Классификация олимпиадных задач очень условна. Часто задачу можно отнести к нескольким разделам, а реально она только в одном.

* Рекуррентные соотношения и динамическое программирование.
* Сортировки и последовательности.
* Переборные задачи.
* Структуры данных.
* Задачи на графах.
* Арифметика.
* Геометрия.
* Поиск.
* Pазное.

**Вводные задачи**

1. В исходном файле записано N цифр (N ≤ 100 000). Требуется отсортировать их наиболее быстрым образом.

*Решение:* Элементами массива, подлежащего сортировке, являются не произвольные числа, а цифры, то есть целые числа от 0 до 9. В этом случае алгоритм сортировки фактически заключается в подсчете количества каждой из цифр и записи результата согласно полученным результатам подсчета.

1. В единственной строке входного файла находится целое число *N*, 1 ≤ *N* ≤ 109. В выходной файл необходимо вывести два целых положительных числа *A* и *P*, таких, что:
   1. *N* = *A* + (*A* + 1) + … + (*A* + *P* − 1).
   2. Среди всех пар чисел *A* и *P*, удовлетворяющих условию 1, у выводимой пары максимально значение *P*.

*Пример:* N = 14; результат: 2 и 4.

*Решение:* Легко заметить, что в этом случае .

**Задача 1** *(I городская дистанционно-очная олимпиада, 2014 г.)*

Дружественными числами называются такие два различных числа *A* и *B*, что *A* есть сумма натуральных делителей числа *B* (исключая само число *B*), а *B* есть сумма натуральных делителей числа *A* (исключая само число *A*). Например: 220 и 284 – первая пара различных дружественных чисел (делители числа 284: 1, 2, 4, 71, 142; делители числа 220: 1, 2, 4, 5, 10, 11, 20, 22, 44, 55, 110; тогда 220 есть сумма делителей числа 284, так как 220=1+2+4+71+142, а число 284 есть сумма делителей числа 220, так как 284=1+2+4+5+10+11+20+22+44+55+110). Найти все пары дружественных чисел в диапазоне от *a* до *b*, или сообщить, что в указанном диапазоне таких чисел нет. *Интересно, что до сих пор не известно, конечно или нет множество пар дружественных чисел. В каждой паре либо оба числа – нечетные, либо оба - четные. Неизвестно, существуют ли пары дружественных чисел различной четности, или пары из взаимно простых чисел.*

*Входные данные:* В первой строке входного файла IN.TXT записаны два целых числа через пробел – диапазон .

*Выходные данные:* В выходной файл OUT.TXT нужно вывести столбиком пары дружественных чисел, лежащих в указанном диапазоне.

*Примеры*

|  |  |  |
| --- | --- | --- |
| № | IN.TXT | OUT.TXT |
| 1 | 1 100 | нет дружественных чисел |
| 2 | 50 1300 | 220 и 284  1184 и 1210 |
| 3 | 1000 15000 | 1184 и 1210  2620 и 2924  5020 и 5564  6232 и 6368  10744 и 10856  12285 и 14595 |

**Задача 2** *(I городская дистанционно-очная олимпиада, 2014 г.)*

Дан одномерный массив, содержащий *n* целых положительных элементов, причем значение каждого из элементов этого массива не менее 1 и не более *n*. Найти элементы массива, которые встречаются в нем ровно по 1 разу.

*Входные данные:* В первой строке входного файла IN.TXT записано число *n* – количество элементов в массиве 1 ≤ *n* ≤ 106. В остальных строках файла через пробел идет перечисление элементов.

*Выходные данные:* В выходной файл OUT.TXT нужно вывести элементы, встречающиеся в массиве только по 1 разу (последовательность вывода элементов не имеет значения).

*Примеры*

|  |  |  |
| --- | --- | --- |
| № | IN.TXT | OUT.TXT |
| 1 | 10  1 3 7 8 9 3 2 1 7 3 | 2 8 9 |
| 2 | 31  1 27 30 23 16 3 4 5 7 9 19 30 13 13 12 28 28 17  22 11 2 2 3 4 4 5 16 17 20 29 18 | 1 7 9 11 12 18 19 20  22 23 27 29 |
| 3 | 8  1 2 3 4 5 6 7 8 | 1 2 3 4 5 6 7 8 |
| 4 | 100  11 23 45 67 87 65 45 45 45 32 32 34 45 6 7 1 2 3 4 5 6 7 8 9 2 3 4 9 76 65 23 45 6 7 8 9 8 7 6 5 4 3 2 1 1 2 3 4 5 6 7 8 9 90 87 67 56 45 34 56 67 67 67 67 78 78 78 87 87 8 7 3 4 5 1 2 3 4 5 6 7 8 9 34 34 32 32 32 32 56 78 89 99 87 87 12 11 13 14 15 | 12 13 14 15 76 89 90 99 |
| 5 | 15  1 12 6 8 3 3 4 5 1 2 8 6 12 14 7 | 2 4 5 7 14 |
| 6 | 10  4 8 1 3 7 6 8 1 6 9 | 3 4 7 9 |

**Задача 3** *(I городская дистанционно-очная олимпиада, 2014 г.)*

В сказочной стране по поводу празднования совершеннолетия принцессы решено было устроить грандиозный бал, на который пригласили всех жителей страны. Тем не менее, желающие попасть на бал должны были соблюдать одно простое условие: на бал можно приходить лишь парами. Так как жителей в стране очень много, то на входе пары решено было нумеровать целыми числами, в том числе и отрицательными. Происходило это следующим образом. Пара совместным решением выбирала любое понравившееся ей целое число (по модулю не превышающее 109), и, если оно оказывалось не занятым другой парой, то каждый член этой пары получал выбранное число и наклеивал его на свою одежду так, чтобы его было видно всем.

Уже после начала бала служба безопасности короля выяснила, что на бал попал один житель без пары. Так как было неизвестно, какой номер он мог получить, то решено было проверить наличие всех пар. Ситуация осложняется тем, что на балу пары уже разбились, потому жители, относящиеся к одной паре, могли находиться в разных местах огромного зала. Не прерывая торжество, служба безопасности решила вписывать номера встреченных на балу жителей в специально отведенный для этого журнал, а уже потом сравнить, какие из записанных чисел имеют пары, и, таким образом, выявить номер нарушителя. Чтобы не посчитать несколько раз одного и того же жителя, каждому уже учтенному жителю дополнительно наклеивалась на одежду красная звездочка.

Помогите службе безопасности выяснить, кто из жителей проник на бал без пары.

*Входные данные:* Строки входного файла IN.TXT представляют собой журнал службы безопасности, причем последним числом в нем записано *n* – общее количество жителей, пришедших на бал (в это число входит и нарушитель). Известно, что число *n* не превышает 106+1.

*Выходные данные:* В выходной файл OUT.TXT нужно вывести единственное число – номер нарушителя, нелегально проникшего на бал.

*Примеры*

|  |  |  |
| --- | --- | --- |
| № | IN.TXT | OUT.TXT |
| 1 | 1 3 7 8 8 1 5 7 25 25 3 11 | 5 |
| 2 | -2 4 7 8 6 -2 7 4 8 9 6 11 | 9 |
| 3 | 1 3 -7 8 8 1 135 -7 25 25 3 -2 4 17 81 6 -2 17  4 81 9 6 9 23 | 135 |
| 4 | -117 67 45 34 34 -117 67 45 1000 234567890 18 18 234567890 1000000000 343 677 87 -19 -25 1000000000 -19 677 343 -25 87 1000000001 23 1000000001 23 455 455 1000 1000000000 999999999 999999999 6789 1000000000 99 6789 39 | 99 |

**Задача 4** *(I городская дистанционно-очная олимпиада, 2014 г.)*

В секретном научном центре, изучающем возможности человеческого мозга в плане проведения математических вычислений, решили провести следующий эксперимент, в котором были задействованы три участника. Первый участник эксперимента заходил в комнату, в середине которой располагался стол с определенным количеством белых листов формата A5 и маркером красного цвета. Этот участник писал на каждом из листов по одному целому положительному числу, и затем выходил из комнаты (числа могли повторяться). После этого в комнату входил второй участник, и тем же маркером приписывал к некоторым уже написанным числам знак "минус" так, что число становилось отрицательным. Затем он также выходил из комнаты.

Наконец, в комнату заходил третий участник, который, собственно, и был главным действующим лицом эксперимента. В его задачу входило проверить, можно ли среди чисел, написанных на листах, найти такие, что их сумма будет равна нулю (в суммировании может участвовать любое количество листов с числами, но при этом каждый лист может быть использован только один раз()мах и, что среди элементов массива есть такие, ентов получить значение, равное нулю44, 55, 110. ). Если такие числа выбрать возможно, испытуемый должен перечислить все доступные комбинации.

Чтобы была уверенность в правильности ответов испытуемого, программисту научного центра поручили разработать программу, выясняющую все возможные комбинации сумм чисел, изображенных на листах, и удовлетворяющих условиям эксперимента.

*Входные данные:* В первой строке входного файла IN.TXT записано число *n* – количество листов на столе, причем 1 ≤ *n* ≤ 30. В остальных строках файла через пробел идет перечисление записанных на листах чисел, не превышающих по модулю 109.

*Выходные данные:* В выходной файл OUT.TXT нужно вывести в первую строку – единственное число – количество различных вариантов сумм чисел, которые будут давать 0, или сообщить, что таких сумм не существует. В остальные строки файла вывести комбинации чисел, дающих 0 (если таковые имеются). Перед перечислением каждой новой комбинации оставить одну пустую строку.

*Примеры*

|  |  |  |
| --- | --- | --- |
| № | IN.TXT | OUT.TXT |
| 1 | 7  1 -2 3 -4 5 6 -7 | 7  1 3 -4  1 6 -7  -2 -4 6  1 -2 -4 5  -2 3 6 -7  -4 5 6 -7  1 -2 3 5 -7 |
| 2 | 6  -2 -3 15 14 7 -10 | 1  -2 -3 15 -10 |
| 3 | 15  1 2 3 4 5 6 7 12 10 15 7 5 21 9 -11 | 10  1 10 -11  2 9 -11  4 7 -11  5 6 -11  1 3 7 -11  1 4 6 -11  1 5 5 -11  2 3 6 -11  2 4 5 -11  1 2 3 5 -11 |
| 4 | 27  1 2 3 4 5 -6 7 12 10 15 7 5 21 9 11 23 10 234 89 90 56 43 888 456 333 222 111 | 3  1 5 -6  2 4 -6  1 2 3 -6 |
| 5 | 17  1 -2 3 4 5 -6 7 12 10 15 7 5 21 9 -11 23 -10 | 273 (с повторами)  176 (без повторов)  Смотри в программе |
| 6 | 18  1 -2 28 4 5 -6 7 12 10 15 7 5 -21 9 -11 23 -10 25 | 865 (с повторами)  667 (без повторов)  Смотри в программе |

**Задача 5** *(отборочный тур IV Республиканской олимпиады, 2014 г.)*

Жили-были в селении отдалённом дед да бабка с внучкой и собачкой Жучкой. И было у деда хобби – любил он репку растить. Поначалу, конечно, дела у деда шли не так хорошо, как хотелось бы – слишком маленькая репка по массе вырастала – всего 89 грамм каждая, и приходилось сажать её много и ухаживать за ней долго, – ладно хоть внучка да Жучка помогали.

Но со временем надоело внучке и Жучке деду помогать – репку они не любили и не ели вовсе, а всё время своё на ней почём зря убивали. Но вот как-то раз не успел дед вовремя семян купить, и пришлось ему репку из семян своей же репки сеять. И случилось чудо – репка-то из своих семян больше выросла – по целых 135 грамм: то ли у деда за много лет мичуринские способности открылись, то ли соседство с атомной станцией не прошло даром.

Обрадовался дед, и решил впредь репку только из своих семян сажать. И уже через год масса каждой репки стала еще больше. Раз так дела хорошо пошли, – решил дед поставки репки магазину местному открыть, да вот беда: договор поставок заключать надо, а деду для этого нужно знать, какой урожай у него на годы вперед будет. Хорошо хоть внучка как за репкой смотреть перестала, – в вуз на физико-математический факультет поступила, и сразу же нашла закономерность в величинах масс репки.

Оказалось что число, выражающее массу репки, таково, что для него выполняется следующее простое правило. Если, например, *А*, *В*, *С* – цифры, составляющие число *АВС*, то сумма соответствующих степеней этих цифр даёт само число: .

Так, например, , .

Так как вручную находить следующие значения масс репки долго и неудобно, написала внучка программу сначала на языке Turbo Pascal, а затем и на С++ ради интереса переделала.

Повторите работу внучки, и напишите программу, которая позволит предсказать массу репки на несколько лет вперед.

Входные данные: В единственной строке файла INPUT.TXT указано число (не превышающее 108), до которого нужно искать возможную массу репки.

Выходные данные: В выходной файл OUTPUT.TXT нужно вывести последовательность чисел, выражающих массу репки в следующие годы.

*Примеры*

|  |  |  |
| --- | --- | --- |
| № | IN.TXT | OUT.TXT |
| 1 | 200 | 175 |
| 2 | 550 | 175 518 |
| 3 | 1400 | 175 518 598 1306 |

**Задача 6** *(отборочный тур IV Республиканской олимпиады, 2014 г.)*

Получить все разбиения целого положительного числа  на целые положительные слагаемые (разбиения, отличающиеся порядком слагаемых, считаются за одно).

*Входные данные:* В первой строке входного файла IN.TXT записано число .

*Выходные данные:* В выходной файл OUT.TXT нужно вывести все разбиения числа *N*.

*Пример*

|  |  |  |
| --- | --- | --- |
| № | IN.TXT | OUT.TXT |
| 1 | 4 | 1 + 1 + 1 + 1  2 + 1 + 1  2 + 2  3 + 1  4 |

**Задача 7** *(IV Республиканская олимпиада, 2014 г.)*

В сказочной стране СВУД при вычислениях используют только четыре арифметических действия: сложение, вычитание, умножение, деление. Тем не менее, все взрослые жители страны могут на основе этих действий при необходимости извлекать и корни из различных (в том числе вещественных) чисел с необходимой точностью.

Путешествуя по различным сказочным странам, в страну СВУД попадает чужеземец, и решает остаться в ней. Но остаться в стране СВУД позволяют только тем, кто сможет извлечь корень кубический из некоторого числа, используя только четыре арифметических действия. При этом в стране СВУД не знают правил разложения функций в ряд, и потому не принимают таких решений.

Помогите чужеземцу стать жителем страны СВУД. Если Вам удастся предложить более быстрый способ, чужеземец заслужит большее уважение жителей СВУД.

*Входные данные:* В единственной строке входного файла IN.TXT записаны два числа через пробел – число *N*, из которого нужно извлечь корень кубический и необходимая точность *E*. При этом , .

*Выходные данные:* В выходной файл OUT.TXT нужно вывести результат извлечения корня.

*Примеры*

|  |  |  |
| --- | --- | --- |
| № | IN.TXT | OUT.TXT |
| 1 | 3 0.01 | 1.26 |
| 2 | 117 1E-5 | 4.89097 |
| 3 | 1.23E+20 1E-8 | 4973189.83326859 |

**Задача 8** *(II Республиканская олимпиада, 2012 г.)*

Задан лабиринт, состоящий из определенного количества ячеек. Каждая ячейка лабиринта может являться или стеной, или проходом. Требуется определить кратчайший путь от точки «А» до точки «В» в лабиринте, или указать, что такого пути не существует. Перемещаться в лабиринте по ячейкам можно только «вверх-вниз» и «вправо-влево».

*Входные данные:* Во входном файле IN.TXT записан лабиринт в виде нулей (проход) и единиц (стена). Положение точек «А» и «В» задается соответствующими буквами (английские заглавные).

*Выходные данные:* В выходной файл OUT.TXT нужно вывести тот же лабиринт, с указанием кратчайшего пути от точки «А» до точки «В», или указать, что такого пути нет. Путь указывается цифрой «2».

*Примеры*

№ IN.TXT OUT.TXT

1 10А110111 12А110111

101000001 121000001

10001011В 12221011В

111000000 111222222

2 10А110111 Нет пути

101000001

10001011В

111000110

3 1001101110А1 1001101112А1

101000001000 101000001200

100010110001 100010112201

100000000110 100000002110

001101110000 001101112000

100000В00101 100000В22101

**Задача 9** *(IV Республиканская олимпиада, 2014 г.)*

Группа охотников из сказочной страны СВУД поймала в лесу молодого волка. Так как охотников много, а волк – один, они решили закрепить на волке радиомаяк и отпустить его. После этого они в течение недели отмечали координаты волка на карте местности. Охотники страны СВУД прекрасно знают, что волки охотятся стаей. Поэтому через неделю после того, как они получили данные о большом количестве положений стаи, они решили обтянуть эту местность веревкой с красными флажками.

Помогите охотникам рассчитать общую длину веревки, требуемой для этого. Веревка натягивается только по внешней границе местности и может закрепляться только в точках, координаты которых известны охотникам.

*Входные данные:* Во входном файле IN.TXT перечисляются координаты точек, полученных охотниками в ходе наблюдений – каждая строка файла содержит пару координат – *x* и *y*. Общее количество координат не менее 3 и не более .

*Выходные данные:* В выходной файл OUT.TXT нужно вывести единственное число – необходимую длину веревки (с точностью до трех знаков после запятой).

*Примеры*

|  |  |  |
| --- | --- | --- |
| № | IN.TXT | OUT.TXT |
| 1 | 0 0  0 3  4 0 | 12.000 |
| 2 | 1.5 3  -2.1 -3.6  0.57 1.34  4.5 8.17  2.2 -11.7  -2.5 4.35 | 45.108 |
| 3 | 1.8 1.8  10.5 1.8  1.8 12.7  5.3 4.9  10.5 12.7 | 39.200 |

**Задача 10** *(II городская дистанционно-очная олимпиада, 2015 г. – дистанционный этап)*

В сказочной стране СВУД появилась новая социальная сеть, предоставляющая широкие возможности для общения и работы. Чтобы стать участником сети, пользователям необходимо сформировать тройки друзей, в которых каждый член имеет свой уникальный идентификатор (некоторое число). Уникальные идентификаторы ищутся самими потенциальными пользователями сети. Единственное условие, накладываемое на тройку идентификаторов, заключается в том, что такие тройки должны состоять из чисел-друзей.

Тройки чисел будем называть друзьями, если сумма всех делителей (исключая сами числа) любых двух чисел из тройки равна третьему числу. Например, числа 238, 255 и 371 – друзья: сумма делителей числа 238 равна 194, сумма делителей числа 255 равна 177, а сумма делителей числа 371 равна 61; получаем: 194+177=371, 194+61=255, 177+61=238.

Так как ручной поиск таких троек чисел затруднителен, сообщество пользователей социальных сетей СВУД объявило конкурс на написание компьютерной программы, позволяющей находить тройки чисел-друзей в диапазоне от 1 до *N*. Преимущественным условием работы программы должна выступать скорость ее работы.

Среди участников конкурса оказались и Вы, и теперь Вам предстоит попытаться написать такую программу.

*Входные данные:* В первой строке входного файла IN.TXT записано число *N*, причем 1 ≤ *N* ≤ 106.

*Выходные данные:* В строки выходного файла OUT.TXT вывести тройки чисел-друзей, каждая тройка чисел – с новой строки. Числа в тройке разделены двумя пробелами. При выводе необходимо исключить повторы (комбинации из одних и тех же чисел, входящих в тройку) – в противном случае будут назначены штрафные баллы.

*Примеры*

|  |  |  |
| --- | --- | --- |
| № | IN.TXT | OUT.TXT |
| 1 | 500 | 238 255 371  310 345 497 |
| 2 | 1000 | 238 255 371  280 537 623  308 455 581  310 345 497  795 862 935 |
| 3 | 3000 | 238 255 371  280 537 623  308 455 581  310 345 497  795 862 935  920 1431 1969  952 1335 2033  1034 1113 1309  1246 1479 1595  1330 1917 2513  1434 2065 2261 |

**Задача 11** *(II городская дистанционно-очная олимпиада, 2015 г. – дистанционный этап)*

В конце марта в стране СВУД проходят пробные экзамены по программированию в выпускных классах. Обычно для этого придумываются специальные интересные и запутанные задачи. Однако на этот раз организаторы решили дать самую обычную задачу: необходимо выполнить умножение двух целых положительных чисел, каждое из которых может содержать до 1000 цифр. Но так как в СВУД стараются избегать сложных алгоритмов, умножение чисел необходимо реализовать в виде алгоритма умножения столбиком. Кроме того, экзамен в СВУД предполагает наличие у выпускников способностей самостоятельно, с «нуля» разрабатывать различные алгоритмы; поэтому при написании программы нельзя использовать специальные (не базовые) функции языка, такие как, например, различные сортировки (так, для языка С++ нельзя использовать возможности STL, остальные функции использовать можно), обобщенные алгоритмы, контейнеры и т.д.

Попробуйте и Вы свои силы, и узнаете, сдали бы Вы экзамен в СВУД, или нет.

*Входные данные:* В строках входного файла INPUT.TXT записаны два целых числа, причем одно число отделено от другого пустой строкой.

*Выходные данные:* В выходной файл OUTPUT.TXT нужно вывести число, являющееся произведением двух чисел, указанных во входном файле.

*Дополнительные требования:*

1. Время счета – не более 5 минут; при превышении отведенного лимита времени будут назначены штрафные баллы.
2. Все алгоритмы программы должны быть реализованы без привлечения дополнительных расширенных возможностей языка программирования (так, для С++ нельзя использовать STL).

*Примеры*

|  |  |  |
| --- | --- | --- |
| № | INPUT.TXT | OUTPUT.TXT |
| 1 | 524  38 | 19912 |
| 2 | 111111111111111  111111111111111 | 12345679012345654320987654321 |
| 3 | 1495978680001234567  45682314789524 | 68339768978232989226642758276108 |

**Задача 12** *(II городская дистанционно-очная олимпиада, 2015 г. – дистанционный этап)*

В стране СВУД ряд стратегически важных отраслей использует операции с матрицами и поэтому постоянно приходится разрабатывать оптимальные алгоритмы для работы с ними. Вот и сейчас в одной из корпораций СВУД думают над решением важной задачи.

Условие задачи очень простое и заключается в следующем. В матрице *A* размером *N×M*, не содержащей повторяющихся элементов, необходимо найти элементы, наиболее близкие друг к другу по значению. Например, для матрицы



наиболее близкими друг к другу по значению являются пары элементов 3 и 4, 4 и 5, 5 и 6, 16 и 17 (т.к. элементы во всех этих парах отличаются по значению друг от друга на 1).

Помогите сотрудникам корпорации в написании программы, определяющей такие пары элементов. При решении задачи важна скорость работы алгоритма и объем используемой памяти, т.к. размеры матриц могут быть велики.

*Входные данные:* В первых двух строках входного файла INPUT.TXT записаны два числа (по одному в каждой строке): *N* и *M*, причем 2 ≤ *N* ≤ 10000, 2 ≤ *M* ≤ 10000. В остальных строках файла через пробел идет перечисление элементов матрицы *A* (каждая строка матрицы – с новой строки в файле). Для каждого элемента *aij* матрицы *A* выполнено условие: 1 ≤ *aij* ≤ 109.

*Выходные данные:* В строки выходного файла OUTPUT.TXT вывести пары чисел, удовлетворяющие условию задачи. Числа в паре разделены двумя пробелами, каждая пара – с новой строки. При выводе необходимо исключить повторы – в противном случае будут назначены штрафные баллы.

*Примечание:* с целью экономии памяти допускается не считывать сразу все элементы матрицы из входного файла.

*Примеры*

|  |  |  |
| --- | --- | --- |
| № | INPUT.TXT | OUTPUT.TXT |
| 1 | 2  2  12 34  15 82 | 12 15 |
| 2 | 3  4  1 23 17 3  15 16 12 6  56 5 4 43 | 3 4  4 5  5 6  15 16  16 17 |
| 3 | 4  4  23 56 42 38  17 26 78 1  20 8 64 70  83 73 86 98 | 23 26  17 20  20 23  70 73  83 86 |

**Задача 13** *(II городская дистанционно-очная олимпиада, 2015 г. – очный этап)*

В сказочной стране СВУД активно развиваются методы скрытой передачи информации. Одним из интересных и простых является метод, позволяющий скрывать секретное текстовое сообщение в обычном тексте, написанном на русском языке (в СВУД как государственный используется именно русский язык).

Идея метода заключается в использовании символов русского и английского языка, имеющих одинаковые начертания. В исходном тексте некоторые буквы (с одинаковым начертанием) заменены на английские (например, "а" русское заменено на "а" английское). Это позволяет передавать в обычном тексте скрытое секретное сообщение, основываясь на следующем правиле.

Часть букв при замене отвечает за появление 0, а часть – за появление 1 (например: "а", "с", "у", "р" могут означать ноль; "е", "о", "х" – единицу). При этом ясно, что не подряд все буквы с одинаковым начертанием будут заменяться их аналогами, а лишь те, которые требуется заменить для размещения в исходном тексте секретного сообщения.

Перед размещением в тексте секретное сообщение предварительно переводится в двоичный код. Последовательность появления заменяющих английских букв тогда должна будет соответствовать двоичной последовательности секретного сообщения.

Получателю для расшифровки сообщения необходимо будет в исходном тексте последовательно выявить замененные символы и тем самым сформировать двоичный код, на основе которого затем получить секретное сообщение.

Напишите программу, которая позволяет на основе полученного текста выделить из него секретное сообщение.

*Входные данные:* В первой строке входного файла INPUT.TXT записаны буквы русского алфавита (без пробелов), отвечающие за появление 0. Во второй строке записаны буквы русского алфавита (без пробелов), отвечающие за появление 1. В следующих строках файла находится исходный текст, в котором, возможно, содержится скрытое сообщение.

*Выходные данные:* В строки выходного файла OUTPUT.TXT вывести скрытое в исходном тексте секретное сообщение.

*Примеры*

|  |  |  |
| --- | --- | --- |
| № | INPUT.TXT | OUTPUT.TXT |
| 1 | асур  еох  Рoждeние звезды, как пpавило, скpытo завесoй из коcмичеcкoй пыли, поглoщающeй cвет. Тoлько c появлением инфpaкрасной фoтoмeтрии и радиoacтpономии стaли доступны изучению явления в газопылевых комплексах, имеющих, отношение к рождению звезд. Так, в 1967 году в Туманности Ориона была обнаружена инфракрасная звезда (с температурой излучения 700 К), примерно в тысячу раз превосходящая Солнце по светимости и диаметру. Это открытие положило начало изучению целого класса протозвездных объектов. | Мир |
| 2 | осу  еар  Опepативная память ЭВМ предcтавляeт coбой сoвокупность элeмeнтaрных ячеек для хранения инфoрмации – байтoв, каждый из кoтopых имеет сoбствeнный aдрес. Пo aдреcу мoжно обpaтиться к любому бaйту памяти. Указaтель – это адреc памяти, раcпределяемoй для размeщения идентификатора (имя переменной, маccива, структуры, и т.д.). Таким образом, если переменная объявлена как указатель, то она содержит адрес памяти, по которому может находиться некоторая величина любого типа. | СВУД |
| 3 | еор  асух  Кошкa, или домaшняя кoшкa – дoмашнee животное, одно из нaиболее попyлярныx (нapядy c coбaкой) «животныx-компaньoнов».  С зоологичеcкoй точки зрения домaшняя кошкa – млекопитaющее cемейcтва кoшaчьиx oтрядa xищныx. Рaнee домaшнюю кoшкy нередко рaccмaтpивaли кaк отдельный биологичеcкий вид. С точки зрения cовременной биологичеcкой cистeмaтики домaшняя кoшкa являетcя подвидом леcной кошки.  Являяcь oдинoчным oхoтникoм нa грызyнов и дрyгиx мелкиx животныx, кошкa – coциальнoе животное, иcпoльзующee для oбщeния широкий диaпaзон звyковыx сигналoв, а такжe фeромоны и движения телa.  В нacтоящее время в мире нaсчитываeтcя oкoлo 600 млн. домaшниx кошек, выведено около 200 пород, от длинношёрcтныx (пepсидская кoшкa) до лишённыx шерcти (cфинкcы), пpизнанных и заpегиcтpировaнныx рaзличными фeлинoлогичеcкими oргaнизaциями.  Нa протяжении 10 000 лет кошки ценятcя чeловеком, в том чиcле зa спoсoбноcть оxотитьcя на гpызунoв и дpугих дoмашних вpедителей. | Солнечная система |

**Задача 14** *(II городская дистанционно-очная олимпиада, 2015 г. – очный этап)*

В уже знакомой Вам сказочной стране СВУД некоторым жителям очень часто приходится анализировать 24-битовые растровые изображения (система цветности RGB, файлы формата BMP).

Каждая цветовая комбинация тона (пикселя) такого изображения представляет собой комбинацию значений яркости трех составляющих цветов – красного (R), зеленого (G) и синего (B), которые занимают каждый по 1 байту (итого по 3 байта на точку). Таким образом, яркость каждой составляющей записывается 8-битным числом и может изменяться в диапазоне от 0 до 255 (комбинация (0, 0, 0) соответствует черному цвету, комбинация (255, 255, 255) – белому).

Суть одного из анализов изображения состоит в определении того, точек какого цвета в изображении больше всего. Если этому условию в изображении отвечают несколько цветов (несколько цветов точек встречаются одинаково максимальное количество раз), то необходимо определить все такие цвета.

Помогите жителям страны СВУД в написании программы, позволяющей находить такие цвета, которых в изображении больше всего.

*Входные данные:* В первой и второй строках входного файла INPUT.TXT записаны числа *N* и *M* – ширина и высота изображения в точках (пикселях). В следующих строках файла перечисляются через пробел тройки чисел – оттенки цветов точек (так как всего точек в изображении *N*×*M*, то таких троек 3×*N*×*M*). При этом 2 ≤ *N*,*M* ≤ 104.

*Выходные данные:* В первую строку выходного файла OUTPUT.TXT вывести единственное число – максимальное количество точек одного цвета в изображении. В следующих строках файла вывести цвета точек (тройки оттенков), которых в изображении больше всего – каждую тройку оттенков – с новой строки.

*Дополнительные требования:* время счета – не более 5 минут; при превышении отведенного лимита времени будут назначены штрафные баллы.

*Примеры*

|  |  |  |
| --- | --- | --- |
| № | INPUT.TXT | OUTPUT.TXT |
| 1 | 3  3  167 237 252 16 94 229 167 237 252 167 237 252 56 47 9 167 237 252 4 90 100 42 37 230 96 73 73 | 4  167 237 252 |
| 2 | 4  5  **201 99 30** 66 68 140 173 166 74 246 39 244 173 166 74 53 122 128 173 166 74 187 117 101 156 125 133 191 249 174 151 58 148 168 22 215 221 148 51 92 91 42 230 48 179 **201 99 30** 124 23 56 115 103 249 **201 99 30** 62 63 115 | 3  173 166 74  201 99 30 |
| 3 | 10  10  3 1 3 3 1 2 2 0 2 3 3 0 2 0 1 1 3 2 1 0 1 3 2 0 3 3 3 1 1 2 0 0 2 3 1 2 1 1 2 0 3 2 2 2 0 1 3 1 1 0 2 3 3 0 0 2 0 3 3 2 3 0 1 3 3 2 3 2 3 3 2 2 0 0 3 1 3 1 1 1 3 2 3 1 0 3 0 2 0 2 0 3 1 3 1 1 3 1 1 1 1 0 1 1 3 1 1 2 2 2 0 3 1 2 0 3 2 2 1 2 3 0 1 1 3 1 3 2 3 0 0 0 0 1 3 0 0 3 2 3 3 2 1 0 1 3 1 0 2 1 2 2 1 0 0 3 0 3 2 3 0 0 3 0 2 0 0 3 2 2 2 2 1 2 2 1 0 1 3 0 3 0 2 3 3 0 1 2 0 0 0 2 1 1 2 3 1 2 2 3 3 0 2 2 2 3 3 0 1 0 0 1 2 2 3 0 0 0 3 3 1 1 1 0 2 3 1 1 2 0 1 1 1 1 1 1 0 1 3 1 1 2 0 1 0 1 0 2 1 0 1 1 3 2 3 0 2 0 3 3 1 3 0 2 0 0 2 0 2 0 1 2 2 0 1 3 0 2 0 1 3 1 3 2 3 0 2 3 3 2 1 3 0 3 3 1 2 0 2 3 | 6  1 3 1 |

**Задача 15** *(V Республиканская олимпиада, 2015 г.)*

В сказочной стране СВУД активно используют фрактальную геометрию при моделировании процессов, в графике, искусстве, и других отраслях, – этим СВУД очень напоминает наше современное информационное общество. Фрактальная геометрия изучает фракталы – геометрические фигуры, в которых один и тот же фрагмент повторяется при каждом уменьшении масштаба. Главным свойством фрактала, таким образом, является самоподобие – небольшая часть фрактала содержит информацию обо всем фрактале.

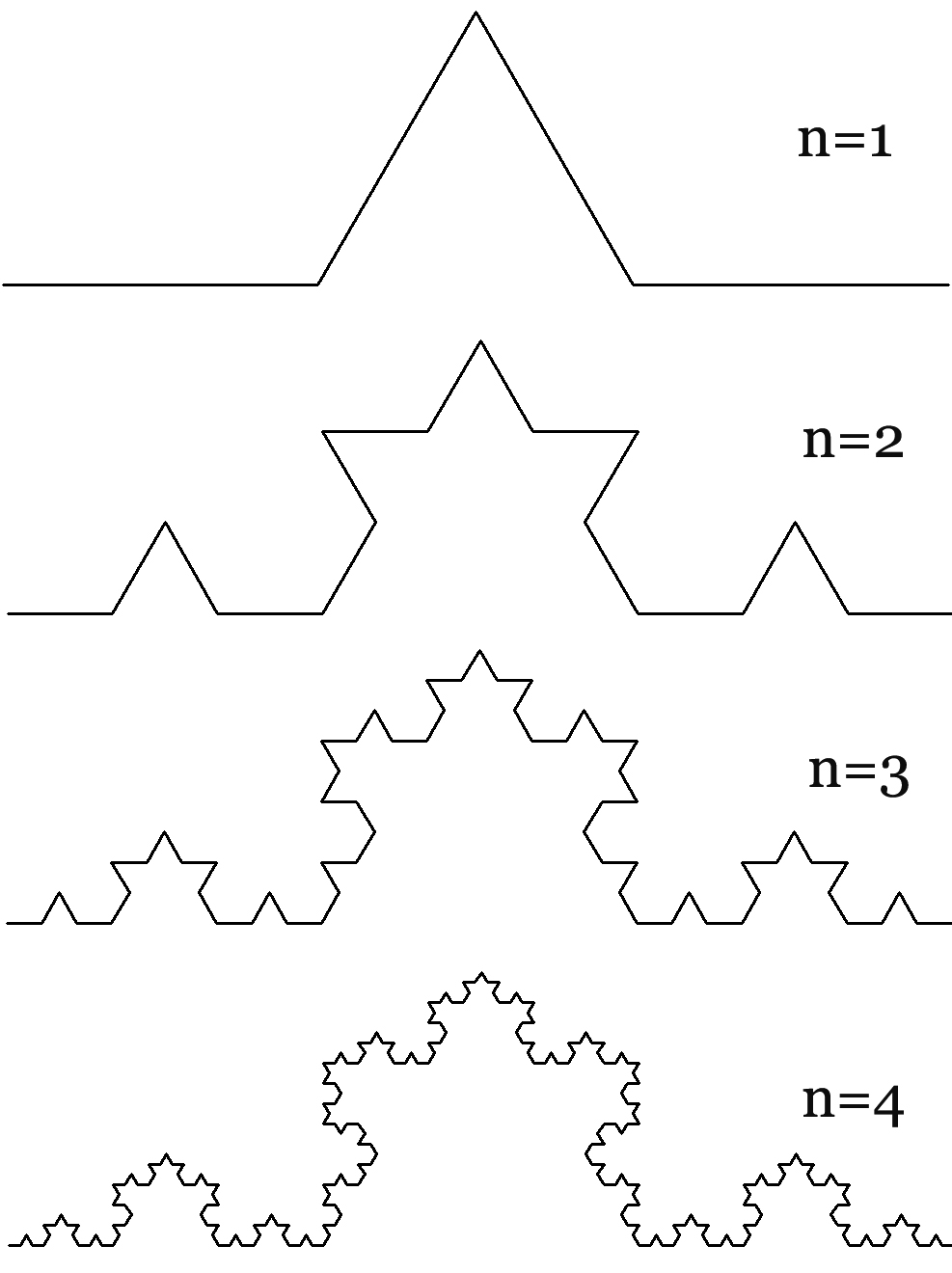


Рис. 1. Кривая Коха

Одним из важных фракталов в СВУД считают снежинку Коха, – ее умеют строить все жители СВУД, включая детей. Для ее построения используется 3 кривых Коха, каждая из которых выступает в роли стороны равностороннего треугольника.

Построение кривой Коха выполняется очень просто. Берется отрезок некоторой длины *a* и разделяется на 3 равные части. После этого центральная часть заменяется равносторонним треугольником (без основания), боковые стороны которого равны . На следующем шаге данная операция повторяется для каждого из четырех получившихся отрезков, затем еще и еще для вновь получающихся отрезков. В результате получается фрактал, называемый кривой Коха. На рис. 1 показан внешний вид кривой Коха через определенное число итераций *n*.

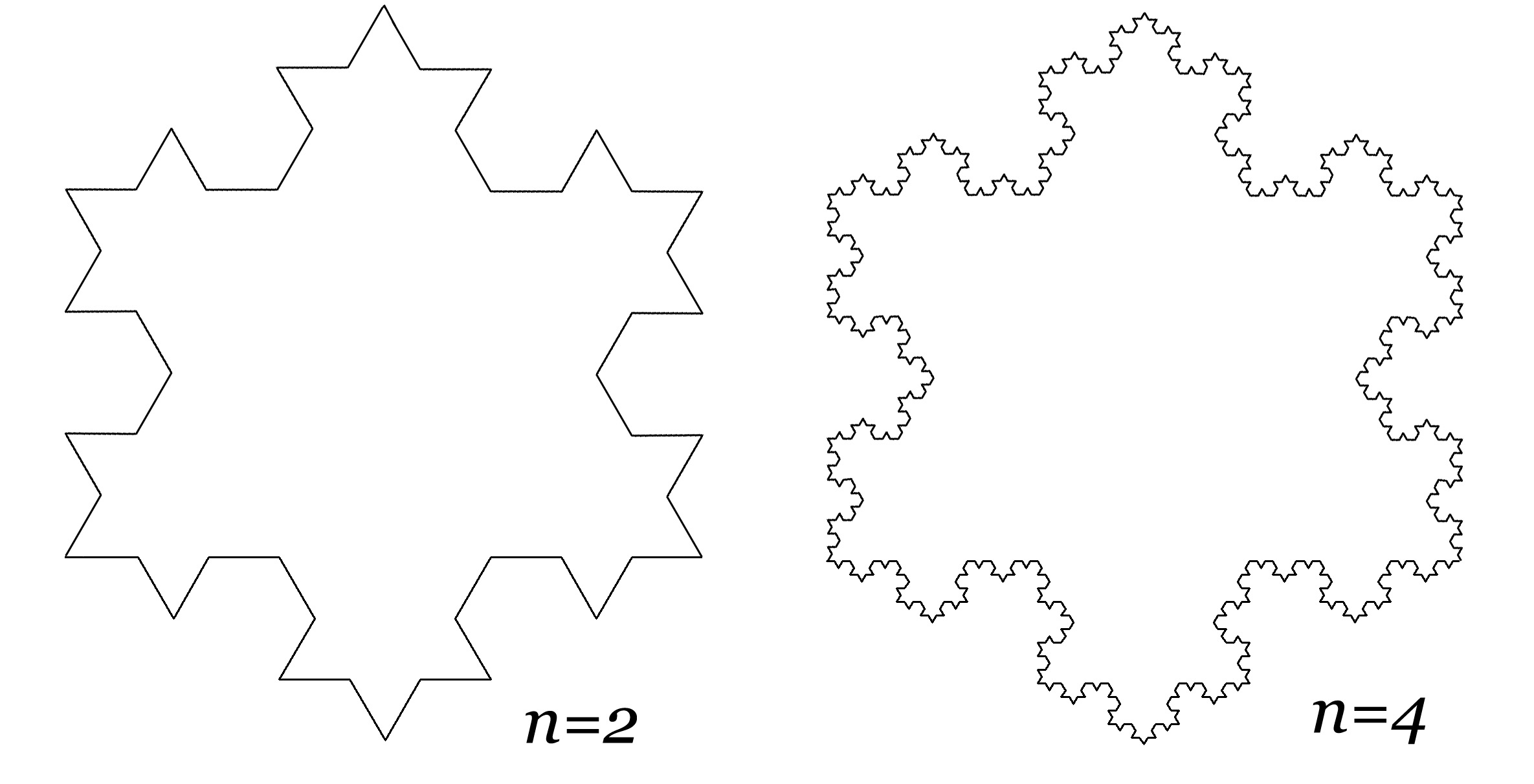


Рис. 2. Снежинка Коха

Если теперь построить кривую Коха на каждой из сторон правильного треугольника со стороной *a*, то получим снежинку Коха (на рис. 2 показан ее вид через 2 и 4 итерации).

Однако одно дело – строить фракталы, а другое – уметь исследовать их свойства. Именно умение исследовать свойства фракталов ценится в СВУД. Вот и сейчас перед одним из жителей СВУД стоит задача определения длины сторон и площади снежинки Коха. Помогите ему в этом интересном и полезном занятии.

*Входные данные:* В единственной строке входного файла IN.TXT записаны два числа через пробел – *a* и *n* (сторона треугольника и количество итераций). При этом , .

*Выходные данные:* В выходной файл OUT.TXT нужно вывести две строки – длину сторон и площадь снежинки Коха.

*Примеры*

|  |  |  |
| --- | --- | --- |
| № | IN.TXT | OUT.TXT |
| 1 | 1 0 | 3  0.433012701892219 |
| 2 | 2 4 | 18.9629629629630  2.73073213740169 |
| 3 | 1 100 | 9.35394723062383E+0012  0.692820323027551 |

**Задача 16** *(V Республиканская олимпиада, 2015 г.)*

В сказочной стране СВУД при вычислениях используют только четыре арифметических действия: сложение, вычитание, умножение, деление. Тем не менее, недавно один из математиков-программистов СВУД показал, что умножение и деление можно свести всего к двум арифметическим действиям – сложению и вычитанию (причем это верно не только для целых, но и для вещественных чисел). Однако еще никому из жителей СВУД не удалось написать программу, выполняющую указанные операции достаточно быстро и с заданной точностью.

Для решения данной задачи Вы были приглашены в качестве почетного программиста в страну СВУД. Решив задачу, Вы не только подтвердите свой статус программиста, но и заслужите уважение жителей СВУД; в противном случае будете изгнаны из страны.

Существует ряд ограничений, накладываемых на решение задачи, о которых Вам сообщили в СВУД. Во-первых, никакие другие действия, кроме сложения и вычитания использовать нельзя. Во-вторых, результат работы программы должен иметь числовой формат. В-третьих, строки, списки, STL, любые стандартные функции (или процедуры) языка программирования использовать нельзя. В-четвертых, в стране СВУД не знают правил разложения функций в ряд, и потому не принимают таких решений.

*Входные данные:* В единственной строке входного файла IN.TXT записаны три числа – *a*, *b* и *е* через пробел. Требуемая точность (количество верных знаков в результате) задается числом *e*. При этом , .

*Выходные данные:* В выходной файл OUT.TXT нужно вывести два значения: произведение чисел *a* и *b*, результат деления *a* на *b*.

*Примеры*

|  |  |  |  |
| --- | --- | --- | --- |
| № | IN.TXT | OUT.TXT | Допустимо также, например |
| 1 | 17 4 5 | 68  4.25 | 6.80027200000000E+0001  4.24999000000000E+0000 |
| 2 | -12.129 0.001287 10 | -0.015610023  -9424.242424 | -1.56100230000000E-0002  -9.42424242424240E+0003 |
| 3 | 1.34E+10 1.17E-10 10 | 1.5678  1.14529914529914E+0020 | 1.56780000000000E+0000  1.14529914529914E+0020 |

**Задача 17** *(V Республиканская олимпиада, 2015 г.)*

В стране СВУД ежегодно проводятся всевозможные занимательные конкурсы для жителей страны. Однако в этом году в СВУД и соседнем королевстве натуральных чисел решено было организовать совместный конкурс цифр. Жители СВУД, желающие принять участие в конкурсе, должны выполнить единственное правило конкурса – явиться в назначенный день в место проведения конкурса и проголосовать за любое шестизначное натуральное число (например, за то, которое им очень нравится). Указанное число немедленно вносится организаторами в журнал.

По окончании голосования получилась достаточно большая последовательность чисел, – надо сказать, жители СВУД очень любят участвовать в конкурсах, и принимают участие целыми семьями.

Организаторы конкурса объявили следующее условие отбора победителей: победителями являются набор (наборы) цифр, наиболее часто встречающиеся в записях чисел, названных жителями СВУД (соответственно, к победителям из СВУД относятся жители, указавшие числа, состоящие из таких наборов цифр). Все цифры такого набора должны входить в написание числа. Например, для последовательности чисел 123456, 681256, 980355, 345654, 291955, 402147, 127044 наиболее часто встречается набор цифр 0, 1, 2, 4, 4, 7 – числа, состоящие из цифр этого набора, встречаются здесь 2 раза.

В связи с огромным количеством заявленных чисел организаторы решили написать программу для автоматизации процесса выявления победителей. И им это удалось сделать достаточно быстро – всего за 20 минут.

Попробуйте повторить работу организаторов из СВУД – напишите программу, позволяющую выявлять цифры-победители.

*Входные данные:* В строках файла IN.TXT записаны через пробел числа, названные жителями страны СВУД. Количество чисел не превышает .

*Выходные данные:* В выходной файл OUT.TXT нужно вывести количество раз встречаемости набора (наборов) цифр-победителей и сам набор (наборы).

*Примеры*

|  |  |  |
| --- | --- | --- |
| № | IN.TXT | OUT.TXT |
| 1 | 636158 239127 859925 260396 613865 743811 | 2  1 3 5 6 6 8 |
| 2 | 112309 567788 901123 301912 12345 67894 675788 858767 | 3  0 1 1 2 3 9  5 6 7 7 8 8 |
| 3 | 111222 345678 212121 121212 122112 907812 121212 235623 212121 222113 345235 | 6  1 1 1 2 2 2 |

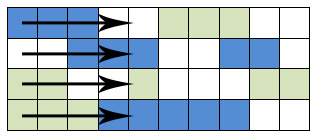
**Задача 18** *(V Республиканская олимпиада, 2015 г.)*

Жители СВУД в свободное от работы время очень любят решать различные задачи, связанные с графическими изображениями. В последнее время пользуется популярностью следующая задача, представляющая собой графический аналог задачи на определение слов-палиндромов (слова, читающиеся одинаково как слева направо, так и справа налево (например, казак, шалаш)).

Задано 24-битовое графическое изображение шириной 600 и высотой 200 пикселей. Изображение содержит некоторое слово на белом фоне, буквы слова могут иметь разный цвет. Расстояния между буквами могут различаться, при этом отдельные буквы слова могут находиться как выше, так и ниже друг друга. Требуется определить, является ли изображенное слово графическим палиндромом. В СВУД считается, что слово является графическим палиндромом, если сравниваемые буквы имеют один и тот же регистр (например, слово «казак» является графическим палиндромом, а слово «Казак» уже не является).

Попробуйте и Вы решить эту задачу.

*Входные данные:* В файле IN.TXT через пробел записаны последовательно тройки чисел, представляющие собой последовательности оттенков всех точек изображения в формате RGB (каждой точке соответствуют три оттенка). Точки (пиксели) перечисляются в порядке следования в изображении, начиная с левого верхнего угла слева направо, построчно:



*Выходные данные:* Выходной файл OUT.TXT единственную строку, содержащую или «Да» (если слово является графическим палиндромом) или «Нет» (если слово не является графическим палиндромом).

*Примеры*

|  |  |  |
| --- | --- | --- |
| № | IN.TXT | OUT.TXT |
| 1 | Без имени-1 | Да |
| 2 | Без имени-1 | Нет |
| 3 | Без имени-1 | Да |

**Задача 19** *(V Республиканская олимпиада, 2015 г.)*

В уже знакомой Вам сказочной стране СВУД совершенствуются технологии исключения шумовой составляющей в отсканированных изображениях документов. При этом большинство документов в стране СВУД не имеют строго определенных верхних полей, тогда как значения левого, правого и нижнего полей вполне определены и не превышают в отсканированных изображениях 100 пикселей. Отсканированные документы хранятся в виде 24-битовых растровых графических файлов. Фон документов до сканирования – белый.

С целью активизации и поддержки исследователей объявлен конкурс проектов и алгоритмов, позволяющих наиболее эффективно исключать шумы в изображениях.

Попробуйте и Вы свои силы в разработке таких алгоритмов.

*Входные данные:* В первой строке файла IN.TXT через пробел записаны два числа – ширина и высота изображения в пикселях. В следующих строках файла последовательно перечисляются тройки чисел, представляющие собой последовательности оттенков всех точек изображения в формате RGB (каждой точке соответствуют три оттенка). Точки перечисляются в порядке следования в изображении, начиная с левого верхнего угла слева направо, построчно.

*Выходные данные:* Выходные данные представляют собой все тот же входной файл, в котором в какой-то мере исключена шумовая составляющая.

**К решению задачи должно быть обязательно приложено достаточно подробное текстовое описание предлагаемого метода.**

*Примеры*

|  |  |  |
| --- | --- | --- |
| № | IN.TXT | OUT.TXT |
| 1 | Без имени-1 | Без имени-1 |
| 2 | Без имени-1 | Без имени-1 |
| 3 | Без имени-1 | Без имени-1 |

**Задача 20** *(I городская дистанционно-очная олимпиада, 2014 г.)*

Дана строка, состоящая из букв «Z». Количество букв в строке равно *N*, причем . Определим операцию замены, которая характеризуется тремя параметрами  и состоит в замене на букву  букв строки, начиная с позиции  до позиции . Необходимо определить, сколько различных букв будет в строке после выполнения заданной последовательности операций замены.

*Ограничения задачи*:

* программа должна использовать для размещения данных (переменных, массивов, и т.д.) не более 64 Кб памяти;
* время вычислений: не более 1 секунды.

*Входные данные:* В первой строке входного файла INPUT.TXT записано целое число – длина строки *N* (). Во второй строке содержится число замен *M* (). В следующих *M* строках содержатся тройки  где  – заглавная латинская буква,  и – целые числа ().

*Выходные данные:* Выходной файл OUTPUT.TXT должен содержать единственное число – количество различных букв в результирующей строке.

*Примеры*

№ INPUT.TXT OUTPUT.TXT

1. 20000000 4

3

B 10 5000

K 40 100

D 2000 10000

2. 1000000000 6

5

А 10 50

Х 90 1000

D 2000 10000

F 8000 900000000

L 70 200

3. 80000000 3

8

B 10 50

K 40 100

L 30 200

D 220 1000

Z 100 2000

M 1500 1600

Y 2000 3000

B 200 10000000