X и Y . Необходимо вычислить результат выполнения указанной операции $R = X + Y$ и определить, будет ли происходить переполнение при его вычислении. Переполнением считается ситуация, при которой в результате выполнения операции будет получаться число больше $2^{31} - 1$ или меньше -2^{31} . В качестве результата при отсутствии переполнения выдать результат выполнения указанной операции R , при возникновении переполнения выдать текстовую строку "OVERFLOW".

Во входном файле *input.txt* находятся целые числа X ($-2^{31} \leq X \leq 2^{31} - 1$) и Y ($-2^{31} \leq Y \leq 2^{31}$).

В выходной файл *output.txt* вывести целую величину R при отсут-

ствии переполнения или текстовую строку **OVERFLOW** в кодировке 1251 при наличии переполнения.

Пример

| Input.txt | Output.txt |
|--------------|------------|
| 2147483646 1 | 2147483647 |
| 1 2147483647 | OVERFLOW |

3 Циклы [ts_16]

ts_1. Требуется посчитать сумму целых чисел от 1 до N .

Время: 1 с. Память: 16 Мб Сложность: 19%

Входные данные:

В единственной строке *input.txt* записано единственное целое число N , не превышающее по абсолютной величине 10^4 .

Выходные данные:

В единственную строку выходного файла *output.txt* нужно вывести одно целое число – сумму чисел от 1 до N .

Пример:

| input.txt | output.txt |
|-----------|------------|
| 5 | 15 |

ts_2. Для делимости числа на 11 необходимо, чтобы разность между суммой цифр, стоящих на четных местах, и суммой цифр, стоящих на нечетных местах, делилась на 11. Требуется написать программу, которая проверит делимость заданного числа на 11.

Входной файл *input.txt* содержит одно натуральное число N , делимость которого надо проверить ($1 \leq N \leq 10^{10000}$).

В выходной файл *output.txt* выведите "YES", если число делится на 11, или "NO" иначе.

Пример

| Input.txt | Output.txt |
|-----------|------------|
| 121 | YES |
| 1211 | NO |

ts_3. Требуется посчитать сумму целых чисел от 1 до N .

Время: 1 с. Память: 16 Мб Сложность: 19%

Входные данные:

В единственной строке *input.txt* записано единственное целое число N , не превышающее по абсолютной величине 10^4

Выходные данные:

В единственную строку выходного файла `output.txt` нужно вывести одно целое число – сумму чисел от 1 до N .

Пример:

| input.txt | output.txt |
|-----------|------------|
| 5 | 15 |

ts_4.

Домашнее задание

(Время: 1 сек. Память: 16 Мб. Сложность: 25%)

Петя успевает по математике лучше всех в классе, поэтому учитель задал ему сложное домашнее задание, в котором нужно в заданном наборе целых чисел найти сумму всех положительных элементов, затем найти где в заданной последовательности находятся максимальный и минимальный элемент и вычислить произведение чисел, расположенных между ними. Так же известно, что минимальный и максимальный элемент встречаются в заданном множестве чисел только один раз. Поскольку задач такого рода учитель дал Пете около ста, то Петя как сильный программист смог написать программу, которая по заданному набору чисел самостоятельно находит решение. А вам слабо?

Входные данные

В первой строке входного файла `input.txt` записано единственное число N – количество элементов массива. Вторая строка содержит N целых чисел, представляющих заданный массив. Все элементы массива разделены пробелом. Каждое из чисел во входном файле не превышает 10^2 по абсолютной величине.

Выходные данные

В первую строку выходного файла `output.txt` нужно вывести два числа: сумму всех положительных чисел и произведение чисел, расположенных между минимальным и максимальным элементами. Значения суммы и произведения не превышают по модулю $3 \cdot 10^4$.

Примеры

| N | INPUT.TXT | OUTPUT.TXT |
|---|-----------------------------|------------|
| 1 | 5 -7 5 -1 3 9 | 17 -15 |
| 2 | 8 3 14 -9 4 -5 1 -12 4 | 26 180 |
| 3 | 10 -5 1 2 3 4 5 6 7 8 -3 | 36 5040 |

ts_5.

Для оптимизации передачи по цифровому каналу связи небольших объемов медленно меняющихся данных, представленных в виде массива байтов фиксированной длиной 10 элементов со значениями от 0 до 99, используется его сжатие (архивация) по следующему правилу. Массив исходных, медленно меняющихся данных, можно представить в следующем виде:

(Число 1)(Число 1)(Число 1)(Число 2)(Число 2)(Число 2)
(Число 2)(Число 3)(Число 2)(Число 2)

Массив сжатых данных формируется за счет удаления последовательностей повторяющихся элементов, идущих подряд, и добавления количества элементов в таких последовательностях. Если значение предыдущего элемента отлично от значения последующего элемента, то считается, что такая последовательность состоит из единственного элемента. Следовательно, в приведенном примере число 1 повторялось 3 раза, число 2 – 4 раза и 2 раза (в правой части массива), а число 3 – 1 раз. Поэтому массив сжатых данных для приведенного примера массива исходных данных может быть представлен в следующем виде:

(Число 1)(3)(Число 2)(4)(Число 3)(1)(Число 2)(2)

Требуется произвести восстановление массива исходных данных на основании заданного массива сжатых данных.

Формат входного файла

Во входном файле *input.txt* находится массив сжатых данных ($0 \leq Y[i] \leq 255$). Элементы массива разделены пробелами. Длина массива всегда четная. Количество элементов массива от 2 до 20.

Формат выходного файла

В выходной файл *output.txt* вывести массив исходных данных. Элементы массива разделить пробелами.

ts_6.

Напечатать введенное число прописью.

ts_7.

Лесенкой называется набор кубиков, в котором каждый более верхний слой содержит кубиков меньше, чем предыдущий. Ниже схематично, с помощью символа (*) – 1 кубик, дан пример возможной лесенки:

```
*
****
*****
*****
```

Требуется написать программу, вычисляющую число лесенок, которое можно построить из N кубиков.

Во входном файле *input.txt* записано натуральное число N ($1 \leq N \leq 100$) – количество кубиков в лесенке.

В выходной файл *output.txt* необходимо вывести число лесенок, которые можно построить из N кубиков.

| Input.txt | Output.txt |
|-----------|------------|
| 3 | 2 |
| 6 | 4 |

ts_8.

В соревнованиях по прыжкам с шестом было замечено одно интересное явление: на очередном этапе соревнований успешные и неуспешные попытки прыжков чередовались: успешный, неуспешный, успешный, неуспешный и так далее (первый был успешным). Спортсменам разрешалась только одна попытка. Тот, кто преодолевал планку, переходил в следующий тур (этап), а тот, кто делал неудачную попытку – выбывал из соревнований. Таким образом, первым выбывал всегда спортсмен с номером 2, а последним – победитель с номером 1.

Требуется написать программу, которая по количеству участников и номеру спортсмена вычислит, каким по счету данный спортсмен выбыл из соревнований. В единственной строке входного файла *input.txt* содержатся два натуральных числа: общее число спортсменов N и порядковый номер спортсмена в стартовом списке M . Числа разделены пробелом ($1 \leq M, N \leq 10^9$).

В единственную строку выходного файла *output.txt* нужно вывести каким по счету спортсмен M выбыл из соревнований. Если это победитель состязания, то выводится число N .

Пример

| Input.txt | Output.txt |
|-----------|------------|
| 4 2 | 1 |
| 4 1 | 4 |
| 9 5 | 7 |

ts_9. Найдите количество чисел из отрезка $[l, r]$, которые делятся на произведение своих цифр.

Входной файл *input.txt* содержит два целых числа l и r ($1 \leq l \leq r \leq 10^9, |r - l| \leq 10^5$).

В выходной файл OUTPUT.TXT выведите ответ на задачу.

Пример

| Input.txt | Output.txt |
|-----------|------------|
| 1 12 | 11 |

ts_10. На городском празднике танцуют девушки в красных и синих юбках. Они двигаются цепочкой и выполняют сложный рисунок танца. Из цепочки девушки выделяются по одной. Первая становится на левом краю сцены, вторая уходит в конец исходной цепочки, третья – на левый край сцены (справа от первой), четвертая – в конец исходной цепочки и т. д., пока все девушки не выстроятся на краю сцены.

Во входном файле Input.txt записано натуральное число N – количество танцующих девушек ($N \leq 1000$).

В выходной файл Output.txt выведите строку, содержащую цепочку из N символов, состоящую из заглавных букв B и R, соответствующих цветам юбок – синему и красному.

ПРИМЕР

| Input.txt | Output.txt |
|-----------|------------|
| 2 | BR |
| 3 | BBR |
| 4 | BBRR |

ts_11. Олимпиадная задача

(Время: 1 сек. Память: 16 Мб. Сложность: 25 %)

Петя успевает по математике лучше всех в классе, поэтому учитель задал ему сложное домашнее задание, в котором нужно в заданном наборе целых чисел найти сумму всех положительных элементов, затем найти где в заданной последовательности нахо-

дятся максимальный и минимальный элемент и вычислить произведение чисел, расположенных между ними. Так же известно, что минимальный и максимальный элемент встречаются в заданном множестве чисел только один раз. Поскольку задач такого рода учитель дал Пете около ста, то Петя как сильный программист смог написать программу, которая по заданному набору чисел самостоятельно находит решение. А вам слабо?

Входные данные

В первой строке входного файла INPUT.TXT записано единственное число N - количество элементов массива. Вторая строка содержит N целых чисел, представляющих заданный массив. Все элементы массива разделены пробелом. Каждое из чисел во входном файле не превышает 10^2 по абсолютной величине.

Выходные данные

В первую строку выходного файла OUTPUT.TXT нужно вывести два числа: сумму всех положительных чисел и произведение чисел, расположенных между минимальным и максимальным элементами. Значения суммы и произведения не превышают по модулю $3 \cdot 10^4$.

Примеры

| N | INPUT.TXT | OUTPUT.TXT |
|---|-----------------------------|------------|
| 1 | 5 -7 5 -1 3 9 | 17 -15 |
| 2 | 8 3 14 -9 4 -5 1 -12 4 | 26 180 |
| 3 | 10 -5 1 2 3 4 5 6 7 8 -3 | 36 5040 |

ts_12. Постулат Бертрانا гласит, что для любого $n > 1$ найдется простое число p в интервале $n < p < 2n$. Такая гипотеза была выдвинута в 1845 году французским математиком Джозефом Бертраном (проверившим его для $n = 3000000$) и доказана в 1850 году Пафнутием Чебышевым.

Ваша задача состоит в том, чтобы решить несколько более общую задачу – а именно по числу n найти количество простых чисел p

из интервала $n < p < 2n$

Напомним, что число называется простым, если оно делится только на само себя и на единицу

Входные данные.

В единственной строке входного файла INPUT.TXT записано одно целое число n ($2 \leq n \leq 50000$).

Выходные данные.

В выходной файл OUTPUT.TXT выведите одно число – ответ на задачу.

ts_13.

Компания <Маша и медведи> является самым крупным интернет-провайдером во всем лесу. Именно поэтому, с просьбой подключить их к Интернету обратились N поросят. Домики поросят расположены в различных точках (x_i, y_i) . Ближайшая точка подключения расположена в точке (x_{net}, y_{net}) . Для того чтобы подключиться к сети всем N поросятам необходимо:

1. провести провод от точки подключения до домика одного из поросят;
2. от подключенного поросенка провести провода ко всем остальным.

При этом провода могут при необходимости пересекаться. Поросята платят деньги в зависимости от длины провода. Количество денег у них ограничено и составляет P тугриков. Они хотят определить: хватит ли им денег на подключение? Так же известно, что единица длины провода стоит тугриков.

В первой строке входного файла *input.txt* находится числа N , и P – целые числа со следующими ограничениями:

$$\begin{aligned} 1 &\leq N \leq 10^3, \\ 1 &\leq C \leq 10^4, \\ 1 &\leq P \leq 10^{15}. \end{aligned}$$

В следующих N строках находятся координаты домов поросят (x_i, y_i) . В последней строке записаны координаты точки соединения (x_{net}, y_{net}) . Все координаты целые и не превосходят 1000 по модулю.

В выходной файл *output.txt* следует вывести "YES", если у поросят

достаточно денег для подключения и "NO" – в противном случае.

ts_14. Сообщество роботов живет по следующим законам:

- один раз в начале года они объединяются в группы по три или пять роботов;
- за год группа из трех роботов собирает 5 новых, а группа из 5 роботов – 9 новых;
- роботы объединяются так, чтобы собрать за год наибольшее количество новых роботов;
- каждый робот живет ровно три года после сборки.
- В начале первого года было K роботов и все они были только что собраны.

Входной файл Input.txt содержит записанные через пробел числа K ($1 \leq K \leq 500$) и N ($1 \leq N \leq 100$).

Выходной файл Output.txt должен содержать одно число – количество роботов в начале N -го года. Количество роботов меньше, чем 2^{31} .

ПРИМЕР

| Input.txt | Output.txt |
|-----------|------------|
| 8 2 | 22 |

ts_15. В написанном выражении $((((1?2)?3)?4)?5)?6$ вместо каждого знака ? вставить знак одного из четырех арифметических действий: +, −, *, / так, чтобы результат вычислений равнялся 35 (при делении дробная часть отбрасывается). Достаточно найти одно решение.

ts_15v2. В выражении $((((1 ? 2) ? 3) ? 4) ? 5) ? 6$ вместо каждого знака ? поставить знак одной из операций +, −, *, / так, чтобы результат вычислений был равен 35.

ts_16. Разбиение числа на слагаемые

Дано натуральное число N . Его можно записать в виде суммы натуральных слагаемых:

$$N = a_1 + a_2 + \dots + a_k,$$

где k, a_1, a_2, \dots, a_k больше нуля.

Будем считать суммы эквивалентными, если они отличаются только порядком слагаемых. Множество эквивалентных сумм при-

веденного вида однозначно представляется последовательностями b_1, b_2, \dots, b_k , упорядоченными по невозрастанию.

Каждую такую последовательность b_1, b_2, \dots, b_k назовем разбиением числа N на k слагаемых.

Найти все разбиения числа N .

4 Одномерные массивы [om_12]

om_1. Имеется пять целых чисел. Среди них:

- если одинаковы 5, то вывести "Impossible иначе
- если одинаковы 4, то вывести "Four of a Kind иначе
- если одинаковы 3 и 2, вывести "Full House иначе
- если есть 5 последовательных, вывести "Straight иначе
- если одинаковы 3, то вывести "Three of a Kind иначе
- если одинаковы 2 и 2, то вывести "Two Pairs иначе
- если одинаковы 2, вывести "One Pair иначе
- вывести "Nothing".

Примечание. 5 чисел берутся из интервала от 1 до 13, разделенных пробелом.

om_2. На столе лежат коробка размера $A1 \times B1 \times C1$ и коробка размера $A2 \times B2 \times C2$. Выясните можно ли одну из этих коробок положить в другую, если разрешены повороты коробок вокруг любого ребра на угол 90 градусов.

Первая строка входного файла содержит три целых числа $A1$, $B1$ и $C1$. Вторая строка входного файла содержит три целых числа $A2$, $B2$ и $C2$. Все числа положительны и не превосходят 1000. Если коробки одинаковы, выведите "Boxes are equal". Если первая коробка может быть положена во вторую, выведите "The first box is smaller than the second one". Если вторая коробка может быть положена в первую, выведите "The first box is larger than the second one". Иначе, выведите "Boxes are incomparable".

| Input.txt | Output.txt |
|----------------|--|
| 1 2 3 3 2 1 | Boxes are equal |
| 2 2 3 3 2 1 | The first box is larger than the second one |
| 2 2 3 3 2 3 | The first box is smaller than the second one |
| 3 4 5 2 4 6 | Boxes are incomparable |

om_3. Вы можете вспомнить хоть одного своего знакомого до двадцатилетнего возраста, который в детстве не играл в компьютерные

игры? Если да, то может быть вы и сами не знакомы с этим развлечением? Впрочем, трудностей при решении этой задачи это создать не должно. Во многих старых играх с двумерной графикой можно столкнуться с подобной ситуацией. Какой-нибудь герой прыгает по платформам (или островкам), которые висят в воздухе. Он должен перебраться от одного края экрана до другого. При этом при прыжке с одной платформы на соседнюю, у героя уходит $|y_2 - y_1|$ единиц энергии, где y_1 и y_2 – высоты, на которых расположены эти платформы. Кроме того, у героя есть суперприем, который позволяет перескочить через платформу, но на это затрачивается $3 * |y_3 - y_1|$ единиц энергии. Конечно же, энергию следует расходовать максимально экономно. Предположим, что вам известны координаты всех платформ в порядке от левого края до правого. Сможете ли вы найти, какое минимальное количество энергии потребуется герою, чтобы добраться с первой платформы до последней?

В первой строке входного файла `input.txt` записано количество платформ n ($1 \leq n \leq 30000$). Вторая строка содержит n натуральных чисел, не превосходящих 30000 – высоты, на которых располагаются платформы.

В выходной файл `output.txt` запишите единственное число – минимальное количество энергии, которую должен потратить игрок на преодоление платформ (конечно же в предположении, что читкоды использовать нельзя).

от_4. В отделе работают 3 сотрудника, которые получают заработную плату в рублях. Требуется определить: на сколько зарплата самого высокооплачиваемого из них отличается от самого низкооплачиваемого.

Входные данные.

В единственной строке входного файла `INPUT.TXT` записаны размеры зарплат всех сотрудников через пробел. Каждая заработная плата это число, не превышающее 10^5 .

Выходные данные.

В выходной файл `OUTPUT.TXT` необходимо вывести целое число – разницу между минимальной и максимальной зарплатой.

от_5. Шахматы

(Время: 1 сек. Память: 16 Мб. Сложность: 18 %)

Совсем недавно Вася занялся программированием и решил реализовать собственную программу для игры в шахматы. Но у него возникла проблема определения правильности хода конем, который делает пользователь. То есть если пользователь вводит значение "C7–D5", то программа должна определить это как правильный ход, если же введено "E2–E4", то ход неверный. Так же нужно проверить корректность записи ввода: если, например, введено "D9–N5", то программа должна определить данную запись как ошибочную. Помогите ему осуществить эту проверку.

Входные данные

В единственной строке входного файла INPUT.TXT записан текст хода, который указал пользователь. Пользователь не может ввести строку, длиннее 5 символов.

Выходные данные

В выходной файл OUTPUT.TXT нужно ввести "YES", если указанный ход конем верный, если же запись корректна (в смысле записи координат), но ход невозможен, то нужно вывести "NO". Если же координаты не определены или заданы некорректно, то вывести сообщение "ERROR".

Примеры

| N | INPUT.TXT | OUTPUT.TXT |
|---|-----------|------------|
| 1 | C7 – D5 | YES |
| 2 | E2 – E4 | NO |
| 3 | BSN | ERROR |

от_6. Сбор Черники

(Время: 1 сек. Память: 16 Мб Сложность: 17%)

В фермерском хозяйстве в Карелии выращивают чернику. Она растет на круглой грядке, причем кусты высажены только по окружности. Таким образом, у каждого куста есть ровно два соседних. Всего на грядке растет N кустов.

Эти кусты обладают разной урожайностью, поэтому ко времени сбора на них выросло различное число ягод – на i -ом кусте выросло a_i ягод.

В этом фермерском хозяйстве внедрена система автоматического сбора черники. Эта система состоит из управляющего модуля и нескольких собирающих модулей. Собирающий модуль за один заход, находясь непосредственно перед некоторым кустом, собирает ягоды с этого куста и с двух соседних с ним.

Напишите программу для нахождения максимального числа ягод, которое может собрать за один заход собирающий модуль, находясь перед некоторым кустом заданной во входном файле грядки.

ВХОДНЫЕ ДАННЫЕ

Первая строка входного файла INPUT.TXT содержит целое число N ($3 \leq N \leq 1000$) – количество кустов черники. Вторая строка содержит N целых положительных чисел a_1, a_2, \dots, a_n – число ягод черники, растущее на соответствующем кусте. Все a_i не превосходят 1000.

ВЫХОДНЫЕ ДАННЫЕ

В выходной файл OUTPUT.TXT выведите ответ на задачу.

ПРИМЕРЫ

| N | INPUT.TXT | OUTPUT.TXT |
|---|--------------|------------|
| 1 | 4 1 2 3 4 | 9 |
| 2 | 3 1 2 3 | 6 |

от_7. Золото племени АББА.

Время: 1 сек. Память: 16 Мб Сложность: 40%

Главный вождь племени Абба не умеет считать. В обмен на одну из его земель вождь другого племени предложил ему выбрать одну из трех куч с золотыми монетами. Но вождю племени Абба хочется получить наибольшее количество золотых монет. Помогите вождю сделать правильный выбор!

Входные данные

В первой строке входного файла INPUT.TXT записаны три натуральных числа через пробел. Каждое из чисел не превышает 10^{100} .

Выходные данные

В выходной файл OUTPUT.TXT нужно вывести одно целое число – максимальное количество монет, которые может взять вождь.

Примеры

| INPUT.TXT | OUTPUT.TXT |
|-----------------------------------|------------------------|
| 5 7 3 | 7 |
| 987531 234 86364 | 987531 |
| 189285 283 4958439238923098349024 | 4958439238923098349024 |

om_8.

В наши дни в космосе находятся сотни спутников, и все они обмениваются данными. При этом система распознавания сигналов работает по схеме <Свой-Чужой>. Один из спутников отправляет запрос другому спутнику в формате двух целых чисел, а второй спутник отвечает первому так же двумя целыми числами. Первые два числа первого спутника представляют собой сумму цифр и количество цифр тех двух чисел, которыми должен ответить второй спутник. При этом в качестве ответа должны получиться числа, представляющие наибольшее и наименьшее возможные значения, которые могут быть сформированы по описанному выше методу.

Во входном файле INPUT.TXT записаны 2 натуральных числа S и K , представляющих сумму и количество цифр соответственно ($K \leq 100$). При этом гарантируется, что возможно составить хотя бы одно K -значное число, сумма цифр которого равна S .

В выходной файл OUTPUT.TXT выведите два числа – ответ второго спутника. При этом следует помнить, что все числа не имеют лидирующих нулей.

ПРИМЕР

| Input.txt | Output.txt |
|-----------|------------|
| 1 3 | 100 100 |
| 3 4 | 3000 1002 |

om_9.

Петя очень любит наблюдать за электронными часами. Он целыми днями смотрел на часы и считал, сколько раз встречается каждая цифра. Через несколько месяцев он научился по любому промежутку времени говорить, сколько раз на часах за это время встретится каждая цифра, и очень гордился этим. Вася решил проверить Петю, но он не знает, как это сделать.

Первая и вторая строки входного файла input.txt содержат начало и конец промежутка времени соответственно. Начальное время не превосходит конечное. Время задается в формате hh:mm:ss ($0 \leq hh \leq 24$, $0 \leq mm < 60$, $0 \leq ss < 60$). hh, mm, ss дополнены ведущими нулями до двух символов. Эти нули также учитываются при подсчете числа цифр.

Выходной файл output.txt должен содержать 10 строк. В i -ой строке должно быть написано, сколько раз встречается цифра $i - 1$.

om_10. Во время военного похода на морского пехотинца Джо было возложено ответственное задание – развести костер и поддерживать в нем огонь ровно M минут. Для этого у Джо есть спички и N поленьев, причем Джо известно точное время сгорания каждого полена. Джо разжигает огонь в момент времени $t = 0$ и сразу бросает в него одно или несколько поленьев. Затем он должен подбрасывать в огонь новые поленья, не позволяя костру угаснуть (т. е. если последнее полено в костре догорает в момент времени t , то новое полено может быть брошено в огонь не позднее $t - 1$). Поленья, брошенные в огонь, загораются мгновенно. Одновременно Джо может бросить в огонь любое количество поленьев. Джо должен бросить в огонь все N поленьев.

Первая строка входного файла Input.txt содержит целые числа N и M ($1 \leq N \leq 100, 1 \leq M \leq 1000$) – количество поленьев и время, в течение которого Джо должен поддерживать огонь в костре.

Вторая строка входного файла содержит N целых чисел A_1, A_2, \dots, A_n , где A_i ($2 \leq A_i \leq 1000$) – время сгорания i -ого полена в минутах.

В выходной файл Output.txt выведите строку "yes", если Джо сможет поддерживать огонь в костре ровно M минут, и строку "no" в противном случае.

| Input.txt | Output.txt |
|-----------------|------------|
| 3 7 2 3 5 | yes |
| 2 5 3 9 | no |
| 4 10 3 3 3 3 | no |

om_11. Драконы

(Время: 1 сек. Память: 16 Мб. Сложность: 40%)

Известно, что у дракона может быть несколько голов и его сила определяется числом голов. Вероятно, вы считаете, что это значение вычисляется как сумма всех голов? Это далеко не так, иначе было бы слишком просто вычислить силу драконьей стаи. Оказывается, что искомое значение равно произведению значений числа голов каждого из драконов. Например, если в стае 3 дракона, у которых 3, 4, 5 голов соответственно, то сила равна $3 \cdot 4 \cdot 5 = 60$. Предположим, что нам известно суммарное значение голов каждой стаи, как нам вычислить максимальное возможное значение силы этого логова драконов? Именно эту задачу Вам и

предстоит решить.

Входные данные

В единственной строке входного файла *input.txt* записано натуральное число N ($0 < N < 100$) – количество голов драконьей стаи.

Выходные данные

В единственную строку выходного файла *output.txt* нужно вывести максимально возможное значение силы, которая может быть у стаи драконов из N голов. И записать комбинацию множителей, дающих этот результат.

Примеры

| N | INPUT.TXT | OUTPUT.TXT |
|---|-----------|---------------------------|
| 1 | 6 | $9 = 3 * 3$ |
| 2 | 8 | $18 = 2 * 3 * 3$ |
| 3 | 13 | $108 = 2 * 2 * 3 * 3 * 3$ |

om_12. Для игры в "Поле чудес" используется круглый барабан, разделенный на сектора, и стрелка. В каждом секторе записано некоторое число. В различных секторах может быть записано одно и то же число. Однажды ведущий игры решил изменить правила. Он сам стал вращать барабан и называть игроку (который барабана не видел) все числа подряд в том порядке, в котором на них указывала стрелка в процессе вращения барабана. Получилось так, что барабан сделал целое число оборотов, то есть последний сектор совпал с первым. После этого, ведущий задал участнику вопрос: какое наименьшее число секторов может быть на барабанае?

Требуется написать программу, отвечающую на этот вопрос ведущего.

В первой строке входного файла *input.txt* записано число N – количество чисел, которое назвал ведущий ($2 \leq N \leq 30000$). Во второй строке записано N чисел, на которые указывала стрелка в процессе вращения барабана. Первое число всегда совпадает с последним (в конце стрелка указывает на тот же сектор, что и в начале). Числа, записанные в секторах барабана – натуральные, не превышающие 32000.

В выходной файл *output.txt* необходимо вывести одно число – минимальное число секторов, которое может быть на барабанае.

Пример

| INPUT.TXT | OUTPUT.TXT |
|---------------------------------|------------|
| 13 5 3 1 3 5 2 5 3 1 3 5 2 5 | 6 |

5 Целочисленная арифметика [tsa_3]

- tsa_1.** Требуется вывести представление целого числа N в виде произведения простых чисел. Входной файл `input.txt` содержит натуральное число N ($2 \leq N \leq 2^{31} - 1$). В выходной файл `output.txt` выведите список простых множителей числа N в порядке неубывания, разделенных знаком `<*>`.

| Input.txt | Output.txt |
|-----------|------------|
| 5 | 5 |
| 30 | 2*3*5 |

- tsa_2.** Известно, что любое четное число, большее 2, представимо в виде суммы 2 простых чисел, причем таких разложений может быть несколько. Впервые гипотезу о существовании данного разложения сформулировал математик Х. Гольбах.

Требуется написать программу, производящую согласно утверждению Гольбаха, разложение заданного четного числа. Из всех пар простых чисел, сумма которых равна заданному числу, требуется найти пару, содержащую наименьшее простое число.

Входной файл `INPUT.TXT` содержит четное число N ($4 \leq N \leq 0998$).

В выходной файл `OUTPUT.TXT` необходимо вывести два простых числа, сумма которых равна числу N . Первым выводится наименьшее число.

Пример

| Input.txt | Output.txt |
|-----------|------------|
| 992 | 73 919 |

- tsa_3.** Катя решила пригласить к себе в гости n друзей. Так как ее друзья очень любят фрукты, то в качестве угощения для них она купила m одинаковых апельсинов. Она хочет разрезать каждый апельсин на одинаковое число равных долек так, чтобы их можно было распределить между гостями (сама Катя апельсины есть не будет), и всем гостям досталось поровну долек. Напишите программу, которая вычисляет минимальное количество долек, на которое необходимо разрезать каждый апельсин, чтобы были выполнены указанные выше условия.

Входной файл `input.txt` содержит два положительных целых числа n и m ($1 \leq n, m \leq 10^9$).

В выходной файл *output.txt* выведите ответ на задачу.

Пример

| Input.txt | Output.txt |
|-----------|------------|
| 2 5 | 2 |
| 2 4 | 1 |

6 Двумерные массивы [dm_8]

dm_1. Винни-Пух

Время: 1 сек. Память: 16 Мб Сложность: 41%

Винни-Пух стоит на прямоугольном поле размером $N \times M$ клеток. В каждой клетке растет по одной ягоде. В начальный момент времени он стоит на левой верхней клетке. Он начинает собирать ягоды по верхнему краю поля. Если он доходит до края поля или до пустой клетки, он поворачивается на 90 градусов вправо и продолжает собирать ягоды. Но дойдя до очередной клетки Вيني вспоминает, что его ждет Пятачок, и он уходит с поля.

Входные данные

В первой строке входного файла INPUT.TXT стоят размеры N и M ($0 < N, M \leq 100$) – высота и ширина, во второй – числа Y и X ($0 < Y \leq N, 0 < X \leq M$) – номера строки и столбца клетки, дойдя до которой Вيني-Пух прекращает собирать ягоды.

Выходные данные

В выходной файл OUTPUT.TXT выведите число ягод, которые соберет Пух.

Примеры

| N | INPUT.TXT | OUTPUT.TXT |
|---|------------|------------|
| 1 | 1 1 1 1 | 1 |
| 2 | 3 3 2 3 | 4 |
| 3 | 5 5 2 3 | 18 |

dm_2. Для заданных натуральных чисел N и K требуется вычислить количество чисел от 1 до N , имеющих в двоичной записи ровно K нулей. Например, если $N = 8$ и $K = 1$, то мы можем записать все числа от 1 до 8 в двоичной системе счисления: 1, 10, 11, 100, 101, 110, 111 и 1000. Откуда видно, что только числа 10, 101 и 110 имеют ровно один ноль в записи, т. е. правильный ответ - 3.

Входные данные

В единственной строке входного файла INPUT.TXT записано два натуральных числа через пробел N и K , не превышающих 10^9 .

Выходные данные

В единственную строку файла OUTPUT.TXT нужно вывести одно

целое число – количество чисел от 1 до N с K нулями в двоичном представлении.

Примеры

| INPUT.TXT | OUTPUT.TXT |
|-----------|------------|
| 8 1 | 3 |
| 13 2 | 4 |
| 1000 5 | 210 |

dm_3. Известный художник решил написать новый шедевр. После многих дней усердной работы он захотел исследовать свое творение. Художник вспомнил, что картина писалась следующим образом: сначала был взят белый холст, имеющий форму прямоугольника шириной w и высотой h . Затем художник нарисовал на этом холсте n прямоугольников со сторонами, параллельными сторонам холста и вершинами, расположенными в целочисленных координатах. Помогите художнику определить площадь незакрашенной части холста.

Первая строка входного файла *input.txt* содержит два натуральных числа w и h ($1 \leq w, h \leq 100$). Во второй строке записано целое число n ($1 \leq n \leq 5000$) – количество прямоугольников. Следующие n строк содержат информацию о всех прямоугольниках. Каждая строка описывает один прямоугольник в виде четырех чисел x_1, y_1, x_2, y_2 , где (x_1, y_1) и (x_2, y_2) – координаты левого верхнего и правого нижнего угла прямоугольника соответственно.

Вывести в выходной файл *output.txt* одно целое число – площадь незакрашенной части холста.

Примеры

| Input.txt | Output.txt |
|---|------------|
| 5 5 2 1 1 3 3 2 2 4 4 | 18 |
| 6 7 3 0 0 5 5 1 1 4 4 2 2 3 3 | 17 |

dm_4. Вова стоит перед лесенкой из N ступеней. На каждой из ступеней написаны произвольные целые числа. Первым шагом Вова может перейти на первую ступень или, перепрыгнув через первую, сразу оказаться на второй. Также он поступает и дальше, пока не

достигнет N -ой ступени. Посчитаем сумму всех чисел, написанных на ступенях через которые прошел Вова. Требуется написать программу, которая определит оптимальный маршрут Вовы, при котором, шагая, он получит наибольшую сумму.

Входной файл *INPUT.TXT* содержит в первой строке натуральное число N – количество ступеней лестницы. Во второй строке через пробел заданы числа, написанные на ступенях лестницы, начиная с первой. Количество ступеней не превышает 1000, числа, написанные на ступенях, не превосходят по модулю 1000.

Выходной файл *OUTPUT.TXT* должен содержать в первой строке наибольшее значение суммы. Во второй строке должны быть записаны через пробел номера ступеней по возрастанию, по которым должен шагать Вова.

dm_5. В неориентированном графе требуется найти длину кратчайшего пути между двумя вершинами.

Во входном файле *input.txt* записано сначала число N – количество вершин в графе ($1 \leq N \leq 100$). Затем записана матрица смежности (0 обозначает отсутствие ребра, 1 – наличие ребра). Затем записаны номера двух вершин – начальной и конечной.

В выходной файл *output.txt* вывести длину кратчайшего пути. Если пути не существует, вывести -1 (минус один).

| Input.txt | Output.txt |
|-----------|------------|
| 5 | 3 |
| 0 1 0 0 1 | |
| 1 0 1 0 0 | |
| 0 1 0 0 0 | |
| 0 0 0 0 0 | |
| 1 0 0 0 0 | |
| 3 5 | |

dm_6. В клубе N человек. Многие из них – друзья. Так же известно, что друзья друзей так же являются друзьями. Требуется выяснить, сколько всего друзей у конкретного человека в клубе.

В первой строке входного файла *Input.txt* заданы два числа:

$$N \text{ и } S \ (1 \leq N \leq 100; 1 \leq S \leq N),$$

где N – количество человек в клубе, а S – номер конкретного человека. В следующих N строках записано по N чисел – матрица смежности, состоящая из единиц и нулей. Причем единица, стоящая в i -й строке и j -м столбце гарантирует, что люди с номерами

i и j – друзья, а 0 – выражает неопределенность.

В выходной файл *Output.txt* вывести количество гарантированных друзей у человека с номером S , помня о транзитивности дружбы.

| Input.txt | Output.txt |
|-----------|------------|
| 0 1 0 | 2 |
| 1 0 1 | |
| 0 1 0 | |

dm_7. Заполните матрицу $n \times n$ числами от 1 до n^2 по спирали, по часовой стрелке, начиная от единицы в верхнем левом углу. Полученную матрицу выведите построчно, разделяя числа пробелами. Программа должна считать число n из файла *input.txt* и вывести результат в файл *output.txt*.

Входной файл содержит одно целое число n ($1 \leq n \leq 10$).

Выведите в выходной файл n строк по n чисел – полученную матрицу.

Пример

| input.txt | output.txt |
|-----------|---|
| 5 | 1 2 3 4 5 16 17 18 19 6 15 24 25 20 7 14 23 22 21 8 13 12 11 10 9 |

dm_8. Только вправо или вниз

(Время: 1 сек. Память: 16 Мб Сложность: 32%)

Игровое поле $N \times M$ заполняется целыми числами, одно неотрицательное целое число в каждой клетке. Цель игры состоит в том, чтобы пройти по любому разрешенному пути от верхнего левого угла до правого нижнего. Целое число в каждой клетке указывает, какой длины шаг должен быть из текущей клетки. Все шаги могут быть или направо или вниз. Если в результате какого-либо шага игрок покидает пределы поля, такой шаг запрещается.

На рис. 1 приведен пример игрового поля 3×4 , где сплошная окружность показывает положение начала (см. условие задачи в сборнике), а пунктирная окружность – цель. Рис. 2 показывает три возможных пути от начала до цели для рассматриваемого примера игрового поля, с удаленными промежуточными числами.

| | | | |
|---|---|---|----|
| 2 | 1 | 1 | 2 |
| 3 | 2 | 1 | 44 |
| 3 | 1 | 1 | 0 |

Рис. 1.

| | | | |
|---|--|--|---|
| 2 | | | |
| | | | |
| 3 | | | 0 |

Рис. 2.

| | | | |
|---|--|---|---|
| 2 | | 1 | |
| | | 1 | |
| | | 1 | 0 |

| | | | |
|---|--|---|---|
| 2 | | 1 | 2 |
| | | | |
| | | | 0 |

Требуется написать программу, которая определит число различных вариантов путей от верхнего левого угла до правого нижнего.

Входные данные

Входной файл *input.txt* содержит в первой строке размеры поля N ($1 \leq N \leq 70$) и M ($1 \leq M \leq 70$). В последующих N строках входного файла, каждая из которых описывает отдельную строку игрового поля, записаны через пробел по M целых чисел – длины шагов из клеток данной строки.

Выходные данные

Выходной файл *output.txt* должен содержать одно число – число различных вариантов путей от верхнего левого угла до правого нижнего. Для каждого поля будет менее чем 2^{31} различных путей.

Пример

| N | input.txt | output.txt |
|---|---------------------------------------|------------|
| 1 | 1 4 2 1 1 2 3 2 1 44 3 1 1 0 | 3 |

7 Обработка строк [str_24]

str_1. Задано время отправления поезда и время в пути до конечной станции. Требуется написать программу, которая найдет время прибытия этого поезда (возможно, в другие сутки).

Входной файл Input.txt содержит две строки. В первой строке задано время отправления, а во второй строке – время в пути. Время отправления задается в формате <НН:ММ>, где НН время в часах, которое принимает значение от 00 до 23, ММ – время в минутах, которое принимает значение от 00 до 59. Время в пути задается двумя неотрицательными целыми числами – количество часов и количество минут. Числа разделяются одним пробелом. Количество часов не превышает 120, минут – 59.

Выходной файл Output.txt должен содержать одну строку – время прибытия поезда на конечную станцию. Формат вывода этого времени совпадает с форматом ввода времени отправления.

| Input.txt | Output.txt |
|----------------|------------|
| 00:00 10 10 | 10:10 |
| 01:02 4 6 | 05:08 |
| 11:00 22 0 | 09:00 |

str_2. Составить программу, удаляющую одну цифру из N -значного числа, такую, чтобы плюс-минус сумма была наибольшей. Плюс-минус сумма – это сумма с чередованием цифр числа с разными знаками: для числа 764 это $+7-6+4$.

Если удалить цифру 7, то будет $+6-4=2$,
если удалить цифру 6, то будет $+7-4=3$,
если удалить цифру 4, то будет $+7-6=1$.

При этом видно, что максимум достигается при удалении средней цифры 6 и равен 3.

Во входном файле INPUT.TXT записано натуральное N -значное число ($2 \leq N \leq 50$).

В выходной файл OUTPUT.TXT выведите значение наибольшей суммы.

str_3.

В теории формальных грамматик и автоматов (ТФГиА) важную роль играют так называемые контекстно-свободные грамматики (КС-грамматики).

КС-грамматикой будем называть четверку, состоящую из множества N нетерминальных символов, множества T терминальных символов, множества P правил (продукций) и начального символа S , принадлежащего множеству N . Каждая продукция p из P имеет форму $A \rightarrow a$, где A нетерминальный символ (A из N), а a – строка, состоящая из терминальных и нетерминальных символов. Процесс вывода слова начинается со строки, содержащей только начальный символ S .

После этого на каждом шаге один из нетерминальных символов, входящих в текущую строку, заменяется на правую часть одной из продукций, в которой он является левой частью. Если после такой операции получается строка, содержащая только терминальные символы, то процесс вывода заканчивается.

Во многих теоретических задачах удобно рассматривать так называемые нормальные формы грамматик. Процесс приведения грамматики к нормальной форме часто начинается с устранения левой рекурсии. В этой задаче мы будем рассматривать только ее частный случай, называемый непосредственной левой рекурсией.

Говорят, что правило вывода $A \rightarrow R$ содержит непосредственную левую рекурсию, если первым символом строки R является A .

Задана КС-грамматика. Требуется Найти количество правил, содержащих непосредственную левую рекурсию.

Первая строка входного файла *INPUT.TXT* содержит количество n ($1 \leq n \leq 1000$) правил в грамматике. Каждая из последующих n строк содержит по одному правилу. Нетерминальные символы обозначаются заглавными буквами латинского алфавита, терминальные – строчными. Левая часть продукции отделяется от правой символами \rightarrow . Правая часть продукции имеет длину от 1 до 30 символов.

В выходной файл *OUTPUT.TXT* выведите ответ на задачу.

Пример

| Input.txt | Output.txt |
|-----------|------------|
| 3 | 2 |
| S->Sabc | |
| S->A | |
| A->AA | |

str_4. Статистика

(Время: 1 сек. Память: 16 Мб. Сложность: 25%)

Вася не любит английский язык, но каждый раз старается получить хотя бы четверку за четверть, чтобы оставаться ударником. В текущей четверти Вася заметил следующую закономерность: по нечетным дням месяца он получал тройки, а по четным – четверки. Так же он помнит, в какие дни он получал эти оценки. Поэтому он выписал на бумажке все эти дни для того, чтобы оценить, сколько у него троек и сколько четверок. Помогите Васе это сделать, расположив четные и нечетные числа в разных строчках. Вася может рассчитывать на оценку 4, если четверок не меньше, чем троек.

Входные данные

В первой строке входного файла *input.txt* записано единственное число N – количество элементов массива ($1 \leq N \leq 100$). Вторая строка содержит N чисел, представляющих заданный массив. Каждый элемент массива – натуральное число от 1 до 31. Все элементы массива разделены пробелом.

Выходные данные

В первую строку выходного файла *output.txt* нужно вывести числа, которые соответствуют дням месяцев, в которые Вася получил тройки, а во второй строке соответственно расположить числа месяца, в которые Вася получил четверки. В третьей строке нужно вывести "YES", если Вася может рассчитывать на четверку и "NO" в противном случае. В каждой строчке числа следует выводить в том же порядке, в котором они идут во входных данных. При выводе числа отделяются пробелом.

Примеры

| N | INPUT.TXT | OUTPUT.TXT |
|---|---------------------------|-------------------------------|
| 1 | 5 4 16 19 31 2 | 19 31 4 16 2 YES |
| 2 | 8 29 4 7 12 15 17 24 1 | 29 7 15 17 1 4 12 24 NO |

str_5.

В доме Вилли установили скоростной лифт новой экспериментальной модели. В этом лифте кнопки с номерами этажей заменены двумя другими кнопками. При нажатии на первую кнопку лифт поднимается на один этаж вверх, а при нажатии на вторую –

опускается на один этаж вниз.

Младшему брату Вилли Дилли очень нравится кататься на новом лифте. Он катается на нем до тех пор, пока не побывает на каждом из этажей хотя бы по одному разу. После этого Дилли довольный возвращается домой. Зная порядок, в котором Дилли нажимал на кнопки лифта, попробуйте определить общее количество этажей в доме Вилли и Дилли.

Первая строка входного файла *input.txt* содержит последовательность нажатий на кнопки лифта. Символ <1> означает, что была нажата первая кнопка, а символ <2> – что была нажата вторая кнопка. Символы <1> и <2> не разделены пробелами. Количество нажатий не превосходит 100. Гарантируется, что лифт никогда не опускался ниже первого и не поднимался выше последнего этажа.

В выходной файл *output.txt* следует вывести одно число – количество этажей в доме Вили и Дилли.

Пример

| Input.txt | Output.txt |
|-----------|------------|
| 21212 | 2 |

str_6. Заданы натуральные числа A, B, C . Требуется написать программу, которая найдет общие цифры в этих числах.

Входной файл *INPUT.TXT* содержит три натуральных числа A, B, C ($1 \leq A, B, C \leq 10^{80}$). Числа разделены одним пробелом.

Выходной файл *OUTPUT.TXT* должен содержать в первой строке количество общих цифр, а во второй строке в порядке возрастания через один пробел общие цифры.

Пример

| Input.txt | Output.txt |
|----------------|------------|
| 1234 2345 3456 | 2 |
| | 3 4 |

str_7. Задано натуральное число N . Требуется написать программу, вычисляющую количество различных трехзначных чисел получающихся из N вычеркиванием цифр из его десятичной записи.

Входной файл *Input.txt* содержит одно натуральное число N ($1 \leq N \leq 10^{100}$).

Выходной текстовый файл *Output.txt* должен содержать одно натуральное число – найденное количество трехзначных чисел.

Пример

| Input.txt | Output.txt |
|-----------------------|------------|
| 12 | 0 |
| 111111111110011111111 | 4 |

str_8.

Программист Билл занимается разработкой программного обеспечения для новейшего робота-исследователя, которого ученые планируют отправить на Марс с целью поиска там следов разумной жизни. Модули, которые отвечают за передвижение робота и сбор проб грунта, Билл уже скачал из Интернета. Оставалось лишь научить робота отличать разумные формы жизни от неразумных. Для этого Боб несколько месяцев посещал программистские форумы, и, наконец, нашел подходящий модуль. Теперь, чтобы определить, является ли тот или иной объект представителем внеземной расы, роботу достаточно сравнить два вещественных числа.

Входной файл *input.txt* состоит из двух строк, в каждой из которых записано по одному вещественному числу без ведущих нулей. Целая и дробная части отделяются точкой, которая может быть опущена, если число целое. Каждое из чисел содержит не более 10000 цифр.

В выходной файл *output.txt* выведите один символ $C < T$, если первое число меньше второго, $C > T$, если больше, и $C = T$, если числа равны.

str_9.

Петя разгадывает головоломку, которая устроена следующим образом. Дана квадратная таблица размера $N \times N$, в каждой клетке которой записана какая-нибудь латинская буква. Кроме того, дан список ключевых слов. Пете нужно, взяв очередное ключевое слово, найти его в таблице. То есть найти в таблице все буквы этого слова, причем они должны быть расположены так, чтобы клетка, в которой расположена каждая последующая буква слова, была соседней с клеткой, в которой записана предыдущая буква (клетки называются соседними, если они имеют общую сторону – то есть соседствуют по вертикали или по горизонтали). Например, в примере показано, как может быть расположено в таблице слово *olympiad*.

```
p O L t e
r w Y M s
o A I P t
b D a n r
```

Л е м е с

Когда Петя находит слово, он вычеркивает его из таблицы. Использовать уже вычеркнутые буквы в других ключевых словах нельзя. После того, как найдены и вычеркнуты все ключевые слова, в таблице остаются еще несколько букв, из которых Петя должен составить слово, зашифрованное в головоломке. Помогите Пете в решении этой головоломки, написав программу, которая по данной таблице и списку ключевых слов выпишет, из каких букв Петя должен сложить слово, то есть какие буквы останутся в таблице после вычеркивания ключевых слов.

Во входном файле *input.txt* записаны два числа N ($1 \leq N \leq 10$) и M ($0 \leq M \leq 200$). Следующие N строк по N заглавных латинских букв описывают ребус. Следующие M строк содержат слова. Слова состоят только из заглавных латинских букв, каждое слово не длиннее 200 символов.

Гарантируется, что в таблице можно найти и вычеркнуть по описанным выше правилам все ключевые слова.

В выходной файл *output.txt* выведите в алфавитном порядке оставшиеся в таблице буквы.

Пример

| Input.txt | Output.txt |
|---|------------|
| 5 3 POLTE RWYMS OAIPT BDANR LEMES OLYMPIAD PROBLEM TEST | AENRSW |

str_10. Нули

(Время: 1 сек. Память: 16 Мб Сложность: 16%)

Требуется найти самую длинную непрерывную цепочку нулей в последовательности нулей и единиц.

Входные данные

В единственной строке входного файла INPUT.TXT записана последовательность нулей и единиц (без пробелов). Суммарное количество цифр не превышает 100.

Выходные данные

В единственную строку выходного файла OUTPUT.TXT нужно вывести искомую длину цепочки нулей.

Пример

| INPUT.TXT | OUTPUT.TXT |
|-----------------|------------|
| 100101110000110 | 4 |

str_11. Задано натуральное число N . Требуется написать программу, которая найдет следующее за ним число, в двоичном разложении которого столько же единиц, сколько в двоичном разложении числа N .

Входной файл *input.txt* содержит одно натуральное число N ($N \leq 2^{30}$).

Выходной файл *Output.txt* должен содержать ответ на задачу.

Пример

| Input.txt | Output.txt |
|-----------|------------|
| 3 | 5 |

str_12. Шаблоном размера n назовем строку длины n , каждый из символов которой входит в множество 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f, g, ?

Шаблоны преобразуются в строки из цифр по следующим правилам:

- символы от 0 до 9 могут быть преобразованы только сами в себя;
- символ a может преобразован в любой из символов 0, 1, 2, 3;
- символ b может преобразован в любой из символов 1, 2, 3, 4;
- символ c может преобразован в любой из символов 2, 3, 4, 5;
- символ d может преобразован в любой из символов 3, 4, 5, 6;
- символ e может преобразован в любой из символов 4, 5, 6, 7;
- символ f может преобразован в любой из символов 5, 6, 7, 8;
- символ g может преобразован в любой из символов 6, 7, 8, 9;
- символ ? может преобразован в любой из символов от 0 до 9;

Даны два шаблона: r_1 и r_2 . Рассмотрим множество S_1 строк, которые могут быть получены из r_1 по описанным правилам, и

множество S_w строк, которые могут быть получены из p_2 . Необходимо найти количество строк, входящих в оба этих множества. Первая строка входного файла *input.txt* содержит шаблон p_1 , вторая – шаблон p_2 . Шаблоны имеют одинаковый положительный размер, не больше 9.

В выходной файл *output.txt* выведите ответ на задачу.

Пример

| Input.txt | Output.txt |
|-----------|------------|
| abc | 0 |
| 999 | |

str_13. Вычислительная биология

В современной биологии ученым часто приходится иметь дело с последовательностями ДНК. Эти последовательности зачастую являются очень длинными, и их ручная обработка требует большого количества времени и сил. Поэтому возникает идея автоматизировать этот процесс.

Для этого можно применять компьютерные методы обработки данных, например, весьма полезными оказываются алгоритмы на строках.

В этой задаче последовательность ДНК будет представляться в виде строки, все символы которой входят в множество $[A, G, C, T]$.

Пусть даны две последовательности ДНК:

$$s = s_1 s_2 \dots s_n \text{ и}$$

$$t = t_1 t_2 \dots t_m.$$

Будем говорить, что t может получиться в результате эволюции из s , если s является подпоследовательностью t , то есть существует такая последовательность индексов

$$1 \leq i_1 < i_2 < \dots < i_n \leq m, \text{ что}$$

$$s_1 = t_{i_1}, s_2 = t_{i_2}, \dots, s_n = t_{i_n}.$$

Необходимо выяснить, может ли последовательность t получиться в результате эволюции из s .

Первая строка входного файла *input.txt* содержит последовательность s , вторая – последовательность t . Размер входного файла не превосходит 256 килобайт.

В выходной файл *output.txt* выведите слово YES, если последовательность t могла получиться в результате эволюции из s , и слово NO – иначе.

| | |
|-----------------|------------|
| Input.txt | Output.txt |
| GTA AGCTA | YES |
| AAAG GAAAAAT | NO |

str_14. На вход программе подается текст, состоящий не более чем из 200 символов. Текст был зашифрован следующим образом. Сначала определили количество букв в самом коротком слове, обозначив полученное число N (словом называется непрерывная последовательность английских букв, слова друг от друга отделяются любыми другими символами). Затем заменили каждую английскую букву в тексте на следующую за ней N -ую по счету в алфавите (алфавит считается циклическим, за буквой Z следует буква A), оставив другие символы неизменными. Строчные буквы при этом остались строчными, а прописные – прописными. Напишите программу, которая будет выводить расшифрованный текст. Программа должна считать входные данные из файла *input.txt* и вывести результат в файл *output.txt*.

ФОРМАТ ВХОДНОГО ФАЙЛА

Входной файл содержит одну строку с текстом. Длина строки не превышает 200 символов.

ФОРМАТ ВЫХОДНОГО ФАЙЛА

Выведите в выходной файл ответ – расшифрованный текст.

ПРИМЕР

| | |
|--|--|
| input.txt | output.txt |
| Jgnnq yqtnf! Vjku oguucig ku xgta ugetgv. | Hello world! This message is very secret. |

str_15. Известно, что основанием позиционной системы счисления называют количество различных символов, используемых для записи чисел в данной системе счисления. Для записи чисел в b -ричной системе счисления, где $b \leq 36$, могут быть использованы первые b символов из следующего списка

$$0, 1, \dots, 9, A, B, \dots, Z.$$

Например, для записи чисел в троичной системы используются символы 0, 1, 2, а в двенадцатеричной –

$$0, 1, \dots, 9, A, B.$$

Требуется написать программу, которая по входной строке S определяет, является ли данная строка записью числа в системе счис-

ления, с основанием не большим 36, и, если является, определит минимальное основание этой системы счисления.

Входной файл *input.txt* содержит в единственной строке входную строку. Длина строки не превышает 255. Все символы строки имеют коды от 32 до 127.

Выходной файл *output.txt* должен содержать одно число. Если строка является записью числа в некоторой системе счисления, то нужно вывести минимальное основание такой системы счисления. Иначе вывести -1.

str_16. Требуется найти разность между неотрицательными числами A и B. Во входном файле *input.txt* в двух строках записаны два неотрицательных целых числа A и B, не превышающие 10^{1000} . В выходной файл *output.txt* вывести значение A - B.

str_17. Статистика.

(Время: 1 сек. Память: 16 Мб. Сложность: 25 %)

Вася не любит английский язык, но каждый раз старается получить хотя бы четверку за четверть, чтобы оставаться ударником. В текущей четверти Вася заметил следующую закономерность: по нечетным дням месяца он получал тройки, а по четным – четверки. Так же он помнит, в какие дни он получал эти оценки. Поэтому он выписал на бумажке все эти дни для того, чтобы оценить, сколько у него троек и сколько четверок. Помогите Васе это сделать, расположив четные и нечетные числа в разных строчках. Вася может рассчитывать на оценку 4, если четверок не меньше, чем троек.

Входные данные

В первой строке входного файла INPUT.TXT записано единственное число N - количество элементов массива ($1 \leq N \leq 100$). Вторая строка содержит N чисел, представляющих заданный массив. Каждый элемент массива – натуральное число от 1 до 31. Все элементы массива разделены пробелом.

Выходные данные

В первую строку выходного файла OUTPUT.TXT нужно вывести

числа, которые соответствуют дням месяцев, в которые Вася получил тройки, а во второй строке соответственно расположить числа месяца, в которые Вася получил четверки. В третьей строке нужно вывести "YES", если Вася может рассчитывать на четверку и "NO" в противном случае. В каждой строчке числа следует выводить в том же порядке, в котором они идут во входных данных. При выводе, числа отделяются пробелом.

Примеры

| N | INPUT.TXT | OUTPUT.TXT |
|---|---------------------------|-------------------------------|
| 1 | 5 4 16 19 31 2 | 19 31 4 16 2 YES |
| 2 | 8 29 4 7 12 15 17 24 1 | 29 7 15 17 1 4 12 24 NO |

str_18. Пусть задано некоторое слово, состоящее из букв английского алфавита длиной не более 80 символов (например, "WORD"). Рассмотрим набор возможных перестановок, состоящих из букв данного слова (например, "RDOW", "WODR" и т. д.). Требуется выбрать из этого множества слово, следующее по алфавиту за исходным.

В единственной строке входного файла Input.txt записано слово, не последнее по алфавиту среди возможных его перестановок. В единственную строку выходного файла Output.txt нужно вывести следующее слово по алфавиту.

| Input.txt | Output.txt |
|-----------|------------|
| abdc | acbd |
| word | wrdo |

str_19. Требуется написать программу, определяющую, в каких системах счисления с основаниями от 2 до 36 это число не содержит одинаковых цифр.

Входной файл INPUT.TXT содержит одно целое число N ($1 \leq N \leq 10^9$), записанное в десятичной системе счисления.

Выходной файл OUTPUT.TXT должен содержать основания систем счисления в порядке возрастания, разделенные одним пробелом.

Пример

| Input.txt | Output.txt |
|-----------|--|
| 100 | 11 12 13 14 15 16 17 18 20 21 22 23 25 26 27 28 29 30 31 32 33 34 35 36 |

str_20. Требуется вычислить факториал целого числа N . Факториал обозначают как $N!$ и вычисляют по формуле:

$$N! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot (N - 1) \cdot N,$$

причем

$$0! = 1.$$

Так же допустимо рекуррентное соотношение:

$$N! = (N - 1)! \cdot N.$$

В единственной строке входного файла *input.txt* записано одно целое неотрицательное число N ($N < 1000$).

В выходной файл *output.txt* нужно вывести одно целое число – значение $N!$.

str_21. Секретное сообщение

(сложность 46%)

На секретную базу в Арктике поступила шифровка – последовательность из N десятичных цифр. Она содержит номер секретной базы в Антарктиде, который является последовательностью из K десятичных цифр. При этом для того, чтобы отличить его от ненужной информации, он повторен в шифровке хотя бы два раза (возможно, эти два вхождения перекрываются).

Напишите программу, которая по шифровке и длине номера секретной базы определяет, содержит ли шифровка номер базы. Учтите, что у базы может быть несколько номеров, и все они могут быть переданы в шифровке.

Входные данные

Первая строка входного файла INPUT.TXT содержит два целых числа: $N(1 \leq N \leq 10^5)$ и $K(1 \leq K \leq 5)$ – длину шифровки и длину номера секретной базы соответственно. Вторая строка содержит N цифр – шифровку. Помните, что цифры в шифровке не разделяются пробелами.

Выходные данные

В выходной файл OUTPUT.TXT выведите "YES", если шифровка содержит номер секретной базы, и "NO" в противном случае.

Примеры

| N | input.txt | output.txt |
|---|-----------------------|------------|
| 1 | 10 5 0123456789 | NO |
| 2 | 13 2 0123400056789 | YES |

str_22. Боги

(Время: 1 сек. Память: 16 Мб Сложность: 26%)

Археологами найден набор древних копий старинных манускриптов с мифами – разными историями о древних богах. К несчастью, переписчики этих манускриптов не отличались особой грамотностью и умудрились в каждом имени сделать ровно по одной орфографической ошибке – т. е. ровно одну из букв божественного имени заменили какой-то другой буквой. Археологи смогли составить список правильных написаний имен богов, так же им удалось выписать из манускриптов все имена собственные. Однако сопоставлять два списка – выше их сил. Помогите им в этом!

Входные данные

Первая строка входного файла Input.txt содержит число N – количество имен богов в списке. Следующие N строк – имена богов. Далее идет строка, содержащая число M – количество "подозрительных" слов, выписанных из манускриптов. Следующие M строк – "подозрительные" слова. Каждое из имен богов и "подозрительных" слов – последовательность из K заглавных букв английского алфавита ($1 \leq N, M, K \leq 30$).

Выходные данные

В выходной файл Output.txt выводится N чисел – для каждого божьего имени выводится число "подозрительных" слов, которые являются именем данного бога с одной ошибкой.

Пример

| N | Input. txt | Output.txt |
|---|--|------------|
| 1 | 3 ZEUS POSEIDON AFINA 4 ZEVS POSEYDON AVYNA ZERS | 2 1 0 |

str_23. В конторе <Рога и Копыта> подходит время подведения годового баланса. В бухгалтерию поступили сведения о том, что, согласно документам, суммарный расход составил A рублей, а суммарный приход – B рублей.

Поскольку с реальным положением дел эти цифры все равно не имеют ничего общего, бухгалтер решил реализовать следующую свою идею. Как известно, при наборе чисел на компьютере люди часто вводят цифры в неправильном порядке. Поэтому бухгалтер хочет найти такой способ переставить цифры в числах A и B , чтобы в результате разность $A - B$ (и, соответственно, количество денег, которые он положит к себе в карман), была максимальна, а в случае можно будет сослаться на ошибку секретаря. При этом нельзя забывать о знаке чисел и о том, что ноль не может быть первой цифрой числа.

Напишите программу, которая поможет бухгалтеру.

Входной файл `input.txt` содержит два целых числа A и B ($-10^9 < A, B < 10^9$).

В выходной файл `output.txt` выведите одно целое число – наибольшую разность чисел, первое из которых может быть получено перестановкой цифр A , а второе – перестановкой цифр B .

str_24. Европейская комиссия планирует принять решение о том, что официальным языком Евросоюза станет английский. Был также разработан план упрощения английской письменности, который планируется реализовать за четыре года. Первоочередной задачей будет избавление от буквы *s*, которая в сочетаниях *si* и *se* будет изменяться на *s*, в сочетании *sk* – опускаться, а в остальных случаях заменяться на *k*. При этом все замены будут производиться в строгом порядке слева направо. То есть, например, в слове

success сначала первая из двух букв с заменится на k, а затем вторая – на s, то есть получится sukses. А слово cck превратится в kk.

На второй год из английских слов изымут все удвоенные буквы: ee изменят на i, oo – на u, а в остальных комбинациях будут просто писать одну букву вместо двух одинаковых. Такие замены также будут делать строго в порядке слева направо. Так, слово ooo превратится в uo, а oou – просто в u (в нем сначала oo заменится на u, а затем uu – на u), слово iee превратится в i (в нем сначала ee заменится на i, а затем ii – на i).

На третий год на конце слова станут опускать букву e, если она не единственная буква в слове.

Наконец, завершением реформы станет отмена артиклей (в английском языке три артикля: a, an и the). При этом удаляться эти артикли будут только тогда, когда они в исходном тексте были словами a, an, the. То есть, например, текст the table после реформ первых трех лет превратится в th tabl, а после реформы четвертого года – просто в tabl. А слово aaaaa после реформы первых лет станет словом a, но поскольку изначально оно не было словом a (артиклем), то оно в итоге так и останется словом a.

Требуется написать программу, которая будет переводить классический английский текст на Евроинглиш. Во входном файле *Input.txt* записана одна строка текста, состоящая не более чем из 200 символов: латинских строчных и заглавных букв, пробелов и знаков препинания (в тексте могут встречаться: точка, запятая, вопросительный и восклицательный знаки, двоеточие, тире, точка с запятой, открывающаяся и закрывающаяся скобки, апострофы, кавычки).

Заглавные буквы могут встречаться только в начале слова. Нигде подряд не могут стоять два пробела. В начале и конце строки не может стоять пробел. Слова отделяются друг от друга пробелами и/или знаками препинания.

В выходной файл *Output.txt* нужно выдать преобразованную строку при ограничениях:

- начинаться с заглавной буквы должны те и только те слова, которые начинались с заглавной буквы в исходном тексте;
- не должно встречаться двух пробелов подряд;
- пробелы между словами и знаками препинания должны остаться там и только там, где они были в исходной строке.

ке, в начале и конце строки пробелов быть не должно.

Пример

| Input.txt | Output.txt |
|--------------------------------------|-------------------------------|
| cacao and coffe | kakao and kofi |
| Cinderella! Where Is The Dress??? | Sinderela! Wher Is Dres??? |
| 'A' is a letter | " is letter |
| !!!Hello!!!A-the-"word" | !!!Helo!!!--"word" |
| Aaaa then the ckckck | A then k |
| "A"-the an | " "_ |
| A the an | |
| success | sukses |

8 Сортировки [s_4]

s_1.

Сортировка чисел в массиве – одна из самых базовых задач в программировании. Разные сортировки отличаются друг от друга требуемым количеством памяти, временем работы, сложностью реализации и некоторыми менее важными параметрами.

Периодически возникают задачи, в которых массив необходимо сортировать, используя некоторую нетривиальную операцию. Одну из таких задач Вам и предлагается решить. В качестве сортирующей операции Вы можете несколько раз использовать разворот последовательности из нескольких подряд идущих первых чисел массива. Например, Вы можете превратить массив (2, 3, 1, 4, 5) в массив (3, 2, 1, 4, 5), развернув первые два элемента, а затем превратить полученный массив в массив (1, 2, 3, 4, 5), развернув в первые три элемента.

Формат входного файла

В первой строке входного файла *input.txt* находится одно натуральное число $N (1 \leq N \leq 20)$ – количество элементов в массиве. В следующей строке перечислены сами элементы массива – натуральные числа, не превышающие 1000. Все элементы массива различны.

Формат выходного файла

В первой строке выходного файла *output.txt* требуется вывести одно целое число t , не превышающее 10000 – количество разворотов, которые Вы собираетесь сделать. В следующей строке необходимо через пробел вывести t натуральных чисел, не превышающих N , где очередное число является количеством первых элементов массива, которые Вы собираетесь развернуть соответствующей операцией. После выполнения выведенных Вами операций массив должен быть отсортирован по возрастанию.

Пример входных и выходных данных

| input.txt | output.txt |
|-----------|------------|
| 5 | 2 |
| 2 3 1 4 5 | 2 3 |

s_2.

Перестановкой из N элементов называется упорядоченный набор из N различных чисел от 1 до N . Найдите по заданной перестановке следующую в лексикографическом порядке.

Будем считать, что за перестановкой

$$(N, N - 1, \dots, 3, 2, 1)$$

следует тождественная перестановка, то есть

$$(1, 2, 3, \dots, N)$$

В первой строке входного файла *input.txt* содержится число N ($1 \leq N \leq 10^4$). Во второй строке содержится перестановка (последовательность натуральных чисел от 1 до N , разделенных пробелами).

Выходной файл *output.txt* должен содержать искомую перестановку.

Пример

| Input.txt | Output.txt |
|----------------|------------|
| 5 2 4 5 3 1 | 2 5 1 3 4 |
| 1 1 | 1 |

s_3.

В музее регистрируется в течение суток время прихода и ухода каждого посетителя. Таким образом, за день получены N пар значений, где первое значение в паре показывает время прихода посетителя и второе значение – время его ухода. Требуется найти максимальное число посетителей, которые находились в музее одновременно.

В первой строке входного файла *input.txt* записано натуральное число N ($N < 10^5$) – количество зафиксированных посетителей в музее в течение суток. Далее идут N строк с информацией о времени визитов посетителей: в каждой строке располагается отрезок времени посещения в формате "ЧЧ:ММ ЧЧ:ММ", ($00 : 00 \leq \text{ЧЧ:ММ} \leq 23 : 59$).

В единственную строку выходного файла *output.txt* одно целое число – максимальное количество посетителей, одновременно находящихся в музее.

Пример

| Input.txt | Output.txt |
|-------------|------------|
| 6 | 4 |
| 09:00 10:07 | |
| 10:20 11:35 | |
| 12:00 17:00 | |
| 11:00 11:30 | |
| 11:20 12:30 | |
| 11:30 18:15 | |

s_4.

В новом выпуске Большой Галактической Энциклопедии N страниц. Петя считает страницу счастливой, если произведение цифр, входящих в ее номер, равно K .

Например, если $N = 100$, то для $K = 42$ есть счастливая страница (например, с номером 76), а для $K = 128$ счастливой страницы нет.

Требуется написать программу, которая поможет Пете определить, есть ли счастливые страницы в новом выпуске энциклопедии.

Входной текстовый файл *input.txt* содержит числа N ($1 \leq N \leq 10^9$) и K ($1 \leq K \leq 10^9$), записанные через пробел.

Выходной текстовый файл *output.txt* должен содержать "YES", если счастливые страницы есть, и "NO" иначе.

9 Записи [z_5]

z_1.

Антон в школе начал изучать математику. Его внимание привлекло новое для него понятие числовой прямой. Антон быстро научился вычислять расстояния между двумя точками на этой числовой прямой, задавать отрезки и интервалы на ней. Готовясь к контрольной работе, Антон столкнулся со следующей задачей: "На числовой прямой задано N точек. Необходимо найти среди них две ближайшие". Расстояние между двумя точками числовой прямой x и y равно $|x - y|$. Первая строка входного файла `Input.txt` содержит количество точек N ($2 \leq N \leq 10^5$).

Вторая строка содержит N различных целых чисел – координаты точек на прямой. Числа в строке разделены пробелом. Значения всех координат не превосходят 10^9 по абсолютной величине.

В первой строке выходного файла `Output.txt` необходимо вывести минимальное расстояние между двумя точками, заданными во входном файле. Во второй строке выходного файла необходимо вывести номера точек, которым соответствует найденное расстояние. Точки нумеруются натуральными числами от 1 до N в том порядке, в котором они заданы во входном файле. Если ответов несколько, вывести тот из них, в котором точки расположены левее других на числовой прямой. Первым выводится номер левой точки, далее через пробел – номер правой точки.

| input.txt | output.txt |
|------------|------------|
| 5 | 1 |
| 10 3 6 2 5 | 4 2 |

z_2.

Всем известно, что дачники – народ странный, почти такой же, как и программисты. Строят они свои дачи непонятно где, да и выращаются там непонятно что и непонятно зачем. А уж как они туда добираются, это другая история: кто на автобусе, кто на электричке, кто на автомобиле, ну а кто-то вовсе пешком ходит от дома и до самого участка. Так что не стоит удивляться, если вдруг Вы узнаете, что некое садоводческое товарищество располагается на острове, а дачники добираются до него самолетом. Да еще и на этом острове может не быть посадочной полосы, так что высадиться на остров можно, только прыгая с парашютом (мы уж не рассматриваем то, как они возвращаются с дач домой). Рассмотрим этот уникальный случай. Пилот всегда старается осуществить

высадку парашютистов таким образом, чтобы дачники приземлялись как можно ближе к своим прямоугольным участкам. Пилоту интересно знать: сколько дачников приземлится на свои участки?

В первой строке входного файла `input.txt` записано натуральное число N ($1 \leq N \leq 1000$) – количество дачников, далее идут N строк, в каждой из которых описаны координаты каждого дачника и его участка: $X Y X1 Y1 X2 Y2 X3 Y3 X4 Y4$ где (X, Y) – координаты приземления парашютиста $(X1, Y1, X2, Y2, X3, Y3, X4, Y4)$ – координаты прямоугольного участка на плоскости, указанные последовательно. Все координаты – целые числа, не превышающие 30000 по абсолютной величине.

В выходной файл `OUTPUT.TXT` нужно вывести количество дачников, приземлившихся на свой участок.

ПРИМЕР

| Input.txt | Output.txt |
|---|------------|
| 3 6 6 3 6 6 9 8 7 5 4 13 5 9 2 9 8 12 8 12 2 3 2 2 1 2 3 6 3 6 1 | 2 |

z_3.

На автобусную остановку каждую минуту подходит автобус одного из маршрутов. Диспетчерская служба собрала данные за N минут – номера маршрутов каждого автобуса. Требуется определить максимально возможное время ожидания для пассажира, желающего уехать определенным маршрутом. Т. е. в данной последовательности номеров маршрутов нужно найти два самых удаленных числа, равных между собой.

Например, для последовательности 2, 11, 2, 2, 25, 11, 25, 11 максимальное время ожидания равно 4 (для маршрута номер 11).

Входной файл `Input.txt` содержит в первой строке число N ($2 \leq N \leq 10^6$). Во второй строке записаны N чисел – номера маршрутов. Все числа натуральные и не превышают 100. Каждый номер маршрута встречается не менее двух раз.

В выходной файл `Output.txt` вывести ответ на задачу.

ПРИМЕР

| Input.txt | Output.txt |
|---------------------------|------------|
| 8 2 11 2 2 25 11 25 11 | 4 |
| 4 23 23 41 41 | 1 |

z_4.

Иван Иванович любит ходить на скачки, надеясь на них заработать кругленькую сумму. Ему приглянулась лошадь с номером K , и он решил проверить, сможет ли она выиграть у всех остальных лошадей. Иван Иванович раздобыл информацию, в которой для некоторых пар лошадей сообщается, какая из этих лошадей быстрее. Также он узнал, что у всех лошадей разные скорости. Требуется написать программу, которая поможет Ивану Ивановичу точно определить может ли выиграть выбранная им лошадь. Входной файл `Input.txt` содержит в первой строке два целых числа N ($1 \leq N \leq 100$) и K ($1 \leq K \leq N$), где N – количество лошадей, принимающих участие в скачках, K – номер лошади, на которую хочет сделать ставку Иван Иванович. Следующие строки содержат по два числа X и Y ($1 \leq X, Y \leq N$), обозначающие, что лошадь с номером X быстрее лошади с номером Y . Пары X и Y не повторяются. Набор данных завершается строкой, содержащей единственный ноль. Эту строку обрабатывать не нужно. Гарантируется, что информация, раздобытая Иваном Ивановичем, корректна. Выходной файл `Output.txt` должен содержать слово "Yes", если Иван Иванович уверен в своем выигрыше и "No" в противном случае.

| Input.txt | Output.txt |
|------------------------|------------|
| 3 1 1 2 1 3 0 | Yes |
| 3 2 2 3 0 | No |
| 4 2 3 1 2 3 0 | No |

z_5.

Радар подвергается атаке из четырех точек, являющихся вершинами квадрата, в центре которого и стоит радар. Радар укомплектован специальным щитом, позволяющим блокировать удар, но щит может защищать радар только с одной из четырех сторон, и поворот щита требует времени. Изначально щит направлен в сторону той вершины, откуда будет первая атака.

Известно время запуска и скорость ракет, ведущих атаку. Требуется определить, сколько ракет удастся отбить.

Первые четыре строки входного файла `input.txt` содержат время

запуска в секундах T_x ($0 \leq T_x \leq 1000$) и скорость полета в метрах в секунду V_x x -ой ракеты ($0 < V_x \leq 1000$). Ракеты перечисляются по часовой стрелке. Далее задано время в секундах, необходимое для поворота щита на 90 градусов $T_{пов}$ ($0 < T_{пов} \leq 1000$) и половина диагонали квадрата D – расстояние в метрах, предстоящее каждой из ракет ($0 < D < 1000$). Все числа – целые.

В выходной файл Output.txt вывести "ALIVE", если радар уцелеет при всех выстрелах, в противном случае следует вывести число успешно отраженных ракет.

Примеры

| Input.txt | Output.txt |
|---|------------|
| 0 10 5 10 10 10 15 10 5 100 | ALIVE |
| 0 10 10 10 5 10 15 10 5 100 | 1 |

10 Динамическая память [din_4]

din_1. На прямой ветке железной дороги расположено несколько станций. Задана стоимость проезда между любыми двумя станциями. Требуется написать программу нахождения минимальной стоимости проезда между любыми двумя станциями. Двигаться по железной дороге можно только в одном направлении (от станции с меньшим номером к станции с большим).

Входной файл *input.txt* содержит в первой строке натуральное число N , не большее 250. Всего на дороге расположено $N + 1$ станций, пронумерованных от 0 до N . В следующих строках записано $N(N+1)/2$ чисел, задающих стоимости проезда между станциями: сначала стоимость проезда от станции 0 до станций 1, 2, 3, ..., N , затем от станции 1 до 2, 3, ..., N и т. д. от станции $N - 1$ до станции N . Все стоимости проезда – натуральные числа, не превосходящие 10000.

Выходной файл *output.txt* должен содержать одно число – минимальную стоимость проезда от станции 0 до станции N с возможными пересадками.

Пример

| Input.txt | Output.txt |
|-----------|------------|
| 3 | 12 |
| 7 10 20 | |
| 4 8 | |
| 2 | |

Пояснение: В приведенном примере всего 4 станции с номерами 0, 1, 2, 3. Оптимальный маршрут проходит через станции с номерами 0, 2 и 3. Его стоимость равна $10 + 2 = 12$.

din_2. Последовательность

$$a_1, a_2, a_3, \dots, a_{n-1}, a_n$$

называется пилообразной, если она удовлетворяет одному из следующих условий:

1) $a_1 < a_2 > a_3 < \dots > a_{n-1} < a_n$

2) $a_1 > a_2 < a_3 > \dots < a_{n-1} > a_n$

Дана числовая последовательность. Требуется определить длину самой длинной ее пилообразной подпоследовательности.

В первой строке входного файла *Input.txt* записано натуральное число N – количество элементов последовательности. Во второй строке файла через пробел записаны N элементов целочисленной последовательности. Ограничения: $N < 10^6$, $|a_i| < 32000$.

В единственную строку выходного файла *Output.txt* нужно вывести одно целое число – длину самой длинной непрерывной пилообразной подпоследовательности.

Пример

| Input.txt | Output.txt |
|-------------------------------|------------|
| 3 1 2 3 | 2 |
| 12 5 7 6 3 4 2 7 1 8 9 4 5 | 7 |
| 5 1 -2 3 -4 5 | 5 |

din_3. Петя, изучая, как меняется курс рубля по отношению к доллару и евро, вывел закон, по которому происходят эти изменения (или думает, что вывел). По этому закону Петя рассчитал, каков будет курс рубля по отношению к доллару и евро в ближайшие N дней. У Пети есть 100 рублей. В каждый из дней он может обменивать валюты друг на друга по текущему курсу без ограничения количества (при этом курс доллара по отношению к евро соответствует величине, которую можно получить, обменяв доллар на рубли, а потом эти рубли – на евро).

Поскольку Петя будет оперировать не с наличной валютой, а со счетом в банке, то он может совершать операции обмена с любым (в том числе и нецелым) количеством единиц любой валюты.

Напишите программу, которая вычисляет, какое наибольшее количество рублей сможет получить Петя к исходу N -го дня. Законы изменения курсов устроены так, что в течение указанного периода рублевый эквивалент той суммы, которая может оказаться у Пети, не превысит 10^8 рублей.

Первая строка входного файла *Input.txt* содержит одно число N ($1 \leq N \leq 5000$). В каждой из следующих N строк записано по 2 числа, вычисленных по Петиним законам для соответствующего дня – сколько рублей будет стоить 1 доллар, и сколько рублей будет стоить 1 евро. Все эти значения не меньше 0.01 и не больше 10000. Значения заданы точно и выражаются вещественными числами не более, чем с двумя знаками после десятичной точки.

В выходной файл *Output.txt* вывести искомую величину с двумя

знаками после десятичной точки.

| Input.txt | Output.txt |
|-----------|------------|
| 4 | 4000.00 |
| 1 10 | |
| 10 5.53 | |
| 5.53 1.25 | |
| 6 5 | |

din_4.

Рассмотрим компьютерную сеть с настроенной TCP/IP маршрутизацией. Будем рассматривать некоторую ее модификацию. А именно в этой сети находить N подсетей. Каждая подсеть характеризуется своей маской. Маска подсети представляет собой 4 однобайтных числа, разделенных точкой. Причем для масок выполнено следующее свойство:

если представить маску в двоичном виде, то сначала она будет содержать k единиц, а потом q нулей, причем $k+q = 32$. Например, 255.255.255.0 – маска подсети, а 192.168.0.1 – нет.

Поясним, как получается двоичное представление IP-адреса. Для этого числа, составляющие IP-адрес, представляются в двоичной системе счисления (при этом каждое из них дополняется ведущими нулями до длины в 8 цифр), после чего удаляются точки. Получившееся 32-битное число и есть двоичное представление IP-адреса. Например, для адреса 192.168.0.1 этот процесс выглядит так:

192.168.0.1 =>

11000000.10101000.00000000.00000001 =>

=> 1100000010101000000000000000000001.

Будем говорить, что два компьютера с IP_1 и IP_2 лежат в подсети, если $IP_1 \wedge Mask = IP_2 \wedge Mask$, где $Mask$ – маска этой подсети, а \wedge – операция побитового логического "И". IP компьютера представляет собой также 4 однобайтных числа, разделенных точкой.

В первой строке входного файла *input.txt* записано число N – количество подсетей. В следующих N строках перечислены маски этих подсетей. В $N+2$ строке находится число M ($0 \leq M \leq 10000$). В следующих M строках записаны пары IP-адресов, разделенных пробелом.

Для каждой пары IP-адресов в отдельной строке выходного файла *output.txt* вывести количество подсетей в которых лежат оба компьютера.

Пример

| Input.txt | Output.txt |
|---------------------------|------------|
| 2 | 1 |
| 255.255.255.255 | 1 |
| 255.255.255.0 | 0 |
| 3 | |
| 192.168.31.1 192.168.31.2 | |
| 192.168.31.3 192.168.31.4 | |
| 192.168.31.1 192.167.31.2 | |

11 Комбинаторика [cmb_1]

cmb_1. Петя играет в интересную игру. Для этой игры необходима монетка. Петя подбрасывает ее n раз и считает, сколько раз выпадает <решка>. Если <решка> выпадает хотя бы m раз, то Петя считает, что он выиграл игру. Однажды Петя задумался, какова вероятность того, что он выиграт игру. Для этого он хочет найти количество последовательностей результатов подбрасывания монетки, содержащих ровно n подбрасываний, при которых <решка> выпала хотя бы m раз.

Входной файл *input.txt* содержит два целых числа: n и m ($1 < n < 20$, $0 \leq m \leq n$).

В выходной файл *output.txt* вывести ответ на задачу.

12 for MSP [msp_2]

msp_1. for MSP, Вадим Калмыков, 22.09.08, 23:21

Для того чтобы проверить, как ее ученики умеют считать, Мария Ивановна каждый год задает им на дом одну и ту же задачу – для заданного натурального найти минимальное натуральное N такое, что N в степени N делится на A . От года к году меняется только число .

Во входном файле `Input.txt` содержится единственное число A ($1 \leq A \leq 10^9$). В выходной файл `Output.txt` необходимо вывести единственное число N .

Пример

| Input.txt | Output.txt |
|-----------|------------|
| 8 | 4 |
| 13 | 13 |

msp_2. Кубическое уравнение

(Время: 1 сек. Память: 16 Мб Сложность: 56%)

Написать программу, которая будет искать все целые X , удовлетворяющие уравнению

$$A \cdot X^3 + B \cdot X^2 + C \cdot X + D = 0,$$

где A, B, C, D – заданные целые коэффициенты.

Во входном файле `input.txt` записаны четыре целых числа: A, B, C, D . Все числа по модулю не превышают $2 \cdot 10^9$.

В выходной файл `output.txt` выведите сначала количество решений этого уравнения в целых числах, а затем сами корни в возрастающем порядке. Если уравнение имеет бесконечно много корней, то следует вывести в выходной файл одно число -1 (минус один).

Примеры

| | | |
|---|-----------|-----|
| 1 | 1 0 0 -27 | 1 3 |
| 2 | 0 1 2 3 | 0 |