**Расстояние Хэмминга 65**

*(Время: 1 сек. Память: 16 Мб Сложность: 27%)*

В связи с особенностями линии связи, используемой для передачи сообщений из пункта A в пункт B, каждый бит принятого сообщения с вероятностью 0.001 содержит ошибку.

Из пункта A в пункт B было послано одно из n сообщений m1, m2, ..., mn. В пункте B было принято сообщение s.

Ваша задача заключается в определении наиболее вероятного исходного сообщения. Очевидно, что оно будет одним из тех сообщений, расстояние Хэмминга между которым и строкой s минимально.

Расстоянием Хэмминга двух строк a и b одинаковой длины называется количество позиций, в которых эти строки различаются (количество элементов в множестве {i | 1 ≤ i ≤ |a|, ai ≠ bi }).

**Входные данные**

Первая строка входного файла INPUT.TXT содержит s — принятое сообщение. Вторая строка содержит целое число n — количество сообщений, которые могли быть отправлены. Следующие n строк содержат mi — эти сообщения. Длины всех сообщений равны (|s| = |m1| = |m2| = ... = |mn|). Сообщения непустые, состоят только из символов 0 и 1. Размер входного файла не превосходит 60 Кб.

**Выходные данные**

В первую строку выходного файла OUTPUT.TXT выведите k — количество сообщений, на которых достигается минимум расстояния Хэмминга. Во вторую строку выведите в порядке возрастания k чисел — номера этих сообщений.

**Пример**

|  |  |  |
| --- | --- | --- |
| № | INPUT.TXT | OUTPUT.TXT |
| 1 | 010101 3 110011 011001 000111 | 2 2 3 |

**Весовая комбинаторика 317**

*(Время: 0,5 сек. Память: 16 Мб Сложность: 27%)*

Ириска весит X грамм, мандарин – Y грамм, пряник – Z грамм.

Требуется написать программу, которая определит, сколько различных комбинаций весом ровно W грамм можно получить.

**Входные данные**

В единственной строке входного файла INPUT.TXT содержится четыре целых числа X, Y, Z и W (1 ≤ X, Y, Z ≤ 100, 1 ≤ W ≤ 1000).

**Выходные данные**

Выходной файл OUTPUT.TXT должен содержать одно целое число – количество вариантов подарков.

**Пример**

|  |  |  |
| --- | --- | --- |
| № | INPUT.TXT | OUTPUT.TXT |
| 1 | 10 25 15 40 | 3 |

**Нахождение периметра неправильной фигуры 265**

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

Из шахматной доски по границам клеток выпилили связную (не распадающуюся на части) фигуру без дыр. Требуется определить ее периметр.

**Входные данные**

Во входном файле INPUT.TXT сначала записано число N (1 ≤ N ≤ 64) – количество выпиленных клеток. В следующих N строках указаны координаты выпиленных клеток, разделенные пробелом (номер строки и столбца – числа от 1 до 8). Каждая выпиленная клетка указывается один раз.

**Выходные данные**

В выходной файл OUTPUT.TXT выведите одно число – периметр выпиленной фигуры (сторона клетки равна единице).

**Примеры**

|  |  |  |
| --- | --- | --- |
| № | INPUT.TXT | OUTPUT.TXT |
| 1 | 3 1 1 1 2 2 1 | 8 |
| 2 | 1 8 8 | 4 |

**Распаковка строки 231**

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

Будем рассматривать только строчки, состоящие из заглавных английских букв. Например, рассмотрим строку AAAABCCCCCDDDD. Длина этой строки равна 14. Поскольку строка состоит только из английских букв, повторяющиеся символы могут быть удалены и заменены числами, определяющими количество повторений. Таким образом, данная строка может быть представлена как 4AB5C4D. Длина такой строки 7. Описанный метод мы назовем упаковкой строки.

Напишите программу, которая берет упакованную строчку и восстанавливает по ней исходную строку.

**Входные данные**

Входной файл INPUT.TXT содержит одну упакованную строку. В строке могут встречаться только конструкции вида nA, где n — количество повторений символа (целое число от 2 до 99), а A — заглавная английская буква, либо конструкции вида A, то есть символ без числа, определяющего количество повторений. Строка содержит от 1 до 80 символов.

**Выходные данные**

В выходной файл OUTPUT.TXT выведите восстановленную строку. При этом строка должна быть разбита на строчки длиной ровно по 40 символов (за исключением последней, которая может содержать меньше 40 символов).

**Примеры**

|  |  |  |
| --- | --- | --- |
| № | INPUT.TXT | OUTPUT.TXT |
| 1 | 3A4B7D | AAABBBBDDDDDDD |
| 2 | 22D7AC18FGD | DDDDDDDDDDDDDDDDDDDDDDAAAAAAACFFFFFFFFFF FFFFFFFFGD |
| 3 | 95AB | AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAB |
| 4 | 40AB39A | AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA BAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA |

**Сравнение коллекций 522**

Два массива называются похожими, если совпадают множества чисел, встречающихся в этих массивах.

Требуется написать программу, которая определит: похожи ли два заданных массива.

**Входные данные**

Входной файл INPUT.TXT содержит в первой строке два числа M и N - длины массивов (1 ≤ M, N ≤ 16000). Во второй строке записаны M чисел – элементы первого массива. В третьей строке записаны N чисел – элементы второго массива. Числа в строках разделены пробелами, элементы массивов - целые числа, не превышающие 32000 по абсолютной величине.

**Выходные данные**

Выходной файл OUTPUT.TXT должен содержать 1, если массивы похожи и 0 иначе.

**Примеры**

|  |  |  |
| --- | --- | --- |
| № | INPUT.TXT | OUTPUT.TXT |
| 1 | 4 3 1 2 3 2 1 2 3 | 1 |
| 2 | 2 3 1 2 2 3 1 | 0 |

**Алфавитная комбинаторика 878**

Воспитательница Галя работает в детском саду. Кроме детских игр в этом детском саду проходят занятия. Вот уже неделю ребята изучают буквы английского алфавита. Каждое утро воспитательница выстраивает всех своих подопечных в ряд и они играют в игру.

Первый ребенок в ряду громко называет первую букву алфавита – A. Второй должен назвать B, третий – C и так далее. По счастливому стечению обстоятельств всего в группе 26 детей и столько же сколько и букв в английском алфавите.

Если каждый ребенок без ошибки назовет свою букву, то группа отпразднует знание английского алфавита и ребята по этому случаю съедят большой торт. Однако, пока что группе не удается правильно назвать все 26 букв и каждое утро то один, то другой называет свою букву неправильно и игра на этом заканчивается.

Ребята учат буквы подряд и каждый из них знает первые несколько букв алфавита и не может назвать остальные. Поэтому возможность выиграть напрямую зависит от их расстановки. К примеру, если последним в ряду окажется ребенок, знающий английский алфавит только до буквы D, то букву Z он назвать не сможет и группа не выиграет независимо от того, насколько хорошо выучили алфавит остальные ребята.

Галя считает, что группа в целом уже достаточно хорошо знает алфавит и хочет помочь своим ребятам выиграть. Для этого ей нужно всего лишь расставить их так, чтобы первый ребенок в ряду знал алфавит хотя бы до буквы A, второй хотя бы до буквы B и так далее, последний в ряду должен знать все буквы.

Помогите Гале решить: в каком порядке расставить ребят, чтобы они смогли выиграть или выясните, что это пока невозможно.

**Входные данные**

Входной файл INPUT.TXT содержит строку, состоящую из 26 заглавных букв английского алфавита, записанных слитно; i-я из этих букв говорит о том, до какой буквы знает алфавит i-й ребенок.

**Выходные данные**

В первой строке выходного файла OUTPUT.TXT выведите YES, если воспитательнице удастся выстроить своих детей в ряд так, чтобы они выиграли и NO, если им для этого еще надо поучиться. Если расстановка возможна, во второй строке выведите перестановку из 26 чисел от 1 до 26 через пробел – порядок детей в ряду. Если решений несколько, можно вывести любое.

**Примеры**

|  |  |  |
| --- | --- | --- |
| № | INPUT.TXT | OUTPUT.TXT |
| 1 | ABCDEFGHIJKLMNOPQRSTUVWXYZ | YES 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 |
| 2 | BCARTYXYZZYZZYXYXYZZYZZYXV | YES 3 1 2 4 5 26 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 25 24 23 |
| 3 | AAZZZZZZZZZZZZZZZZZZZZZZZZ | NO |

**Ряд Фибоначчи 623**

*(Время: 2 сек. Память: 16 Мб Сложность: 23%)*

Вам наверняка знакомы числа Фибоначчи: 1, 1, 2, 3, 5, 8, 13, 21... Они определяются рекуррентным соотношением: Fn = Fn-1 + Fn-2, F0 = F1 = 1.

Требуется найти последнюю цифру n-го числа Фибоначчи.

**Входные данные**

Во входном файле INPUT.TXT содержится одно целое число n (0 ≤ n ≤ 108).

**Выходные данные**

В выходной файл OUTPUT.TXT необходимо вывести одно число - последнюю цифру числа Fn.

**Примеры**

|  |  |  |
| --- | --- | --- |
| № | INPUT.TXT | OUTPUT.TXT |
| 1 | 1 | 1 |
| 2 | 5 | 8 |

**Числовая сортировка 246**

*(Время: 1 сек. Память: 16 Мб Сложность: 23%)*

Ежедневно диспетчеру железнодорожной станции приходится переставлять вагоны во многих поездах, чтобы они шли в заданном порядке. Для этого диспетчер может расцепить пришедший на станцию состав в произвольных местах и переставить образовавшиеся сцепки из одного или нескольких вагонов в произвольном порядке. Порядок вагонов в одной сцепке менять нельзя, также нельзя развернуть всю сцепку так, чтобы последний вагон в сцепке оказался первым в ней.

Диспетчер просит вашей помощи в определении того, какое минимальное число соединений между вагонами необходимо расцепить, чтобы переставить вагоны в составе в требуемом порядке.

**Входные данные**

В первой строке входного файла INPUT.TXT содержится целое число N (1 ≤ N ≤ 100). Во второй строке содержится перестановка натуральных чисел от 1 до N (то есть все натуральные числа от 1 до N в некотором порядке). Числа разделяются одним пробелом. Эта перестановка задает номера вагонов в приходящем на станцию составе. Требуется, чтобы в уходящем со станции составе вагоны шли в порядке их номеров.

**Выходные данные**

Программа должна записать в выходной файл OUTPUT.TXT единственное целое число, равное минимальному количеству соединений между вагонами, которое нужно расцепить в данном составе, чтобы их можно было переставить по порядку.

**Примеры**

|  |  |  |
| --- | --- | --- |
| № | INPUT.TXT | OUTPUT.TXT |
| 1 | 4 3 1 2 4 | 2 |
| 2 | 5 5 4 3 2 1 | 4 |
| 3 | 2 1 2 | 0 |

**Комбинаторный перебор 893**

*(Время: 1 сек. Память: 16 Мб Сложность: 21%)*

В соревнованиях по хоккею участвует N команд. Сколько существует вариантов распределения комплектов золотых, серебряных и бронзовых медалей, если одно призовое место может занять только одна команда?

**Входные данные**

Входной файл INPUT.TXT содержит целое число N (1 ≤ N ≤ 104) – количество команд.

**Выходные данные**

В выходной файл OUTPUT.TXT выведите единственное число – искомое количество способов награждения участников.

**Примеры**

|  |  |  |
| --- | --- | --- |
| № | INPUT.TXT | OUTPUT.TXT |
| 1 | 3 | 6 |
| 2 | 20 | 6840 |

**Морской бой 493**

«Морской бой» - игра для двух участников, в которой игроки по очереди называют координаты на неизвестной им карте соперника. Если у соперника по этим координатам имеется корабль, то корабль или его часть «топится», а попавший получает право сделать еще один ход. Цель игрока - первым поразить все корабли противника.

«Морской бой» очень популярен среди учеников одной физико-математической школы. Ребята очень любят в него играть на переменах. Вот и сейчас ученики Иннокентий и Емельян начали новую партию.

Правила, по которым ребята расставляют корабли перед началом партии, несколько отличаются от классических. Во-первых, игра происходит на поле размером N×M, а не 10×10. Во-вторых, число кораблей, их размер и форма выбираются ребятами перед партией - так играть намного интереснее.

Емельян уже расставил все свои корабли, кроме одного однопалубного. Такой корабль занимает ровно одну клетку.

Задана расстановка кораблей Емельяна. Найдите число способов поставить оставшийся однопалубный корабль. При этом учитывайте, что по правилам его можно ставить только в ту клетку, все соседние с которой не заняты. В этой задаче соседними считаются клетки, имеющие общую сторону.

**Входные данные**

Первая строка входного файла INPUT.TXT содержит два числа: N и M (1 ≤ N, M ≤ 100). Последующие N строк описывают игровое поле - каждая из них содержит M символов. Символом «.» (точка) обозначена свободная клетка, символом «\*» (звездочка) - занятая кораблем.

**Выходные данные**

В выходной файл OUTPUT.TXT выведите ответ на задачу.

**Примеры**

|  |  |  |
| --- | --- | --- |
| № | INPUT.TXT | OUTPUT.TXT |
| 1 | 4 4 \*\*\*\* \*\*.. \*... \*... | 4 |
| 2 | 4 3 \*\*\* ... ... \*\*\* | 0 |

**Сравнение чисел 13**

*(Время: 1 сек. Память: 16 Мб Сложность: 20%)*

Петя и Вася часто играют в различные логические игры. Недавно Петя поведал Васе о новой игре «Быки и коровы» и теперь они играют в эту игру сутками. Суть игры очень проста: Петя загадывает четырехзначное число, состоящее из различных цифр. Вася отгадывает задуманное Петей число, перебирая возможные варианты. Каждый раз Вася предлагает вариант своего четырёхзначного числа, состоящего из различных цифр, а Петя делает Васе подсказку: сообщает количество быков и коров, после чего Вася с учетом подсказки продолжает отгадывание числа до тех пор, пока не отгадает. Быки – это количество цифр в предложенном Васей числе, совпадающих по значению и стоящих в правильной позиции в задуманном Петей числе. Коровы – количество цифр, совпадающих по значению, но находящихся в неверной позиции. Например, если Петя задумал число 5671, а Вася предложил вариант 7251, то число быков равно 1 (только цифра 1 на своем месте), а число коров равно 2 (только цифры 7 и 5 не на своих местах). Петя силен в математике, но даже он может ошибаться. Помогите Пете написать программу, которая бы по загаданному Петей и предложенному Васей числам сообщала количество быков и коров.

**Входные данные**

В единственной строке входного файла INPUT.TXT записано два четырехзначных натуральных числа A и B через пробел, где А – загаданное Петей число, а В – предложенный Васей вариант.

**Выходные данные**

В выходной файл OUTPUT.TXT нужно вывести два целых числа через пробел — количество быков и коров.

**Примеры**

|  |  |  |
| --- | --- | --- |
| № | INPUT.TXT | OUTPUT.TXT |
| 1 | 5671 7251 | 1 2 |
| 2 | 1234 1234 | 4 0 |
| 3 | 2034 6234 | 2 1 |

**Статистика оценок 5**

*(Время: 1 сек. Память: 16 Мб Сложность: 15%)*

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 целочисленного массива (1 ≤ N ≤ 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**

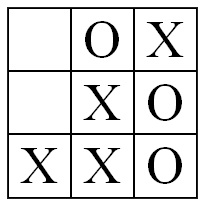
В первую строку выходного файла OUTPUT.TXT нужно вывести числа, которые соответствуют дням месяцев, в которые Вася получил тройки, а во второй строке соответственно расположить числа месяца, в которые Вася получил четверки. В третьей строке нужно вывести «YES», если Вася может рассчитывать на четверку 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**

|  |  |  |
| --- | --- | --- |
| № | 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 |

**Оценка комбинации 895**

*(Время: 1 сек. Память: 16 Мб Сложность: 19%)*

*Крестики-нолики* – логическая игра между двумя противниками на квадратном поле 3 на 3 клетки. Один из игроков играет «крестиками» (тот, кто ходит первым), другой – «ноликами». Игроки по очереди ставят на свободные клетки поля знаки (один всегда «крестики», другой всегда «нолики»). Первый, выстроивший в ряд три своих фигуры по вертикали, горизонтали или диагонали, выигрывает и на этом игра заканчивается. В том случае, когда все клетки заполнены и победитель не определен, игра завершается ничьей.

По состоянию игрового поля в конце игры требуется определить результат игры для первого игрока: выиграл, проиграл или сыграл вничью.

**Входные данные**

Входной файл INPUT.TXT содержит информацию об игровом поле – три строки по три символа в каждой. Символ «X» (ASCII 88) означает «крестик», символ «O» (ASCII 79) - «нолик», а символ «.» (ASCII 46) - пустую клетку.

**Выходные данные**

В выходной файл OUTPUT.TXT выведите в случае победы первого игрока «Win», в случае его проигрыша – «Lose» и «Draw» в случае ничьей.

**Примеры**

|  |  |  |
| --- | --- | --- |
| № | INPUT.TXT | OUTPUT.TXT |
| 1 | .OX .XO XXO | Win |
| 2 | OXO .OX OXX | Lose |
| 3 | XOX XOX OXO | Draw |

**Начальные элементы последовательности Фибоначчи 949**

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

Последовательность чисел a1, a2, …, ai,… называется Фибоначчиевой, если для всех i ≥ 3 верно, что ai=ai-1+ai-2, то есть каждый член последовательности (начиная с третьего) равен сумме двух предыдущих.

Ясно, что, задавая различные числа a1 и a2, мы можем получать различные такие последовательности, и любая Фибоначчиева последовательность однозначно задается двумя своими первыми членами.

Будем решать обратную задачу. Вам будет дано число n и два члена последовательности: an и an+1. Вам нужно написать программу, которая по их значениям найдет a1 и a2.

**Входные данные**

Входной файл INPUT.TXT содержит число n и значения двух членов последовательности: an и an+1 (1 ≤ n ≤ 30, члены последовательности от первого до (n+1)-го — целые числа, по модулю не превышающие 2×109).

**Выходные данные**

В выходной файл OUTPUT.TXT выведите два числа — значения первого и второго членов этой последовательности.

**Пример**

|  |  |  |
| --- | --- | --- |
| № | INPUT.TXT | OUTPUT.TXT |
| 1 | 4 3 5 | 1 1 |

**Числовой палиндром 324**

*(Время: 1 сек. Память: 16 Мб Сложность: 10%)*

Требуется написать программу, определяющую, является ли четырехзначное натуральное число N палиндромом, т.е. числом, которое одинаково читается слева направо и справа налево.

**Входные данные**

Входной файл INPUT.TXT содержит натуральное число N (1000 ≤ N ≤ 9999).

**Выходные данные**

В выходной файл OUTPUT.TXT следует вывести слово «YES», если число N является палиндромом, или «NO» – если нет.

**Примеры**

|  |  |  |
| --- | --- | --- |
| № | INPUT.TXT | OUTPUT.TXT |
| 1 | 6116 | YES |
| 2 | 1231 | NO |

**Время года 892**

*(Время: 1 сек. Память: 16 Мб Сложность: 10%)*

По заданному номеру месяца в году требуется определить время года.

**Входные данные**

Входной файл INPUT.TXT содержит натуральное число N (N≤100) – номер месяца.

**Выходные данные**

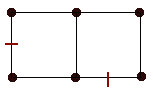
В выходной файл OUTPUT.TXT выведите для летних месяцев значение «Summer», для зимних – «Winter», для весенних – «Spring», для осенних – «Autumn». Если число не соответствует возможному значению месяца, то в этом случае следует вывести «Error».

**Примеры**

|  |  |  |
| --- | --- | --- |
| № | INPUT.TXT | OUTPUT.TXT |
| 1 | 3 | Spring |
| 2 | 15 | Error |

**Устойчивость на разрыв 756**

*(Время: 1 сек. Память: 16 Мб Сложность: 10%)*

Браконьер Петрович использует распространенный незаконный способ рыбалки с использованием рыболовной сети. Но проблема в том, что крупная рыба часто рвет сеть и приходится ее восстанавливать. Однажды Петрович задумался: какое максимальное количество повреждений может быть в рыболовной сети, таких, что сеть не будет разорвана на части? Вам предстоит помочь ему в вычислениях.

Сеть имеет прямоугольную форму размером M×N узлов, все смежные узлы соединены леской. Под разрывом будем понимать только единичный обрыв лески между двумя смежными узлами сети.

Например, если сеть имеет размер 2х2, то внешний вид сети будет напоминать квадрат, где допустим только один разрыв в одном из четырех возможных соединений, т.к. любые 2 разрыва приведут к разделению сети на 2 части.

**Входные данные**

Первая строка входного файла INPUT.TXT содержит два целых числа M и N через пробел – размеры рыболовной сети (1 ≤ M, N ≤ 10 000).

**Выходные данные**

В выходной файл OUTPUT.TXT выведите максимальное число разрывов заданной сети, которые не приведут к распадению рыболовной снасти Петровича.

**Примеры**

|  |  |  |
| --- | --- | --- |
| № | INPUT.TXT | OUTPUT.TXT |
| 1 | 2 2 | 1 |
| 2 | 2 3 | 2 |