Фестиваль «Юные интеллектуалы Среднего Урала»

Муниципальный этап Всероссийской олимпиады по информатике

2012 - 2013 учебный год

Разборы решений и идеи тестов

Фестиваль «Юные интеллектуалы Среднего Урала» Муниципальный этап Всероссийской олимпиады по информатике $2012-2013 \ {\rm учебный} \ {\rm год}$ $8 \ {\rm класc}$

Время выполнения задач — 4 часа Ограничение по времени — 2 секунды на тест Ограничение по памяти — 64 мегабайта

8.1. «Наибольший делитель». Вводится чётное натуральное число, большее двух. Выведите его наибольший делитель, отличный от него самого.

Формат ввода: Вводится единственное чётное натуральное число, не превосходящее 1000.

Формат вывода: Выводится единственное натуральное число — максимальный делитель исходного числа, меньший его.

Пример

input.txt: output.txt:
6 3

8.2. «Урок информатики». В компьютерном классе стоит n компьютеров. На урок информатики пришло m школьников, каждый из которых должен сесть за отдельный компьютер. К сожалению, оказалось так, что не все компьютеры находились в рабочем состоянии, так что k школьникам компьютеров не хватило. По данным n, m, k определите, сколько компьютеров сломано.

Если всем школьникам хватило компьютеров, то все компьютеры в классе считаем рабочими.

Формат ввода: В единственной строке входного файла через пробел записаны три неотрицательных целых числа $n, m, k \ (0 \le n \le 1000, 0 \le k \le m \le 2000)$. Гарантируется, что данные описывают возможную ситуацию.

Формат вывода: Выведите единственное неотрицательное целое число — количество неисправных компьютеров в классе.

Пример

input.txt: output.txt:
12 10 1 3

8.3. «Проблемы агронома». Имеется прямоугольная территория, рассечённая на отдельные участки лесополосами, параллельными его сторонам. Лесопосадки считаются имеющими бесконечно малую ширину, то есть, линиями. На одном из этих участков планируется организовать поле под картошку, причём желательно как можно большего размера. По данным об имеющихся лесополосах найдите площадь такого поля.

1

Формат ввода: В первой строке заданы два натуральных числа n и m — размеры исходной территории в направлении восток — запад и север — юг ($1 \le n, m \le 500$). Во второй строке задано число k_1 — количество лесополос, параллельных южной (нижней) стороне с длиной n. В третьей строке через пробел идут k_1 натуральных чисел в порядке возрастания — расстояния лесополос от южного (нижнего) края территории (все расстояния различны). Соответственно, в третьей строке задано число k_2 — количество лесополос, параллельных западной (левой) стороне с длиной m. В пятой строке через пробел идут k_2 натуральных чисел в порядке возрастания — расстояния лесополос от западного (левого) края территории (все расстояния различны). Считается, что $0 \le k_1, k_2 \le 10$.

Формат вывода: Выведите единственное натуральное число — площадь наибольшего имеющегося участка, ограниченного лесополосами и/или краями территории.

Пример

input.txt: output.txt:
5 5 9
1
2
2
1 4

8.4. «Перед заселением». В новом многоквартирном доме n квартир, имеющих номера от 1 до n. Для того, чтобы набить эти номера на дверь, было закуплено k литер цифр. По заданному k определите, сколько квартир в доме, или сообщите, что закуплено неверное количество литер.

Формат ввода: В первой строке файла задано единственно натуральное число $k \ (1 \leqslant k \leqslant 10000).$

Формат вывода: Выведите число n, если k — количество цифр в числах от 1 до n, или ERROR, если закупщики ошиблись.

Пример 1		Пример 2	
input.txt:	output.txt:	input.txt:	output.txt:
11	10	12	ERROR

8.5. «"Треугольные" слова». Числа вида $1+2+\ldots+n=n(n+1)/2$ называются *треугольными* (потому что они описывают количество точек в треугольнике, в первой строке которого стоит одна точка, во второй — две точки, и т.д. до n-й строчки, в которой стоит n точек). Под *словом* будем понимать последовательность заглавных латинских букв. Каждому слову сопоставим число — сумму номеров его букв в латинском алфавите ABCDEFGHI JKLMNOPQRSTUVWXYZ (номера отсчитываются от 1). Например, слову PASCAL сопоставится число 16+1+19+3+1+12=52, а слову NEWTON сопоставится число 14+5+23+20+15+14=91. При этом

число 52 не является треугольным, а $91 = 12 \cdot (12+1)/2$ — это 12-е треугольное число. Слова, которым сопоставляются треугольные числа, также будем называть *треугольными*. Определите, сколько треугольных слов имеется в предложенном списке.

Формат ввода: В первой строке записано натуральное число n — количество слов, предлагаемых на анализ ($1 \le n \le 100$). В следующих n строках по одному на строке записаны слова. Каждая строка непуста и имеет длину не более 255 символов и завершается символом перевода строки.

Формат вывода: Выведите единственное число — количество треугольных слов среди предложенных.

Пример

```
input.txt: output.txt:
3 2
PASCAL
NEWTON
C
```

Фестиваль «Юные интеллектуалы Среднего Урала» Муниципальный этап Всероссийской олимпиады по информатике 2012 – 2013 учебный год 8 класс. Разбор решений и идеи тестов

Далее в разборах используются обозначения типов: short int — 2-байтовый целый тип (integer в TurboPascal/BorlandPascal и int в BorlandC++), int — 4-байтовый целый тип (longint в TurboPascal/BorlandPascal/FreePascal, long int в BorlandC++, integer в Delphi/PascalABC и int в VisualC++/C++ Builder), int64 — 8-байтный целый (отсутствует в BorlandPascal и BorlandC++, но присутствует в системах программирования под Windows).

Задачи оцениваются по 100 баллов за каждую; стало быть, полная стоимость пакета — 500 баллов. В рамках одной задачи все тесты считать равноценными.

8.1. «Наибольший делитель». Вводится чётное натуральное число, большее двух. Выведите его наибольший делитель, отличный от него самого.

По мнению программного комитета эта задача является «утешительной».

Идея решения проста: наибольший делитель чётного числа, отличный от самого числа — это есть данное число, делённое пополам. Таким образом, нужно считать число, поделить пополам и вывести частное.

Идеи тестов:

10 тестов — числа 2, 10, 16, 50, 52, 100, 128, 500, 512, 1000.

8.2. «Урок информатики». В компьютерном классе стоит n компьютеров. На урок информатики пришло m школьников, каждый из которых должен сесть за отдельный компьютер. K сожалению, оказалось так, что не все компьютеры находились в рабочем состоянии, так что k школьникам компьютеров не хватило. По данным n, m, k определите, сколько компьютеров сломано.

Если всем школьникам хватило компьютеров, то все компьютеры в классе считаем рабочими.

Решение этой задачи несложно, хотя алгоритм требует некоторого размышления. Если всем хватило компьютеров, то есть если k равно 0, то считаем, что неисправных компьютеров нет, ответ равен нулю.

Пусть теперь k не равно нулю. Обозначим количество неисправных компьютеров через b. Тогда доступными для работы являются n-b компьютеров. Стало быть, n-b школьников сели за компьютеры, а k школьникам компьютеров не хватило. Отсюда получаем, что (n-b)+k=m, или b=n+k-m. По условию задачи гарантируется, что эта величина неотрицательна.

Идеи тестов:

1. Минимальный тест — все три числа равны нулю.

- 2. В классе установлено 0 компьютеров, сколько-то ребят пришло.
- 3. Компьютеров больше, чем пришло ребят.
- 4. Ребят меньше, чем компьютеров; все компьютеры неисправны.
- 5. Ребят столько же, сколько компьютеров; всем хватило рабочих мест.
- 6. Ребят столько же, сколько компьютеров; не всем хватило рабочих мест.
- 7. Ребят столько же, сколько компьютеров; никому не хватило рабочих мест.
- 8. Ребят больше, чем компьютеров; неисправных компьютеров нет.
- 9. Ребят больше, чем компьютеров; часть компьютеров неисправна.
- 10. Ребят больше, чем компьютеров; все компьютеры неисправны.
- 11. Установлено максимальное число компьютеров, ребят пришло меньше.
- 12. Установлено максимальное число компьютеров, ребят пришло столько же.
- 13. Установлено максимальное число компьютеров, ребят пришло больше, часть компьютеров неисправна.
- 14. Максимальный тест: n = 1000, m = 2000, k = 1000.
- 15. Максимальный тест: n = 1000, m = 2000, k = 2000.
- 16-20. Случайные тесты.

8.3. «Проблемы агронома». Имеется прямоугольная территория, рассечённая на отдельные участки лесополосами, параллельными его сторонам. Лесопосадки считаются имеющими бесконечно малую ширину, то есть, линиями. На одном из этих участков планируется организовать поле под картошку, причём желательно как можно большего размера. По данным об имеющихся лесополосах найдите площадь такого поля.

Сложность данной задача оценивается программным комитетом как средняя.

Несложно понять, что максимальный участок имеет максимальные стороны среди возможных. Стало быть, решение заключается в поиске длины максимального отрезка в одном направлении, в другом и в последующем перемножении полученных длин.

Некоторую сложность представляет сам поиск. Было бы удобнее, если бы в последовательность координат лесополос были также включены ещё 0 — отметка левого (нижнего) края поля — и n (m) — отметка правого (верхнего) края поля. Соответственно, можно или считать массив и дополнить его указанными элементами, или же аккуратно проводить обработку чисел во время их считывания.

Обработка заключается в следующем (описывается более сложный вариант с обработкой при чтении данных). Читается первый элемент последовательности, объявляется текущим максимумом. При прочтении очередного элемента вычисляется максимум из текущего максимального и разности считанного элемента и предыдущего. После того, как все элементы прочитаны, ищем максимум из текущего максимального и разности координаты соответствующего края и последнего считанного элемента.

- 1. Минимальный тест лесополосы отсутствуют.
- 2. Есть одна лесополоса «север-юг»;
- 3. Есть одна лесополоса «запад-восток»;
- 4. Есть по одной лесополосе в обоих направлениях;
- 5. нет лесополос «запад-восток», есть две лесополосы «север-юг», максимум достигается на первом отрезке (у левого края);
- 6. нет лесополос «запад-восток», есть две лесополосы «север-юг», максимум достигается на среднем отрезке;
- 7. нет лесополос «запад-восток», есть две лесополосы «север-юг», максимум достигается на последнем отрезке (у правого края);
- 8. Максимальный тест по 10 полос в каждом направлении;
- 9-20. Случайные тесты.
- **8.4.** «Перед заселением». В новом многоквартирном доме п квартир, имеющих номера от 1 до п. Для того, чтобы набить эти номера на дверь, было закуплено к литер цифр. По заданному к определите, сколько квартир в доме, или сообщите, что закуплено неверное количество литер.

Задача оценивается программным комитетом как достаточно сложная, так как требует либо реализации алгоритма поиска количества цифр в числе, либо хорошей идеи о решении.

«Лобовое» решение связано с перебором всех чисел от 1 до тех пор, пока количество литер, необходимых для установки номеров от 1 до текущего числа, не станет большим или равным k (при этом количество литер, необходимых для набора очередного числа получается делением этого числа нацело на 10, пока в частном не получится ноль, количество делений и есть количество литер). Если конечный результат равен k, то ответ — последнее рассмотренное число; если результат больше k — закуплено неверное количество литер.

Чуть более изящное решение исключает подсчёт количества цифр для каждого нового числа, а взамен этого увеличивает счётчик цифр при переходе через очередной порядок (от 9×10 , от 99×100 , от 999×1000 , и т.д.).

Наиболее эффективное решение состоит в следующем. Заметим, что для набивки всех однозначных номеров (от 1 до 9) требуется 9 литер; для набивки всех двузначных номеров (от 10 до 99), которых 90 штук, требуется 180 литер; для набивки всех трёхзначных номеров (от 100 до 999), которых 900 штук, требуется 2700 литер; наконец, для набивки всех четырёхзначных номеров (от 1000 до 9999), которых 9000 штук, требуется 36000 литер, что уже гарантированно больше количества закупленных литер. Действовать можно следующим образом. Если k <= 9, то n = k, иначе вычитаем из k число 9. Если новое значение k <= 180, то значит в доме может быть лишь двузначное число квартир; если это значение k кратно 2 (количеству цифр в одном двухзначном числе), то n = k/2 + 9; иначе ошибка. Аналогично, если новое значение k > 180, то вычитаем из него 180 и проводим

рассмотрение трёхзначных чисел, а при необходимости и четырёхзначных.

Идеи тестов:

Поскольку хитрых случаев в алгоритмическом плане данная задача не подразумевает, то предлагается 20 случайных тестов, включая и крайние значения для $n=1,\,9,\,10,\,99,\,100,\,999,\,1000,$ и максимально допустимого n - 2776.

8.5. «"Треугольные" слова». Числа вида $1+2+\ldots+n=n(n+1)/2$ называются треугольными (потому что они описывают количество точек в треугольнике, в первой строке которого стоит одна точка, во второй — две точки, и т.д. до n-й строчки, в которой стоит n точек). Под словом будем понимать последовательность заглавных латинских букв. Каждому слову сопоставим число — сумму номеров его букв в латинском алфавите ABCDEFGHIJKLMNOPQRSTUVWXYZ (номера отсчитываются от 1). Например, слову PASCAL сопоставится число 16+1+19+3+1+12=52, а слову NEWTON сопоставится число 14+5+23+20+15+14=91. При этом число 52 не является треугольным, а $91=12\cdot(12+1)/2-12$ 0 это 12-е треугольное число. Слова, которым сопоставляются треугольные числа, также будем называть треугольными. Определите, сколько треугольных слов имеется в предложенном списке.

Данная задача представляется достаточно сложной, так как включает в себя обработку строк (хотя и не слишком изощрённую) и работу с числами. От соединения этих двух не слишком сложных идей сложность всей задачи возрастает.

Собственно, на первом этапе обработки каждого слова надо найти сумму номеров его букв. В Си это можно сделать просто вычитанием из очередного символа строки символа 'A' (в Си можно вычитать символы друг из друга, получая разницу их кодов) и увеличением разности на 1. Чтобы произвести такое вычисление в Паскале надо предварительно вычислить код символа при помощи функции ord(...) и вычесть из него код символа 'A', после чего также увеличив результат на 1: ord(str[i]) - ord('A') + 1.

Потом надо понять, получилась ли сумма ${\tt s}$ номеров символов треугольным числом. Проще всего это сделать, решив относительно ${\tt n}$ выражение треугольного числа:

$$\frac{n(n+1)}{2} = s \implies n^2 + n - 2s = 0 \implies n_{1,2} = \frac{-1 \pm \sqrt{1+8s}}{2}.$$

Несложно понять, что из двух корней уравнения надо выбрать положительный, тот, который вычисляется со знаком «+». Если полученный корень целый, то число **s** треугольное.

Другой подход к проверке треугольности — просто априори вычислить таблицу треугольных чисел, не превышающих $26 \cdot 255 = 6630$ (максимальная сумма номеров символов 255-символьной строки — если она целиком состоит из букв Z) и каждое очередное число проверять на принадлежность этой таблице.

Соответственно, после написания процедуры проверки строки на треугольность решение всей задачи труда не составляет.

Идеи тестов:

Поскольку хитрых случаев в алгоритмическом плане данная задача не подразумевает, то предлагается 10 случайных тестов, включая и максимальный (с 100 строками по 255 каждая).

Фестиваль «Юные интеллектуалы Среднего Урала» Муниципальный этап Всероссийской олимпиады по информатике 2012-2013 учебный год

9 класс

Время выполнения задач — 4 часа Ограничение по времени — 2 секунды на тест Ограничение по памяти — 64 мегабайта

9.1. «День студента». Пусть 1 января пришлось на день недели с номером n (считаем, что понедельник имеет номер 1, вторник $-2,\ldots$, воскресенье -7). На какой день недели придётся День студента -25 января?

Формат ввода: Задано единственное число от 1 до 7 — номер дня недели 1 января.

Формат вывода: Выведите единственное число — номер дня недели 25 января.

Пример

input.txt: output.txt:
7 3

<u>Примечание к примеру:</u> В 2012 году 1 января было воскресеньем, а 25 января — средой.

9.2. «Шаг вперёд!». Участники школьных соревнований по информатике выстроились в шеренгу по росту. На груди у каждого участника прикреплён его двузначный личный номер. Естественно, номера вдоль шеренги получились в какомто произвольном порядке. Учитель информатики сказал сделать шаг вперёд тем участникам, номера соседей которых имеют одинаковую цифру в разряде единиц или в разряде десятков. То есть, вперёд должен шагнуть, например, участник, номера соседей которого 94 и 24 или 35 и 37, а тот, у кого соседи имеют номера 24 и 52, шагать вперёд не должен. Также не могут шагнуть вперёд и те, кто стоит в начале и конце строя. Сколько человек шагнут из строя?

Формат ввода: В первой строке файла записано натуральное число n — количество школьников ($1 \le n \le 90$). В следующей строке записано через пробел n двузначных натуральных чисел.

Формат вывода: Выведите единственное число — количество школьников, которые должны сделать шаг вперёд.

Пример

input.txt: output.txt:
5 2
10 25 17 75 72

9.3. «Верстаем текст». Текст представляет собой n абзацев. Известно, что абзац с номером k состоит из a_k строк. На страницу входит не более p строк. Абзацы при вёрстке не разбиваются при переходе на новую страницу: абзац целиком лежит на одной странице. Считается, что не бывает абзацев, размером более одной страницы. Сколько страниц будет создано при вёрстке этого текста?

Формат ввода: В первой строке заданы два натуральных числа p и n ($1 \le p \le 100$, $1 \le n \le 1000$). В следующей строке через пробел задано n величин a_k — размеры абзацев в том порядке, в котором они идут в тексте. Для всех индексов k верно, что $1 \le a_k \le p$.

Формат вывода: Выведите единственное натуральное число — получившееся количество страниц.

Пример

9.4. «Упро́щенная бухгалтерия». Главный бухгалтер фирмы «PROSTO» примерно оценивает сумму и выдаёт её характеристику. Например, сумма от 1 до 500 рублей оценивается словом МАLO, от 400 до 5000 рублей — SREDNE, от 3000 до 100000 рублей — MNOGO. Журнал финансовых операций компании включает в себя точную сумму, имевшуюся в начале на счету компании, количество операций, а затем описания операций с оценкой их сумм. Имеется три операции: «+» — приход, «-» — расход и «?» — запрос текущей суммы (см. пример). По имеющимся описаниям финансовых характеристик, используемых главным бухгалтером, и журналу операций определите диапазон, в котором лежит конечная сумма денег, имеющихся в распоряжении компании. Известно, что дела в компании ведутся аккуратно, и в долги компания не влезает.

Формат ввода: В первой строке файла записано натуральное число m — количество финансовых характеристик ($2\leqslant m\leqslant 10$). В следующих m строках идёт описание характеристик: через пробел перечислены натуральные числа a_i — минимальная сумма диапазона характеристики, b_i — максимальная сумма диапазона характеристики ($1\leqslant a_i\leqslant b_i\leqslant 10000$) и s_i — непустая строка (имеющая длину не более 255 символов и состоящая из заглавных латинских букв). Все строки s_i попарно различны. Затем в отдельной строке записано неотрицательное целое число K ($0\leqslant K\leqslant 10000$) — сумма, имевшаяся в распоряжении фирмы в начале. После этого в отдельной строке записано натуральное число n — количество операций в журнале ($0\leqslant n\leqslant 10000$). В последующих n строках приведены описания операций: символ операции «+», «-» или «?», затем через пробел строка, описывающая характеристику данной операции. Гарантируется, что последовательность записей об операциях корректна.

Формат вывода: Выведите через пробел два неотрицательных целых числа: минимально и максимально возможные значения конечной суммы на счету фирмы.

Пример

9.5. «Субнормальные числа». Назовём число k-субнормальным, если в его десятичной записи встречаются десятичные записи всех натуральных чисел из k цифр. Например, 1234567890 является 1-субнормальном числом, а 123456789 не является (нет записи числа 0). Определите, является ли заданное число k-субнормальным. Напомним, что ведущие нули в записи числа не допускаются.

Формат ввода: В первой строке записаны через пробел два натуральных числа $k\ (1\leqslant k\leqslant 4)$ и $n\ (1\leqslant n\leqslant 100000)$. Во второй строке содержится число из n цифр.

Формат вывода: Выведите YES, если данное число является субнормальным. Если же оно таковым не является, выведите минимальное число из k цифр, которое не встречается в данной записи.

Пример 1		Пример 2	
input.txt:	output.txt:	input.txt:	output.txt:
1 10	YES	1 10	2
1234567890		1034567890	

Фестиваль «Юные интеллектуалы Среднего Урала» Муниципальный этап Всероссийской олимпиады по информатике 2012—2013 учебный год 9 класс. Разбор решений и идеи тестов

Далее в разборах используются обозначения типов: short int — 2-байтовый целый тип (integer в TurboPascal/BorlandPascal и int в BorlandC++), int — 4-байтовый целый тип (longint в TurboPascal/BorlandPascal/FreePascal, long int в BorlandC++, integer в Delphi/PascalABC и int в VisualC++/C++ Builder), int64 — 8-байтный целый (отсутствует в BorlandPascal и BorlandC++, но присутствует в системах программирования под Windows).

Задачи оцениваются по 100 баллов за каждую; стало быть, полная стоимость пакета — 500 баллов. В рамках одной задачи все тесты считать равноценными.

9.1. «День студента». Пусть 1 января пришлось на день недели с номером n (считаем, что понедельник имеет номер 1, вторник $-2, \ldots$, воскресенье -7). На какой день недели придётся День студента -25 января?

Программный комитет считает эту задачу очень простой.

Наиболее простой способ решения этой задачи — вычислить на бумажке соответствие между номерами дней первого января и 25-го и перенести эти данные программу, вычисляя их, например при помощи case/switch.

Идеи тестов:

Предполагается пять тестов: с номерами дня недели первого января от 1 до 5.

9.2. «Шаг вперёд!». Участники школьных соревнований по информатике выстроились в шеренгу по росту. На груди у каждого участника прикреплён его двузначный личный номер. Естественно, номера вдоль шеренги получились в каком-то произвольном порядке. Учитель информатики сказал сделать шаг вперёд тем участникам, номера соседей которых имеют одинаковую цифру в разряде единиц или в разряде десятков. То есть, вперёд должен шагнуть, например, участник, номера соседей которого 94 и 24 или 35 и 37, а тот, у кого соседи имеют номера 24 и 52, шагать вперёд не должен. Также не могут шагнуть вперёд и те, кто стоит в начале и конце строя. Сколько человек шагнут из строя?

Программный комитет считает эту задачу достаточно простой.

Основной элемент решения — процедура разделения двузначного числа на цифры десятков и единиц. Понятно, что единицы есть остаток от целочисленного деления числа на 10, а десятки — остаток от такого деления.

Для решения задачи надо для каждого элемента считанного массива, кроме крайних, взять его соседние элементы, разобрать их на единицы и десятки и проверить, равны ли величины единиц или десятков.

Идеи тестов:

Поскольку идея решения не слишком сложна и не подразумевает каких-либо подводных камней, то для проверки предлагается 10 случайных тестов, включая минимальные $(n=1\ \mathrm{u}\ n=2)\ \mathrm{u}$ максимальный (n=90).

9.3. «Верстаем текст». Текст представляет собой п абзацев. Известно, что абзац с номером k состоит из a_k строк. На страницу входит не более p строк. Абзацы при вёрстке не разбиваются при переходе на новую страницу: абзац целиком лежит на одной странице. Считается, что не бывает абзацев, размером более одной страницы. Сколько страниц будет создано при вёрстке этого текста?

По мнение программного комитета, эта задача имеет средний уровень, но при этом является чисто технической.

Процедура, решающая эту задачу, выглядит следующим образом: начинаем суммирование считанных длин абзацев. Как только очередная сумма (то есть суммарное количество строк) превышает размер страницы p, то счётчик страниц увеличивается, а накопленное значение суммы приравнивается последнему прибавленному числу. В конце суммирования, если накопленная сумма отлична от нуля, то счётчик страниц надо увеличить на один (ситуация недозаполненной последней страницы).

- 1. Минимальный тест размер страницы равен 1.
- 2. Все абзацы входят на одну страницу, остаётся свободное место.
- 3. Все абзацы входят на одну страницу, свободного места не остаётся.
- 4. Все абзацы входят на две страницы, первая страница заполнена полностью, на второй остаётся свободное место.
- 5. Все абзацы входят на две страницы, обе страницы заполнены полностью.
- 6. Все абзацы входят на две страницы, обе страницы заполнены не полностью.
- 7. Все абзацы входят на две страницы, первая страница заполнена не полностью, вторая заполнена полностью.
- 8. Максимальный тест: p = 100, n = 1000.
- 9–20. Случайные тесты.
- 9.4. «Упро́щенная бухгалтерия». Главный бухгалтер фирмы «PROSTO» примерно оценивает сумму и выдаёт её характеристику. Например, сумма от 1 до 500 рублей оценивается словом MALO, от 400 до 5000 рублей SREDNE, от 3000 до 100000 рублей MNOGO. Журнал финансовых операций компании включает в себя точную сумму, имевшуюся в начале на счету компании, количество операций, а затем описания операций с оценкой их сумм. Имеется три операции: «+» приход, «-» расход и «?» запрос текущей суммы (см. пример).

По имеющимся описаниям финансовых характеристик, используемых главным бухгалтером, и журналу операций определите диапазон, в котором лежит конечная сумма денег, имеющихся в распоряжении компании. Известно, что дела в компании ведутся аккуратно, и в долги компания не влезает.

Сложность этой задачи достаточно высока, однако возможная вариативность решения позволяет получить некоторые баллы за эту задачу многим участникам.

Особенностью математической формализации является то, что мы должны выполнять действия (сложение и вычитание) не с числами, а с отрезками. Начинаем мы с одноточечного отрезка, содержащего лишь точку начальной суммы денег. Определим, как производятся действия.

<u>Приход.</u> Пусть перед операцией прихода интервальная оценка суммы на счету компании была $[a,b], a \leq b$, и пусть диапазон суммы операции есть $[c,d], c \leq d$. Тогда несложно понять, что оценка суммы на счету после проведения операции есть [a+c,b+d].

<u>Расход.</u> Пусть перед операцией расхода интервальная оценка суммы на счету компании была $[a,b], a\leqslant b$, и пусть диапазон суммы операции есть $[c,d], c\leqslant d$. Тогда несложно понять, что оценка суммы на счету после проведения операции есть [a-d,b-c]; в сравнении с операцией прихода из нижней границы оценки суммы на счету мы вычитаем верхнюю границу оценки суммы операции, и, наоборот, из верхней границы оценки суммы на счету мы вычитаем нижнюю границу оценки суммы операции. По условию задачи («фирма в долги не влезает») гарантируется, что верхняя граница новой суммы неотрицательна: $b-c\geqslant 0$. Однако может оказаться, что нижняя оценка стала отрицательной: a-d<0. В этом случае нижнюю оценку надо заменить нулём — отрицательных сумм на счету компании быть не может.

Запрос остатка. Пусть перед запросом остатка интервальная оценка суммы на счету компании была $[a,b], a \leqslant b$, и пусть диапазон результата запроса есть $[c,d], c \leqslant d$. Тогда несложно понять, что оценка суммы на счету после запроса становится равной пересечению отрезков [a,b] и [c,d]. Известно, что нижняя граница пересечения есть максимум нижних границ пересекаемых отрезков: $a' = \max\{a,c\}$, а верхняя — минимуму из верхних: $b' = \min\{b,d\}$. При этом, если a' > b', то исходные отрезки не пересекаются, однако такая ситуация невозможна по условию задачи — условием гарантируется, что последовательность операций корректна.

Таким образом решение состоит в поэлементном чтении списка операций и проделывания действий над отрезком оценки суммы согласно указанным правилам. Результат, получившийся в конце, и есть ответ.

Также следует обратить внимание на то, что ограничения на входные данные таковы, что промежуточные результаты и ответ могут выходить за границы типа short int, но обязательно укладываются в тип int.

Кроме того, в решении нужно производить чтение и обработку строковых данных, что также добавляет некоторые сложности при написании программы.

Идеи тестов:

- 1. Минимальный тест количество операций равно 0.
- 2. Начальная сумма равна нулю; в процессе вычислений ноль не затрагивается.
- 3. В процессе вычислений нижняя оценка суммы один раз становится отрицательной и требует коррекции.
- 4. В процессе вычислений нижняя оценка суммы несколько раз становится отрицательной и требует коррекции.
- 5. Ответ одноточечный отрезок.
- 6. В процессе вычислений оценка один раз становится одноточечным отрезком.
- 7. В процессе вычислений оценка несколько раз становится одноточечным отрезком.
- 8. В процессе вычислений оценка один раз становится одноточечным отрезком [0,0].
- 9. В процессе вычислений оценка несколько раз становится одноточечным отрезком [0,0].
- 10. Тест минимальный по числу категорий оценок.
- 11. Тест максимальный по числу категорий оценок.
- 12. Тест максимальный по числу операций.
- 13. Максимальный тест: максимальное число оценок, максимальное число операций.
- 14–20. Случайные тесты.

9.5. «Субнормальные числа». Назовём число k-субнормальным, если в его десятичной записи встречаются десятичные записи всех натуральных чисел из k цифр. Например, 1234567890 является 1-субнормальном числом, а 123456789 не является (нет записи числа 0). Определите, является ли заданное число k-субнормальным. Напомним, что ведущие нули в записи числа не допускаются.

С точки зрения членов программного комитета, задача представляется сложной, хотя в первую очередь сложность её техническая.

Допустимым является «лобовое» решение. Считываем число поэлементно в массив или в строку (в Паскале при этом, естественно, не удастся использовать стандартный тип string, поскольку длина строк, им представляемых, не превышает 255 символов). Затем начинаем перебирать все k-значные числа (то есть числа из диапазона от 10^{k-1} до 10^k-1 ; исключение -k=1, здесь диапазон простирается от 0 до 9). Каждое число переводим в массив или строку его цифр и проверяем, содержится ли этот массив/строка в массиве/строке, считанном из файла. При имеющихся ограничениях на входные данные аккуратная реализация этого алгоритма укладывается во временные рамки.

Хотя отметим, что увеличение размера длины строки до 10^6 или разрядности чисел до 5 выводило данный алгоритм из разряда допустимых.

Идеи тестов:

Поскольку данная задача не содержит глубоких идей, то для проверки предлагается 20 тестов. Первые восемь из них содержат субнормальные числа, остальные двенадцать — нет. Длина тестов варьируется от небольших до максимально возможных.

Фестиваль «Юные интеллектуалы Среднего Урала» Муниципальный этап Всероссийской олимпиады по информатике $2012-2013 \ {\rm учебный} \ {\rm год}$ $10 \ {\rm класc}$

Время выполнения задач — 4 часа Ограничение по времени — 2 секунды на тест Ограничение по памяти — 64 мегабайта

10.1. «Профилактика гриппа». На Зимнюю Компьютерную Школу выехало p преподавателей и s школьников. Но зима — очень опасное время, ведь постоянно случаются эпидемии гриппа. Для профилактики заболевания участники школы взяли с собой g стандартных головок чеснока, каждая из которых содержит 10 зубчиков. Для того, чтобы не заболеть, взрослый должен съедать в день два зубчика чеснока, а школьник — один. На сколько дней хватит взятых с собой запасов чеснока, чтобы ежедневно проводить профилактику гриппа для всего состава школы?

Формат ввода: В первой строке через пробел записано через пробел три целых числа p, s и q ($1 \le p, s \le 100, 0 \le q \le 1000$).

Формат вывода: Выведите единственное целое число — количество дней, на которое хватит взятого чеснока.

Пример

input.txt: output.txt:

1 10 5 4

10.2. «Передаточный механизм». Передаточный механизм представляет собой n резиновых валиков, зацепленных в ряд так, что каждый валик кроме первого и последнего сцеплен с двумя соседними: первый — со вторым, второй — с первым и третьим, третий — со вторым и четвёртым, . . . , n-й — с (n-1)-м. Известно, что k-й валик имеет радиус r_k . Начальный валик с номером 1 закрутили с угловой скоростью ω_1 . (Считаем, что если $\omega>0$, то валик вращается против часовой стрелки, а если $\omega<0$, то по часовой стрелке.) С какой угловой скоростью ω_n будет вращаться последний валик с номером n? Напомним, что если два валика с радиусами r' и r'' находятся в зацеплении, то их угловые скорости ω' и ω'' удовлетворяют соотношению $|\omega'| \cdot r' = |\omega''| \cdot r''$ и при этом имеют разные знаки.

Формат ввода: В первой строке через пробел записаны два числа: вещественное ω_1 — угловая скорость первого валика ($|\omega_1| \le 100$) — и целое n — количество валиков ($1 \le n \le 100$). Во второй строке через пробел записано n натуральных чисел r_k — радиусы валиков в том порядке, в котором они находятся в зацеплении ($1 \le r_k \le 1000$).

Формат вывода: Выведите единственное вещественное число, которое с точностью до трёх знаков после запятой совпадает с угловой скоростью ω_n последнего валика.

Пример

5 1 3

<u>Примечание к примеру:</u> Истинный ответ -5/3 — представлен числом, которое отличается от него менее, чем на $5 \cdot 10^{-4}$.

10.3. «Складываем треугольник». У Пети Торопыжкина завалялось несколько счётных палочек, длины которых известны. Он начал доставать их из коробки по четыре штуки и пытаться складывать треугольник из каждой очередной четвёрки палочек, используя все четыре. Четвёрка считается «хорошей», если из неё треугольник складывается, то есть из неё получается замкнутая трёхзвенная ломаная, ограничивающая треугольник с ненулевой площадью. Сколько «хороших» четвёрок у него получится?

Формат ввода: В первой строке записано число n, кратное 4 — количество палочек ($4\leqslant n\leqslant 10000$). Во второй строке через пробел записано n целых чисел — длины палочек в том порядке, в котором они уложены в коробочку и в каком их достаёт Петя. Каждая длина есть целое число от 1 до 1000.

Формат вывода: Выведите единственное целое число — количество четвёрок, из которых можно сложить треугольник.

Пример

1 2 3 4 1 10 10 100

Примечание к примеру: Из первых четырёх палочек треугольник складывается (например, в комбинации (1+2), 3, 4 — палочки с длинами 1 и 2 выкладываются в один отрезок прямой), а из второй четвёрки сложить треугольник нельзя — последняя палочка слишком длинная.

10.4. «Бипалиндромы». Назовём натуральное число k-бипалиндромом, если его записи в десятичной системе и в системе с основанием k являются палиндромами, то есть, одинаково читаются от начала к концу и от конца к началу. Например, число $373_{10}=565_8=1421_6$ является 8-бипалиндромом, но не является 6-бипалиндромом (нижний индекс после цифровой записи указывает основание системы счисления, в которой сделана данная запись). Для заданных чисел k и n определите, сколько натуральных чисел, не превосходящих n, являются k-бипалиндромами.

Формат ввода: Единственная входная строка содержит через два натуральных числа n и k ($1 \le n \le 10^6$, $2 \le k \le 36$, $k \ne 10$).

Формат вывода: Выдайте единственное число — количество k-бипалиндромов в диапазоне от 1 до n.

Пример

input.txt: output.txt:
5 2 3

10.5. «Побелка забора». Тому Сойеру нужно побелить забор, состоящий из n досок. Время покраски k-й доски известно и равно t_k . Одновременно Том может красить или одну доску, или две соседние доски (двумя кистями). Во втором случае время покраски равно наибольшему из времён покраски отдельных досок. Кроме того, пока предыдущая операция не будет завершена, Том не может приступать к следующей операции. За какое наименьшее время можно покрасить этот забор?

Формат ввода: В первой строке записано натуральное число $n \ (1 \le n \le 10000)$ — количество досок в заборе. Во второй строке через пробел перечислены величины $t_k \ (1 \le t_k \le 1000)$. Забор считается незамкнутым — нельзя одновременно красить первую и последнюю доски забора.

Формат вывода: Выведите единственное целое число — минимально возможное время покраски забора.

Пример

input.txt: output.txt:
3 8
1 7 5

Фестиваль «Юные интеллектуалы Среднего Урала» Муниципальный этап Всероссийской олимпиады по информатике 2012 – 2013 учебный год 10 класс. Разбор решений и идеи тестов

Далее в разборах используются обозначения типов: short int — 2-байтовый целый тип (integer в TurboPascal/BorlandPascal и int в BorlandC++), int — 4-байтовый целый тип (longint в TurboPascal/BorlandPascal/FreePascal, long int в BorlandC++, integer в Delphi/PascalABC и int в VisualC++/C++ Builder), int64 — 8-байтный целый (отсутствует в BorlandPascal и BorlandC++, но присутствует в системах программирования под Windows).

Задачи оцениваются по 100 баллов за каждую; стало быть, полная стоимость пакета — 500 баллов. В рамках одной задачи все тесты считать равноценными.

10.1. «Профилактика гриппа». На Зимнюю Компьютерную Школу выехало р преподавателей и s школьников. Но зима — очень опасное время, ведь постоянно случаются эпидемии гриппа. Для профилактики заболевания участники школы взяли с собой д стандартных головок чеснока, каждая из которых содержит 10 зубчиков. Для того, чтобы не заболеть, взрослый должен съедать в день два зубчика чеснока, а школьник — один. На сколько дней хватит взятых с собой запасов чеснока, чтобы ежедневно проводить профилактику гриппа для всего состава школы?

Данная задача является простой.

Дневная потребность Школы в чесноке вычисляется как s+2p зубчиков. Во взятых g головках содержится 10g зубчиков. Стало быть, искомое количество дней есть (10g) div (s+2p).

Идеи тестов:

Для решения предлагается 10 случайных тестов, включая минимальный и максимальный. Также имеется тест с недостаточным количеством чеснока, то есть когда запас чеснока меньше дневной потребности.

10.2. «Передаточный механизм». Передаточный механизм представляет собой п резиновых валиков, зацепленных в ряд так, что каждый валик кроме первого и последнего сцеплен с двумя соседними: первый — со вторым, второй — с первым и третьим, третий — со вторым и четвёртым, ..., n-й — c (n-1)-м. Известно, что k-й валик имеет радиус r_k . Начальный валик с номером 1 закрутили с угловой скоростью ω_1 . (Считаем, что если $\omega>0$, то валик вращается против часовой стрелки, а если $\omega<0$, то по часовой стрелке.) С какой угловой скоростью ω_n будет вращаться последний валик с номером n? Напомним, что если два валика с радиусами r' и r'' находятся в зацеплении, то их угловые скорости ω' и ω'' удовлетворяют соотношению $|\omega'| \cdot r' = |\omega''| \cdot r''$ и при этом имеют разные знаки.

Программный комитет считает данную задачу простой.

Решение задачи вполне прямолинейно: считывая радиус валика (кроме первого), тут же вычисляем его угловую скорость по указанной формуле. Главное — не забыть про смену знака угловой скорости при переходе к следующему валику.

Идеи тестов:

Для решения предлагается 10 случайных тестов, включая крайние случаи одного валика и 100 валиков. Среди тестов имеются тесты с положительными и отрицательными начальными и конечными угловыми скоростями.

10.3. «Складываем треугольник». У Пети Торопыжкина завалялось несколько счётных палочек, длины которых известны. Он начал доставать их из коробки по четыре штуки и пытаться складывать треугольник из каждой очередной четвёрки палочек, используя все четыре. Четвёрка считается «хорошей», если из неё треугольник складывается, то есть из неё получается замкнутая трёхзвенная ломаная, ограничивающая треугольник с ненулевой площадью. Сколько «хороших» четвёрок у него получится?

Программный комитет считает, что сложность этой задачи средняя.

Наиболее разумное решение: для каждой четвёрки перебирать пары палочек, которые будут выкладываться в отрезок прямой, вычислять длины трёх получившихся палочек и проверять для них неравенство треугольника. Не слишком сложный, но требующий некоторого размышления момент — это организация перебора пар. Проще всего, перебирать номер первой палочки от 1 до 3, а второй — от номера первой, увеличенного на 1, до 4.

Идеи тестов:

Для проверки предлагаются 20 случайных тестов, включающих крайние (n=4) и n=10000, а также случаи вырожденных треугольников.

10.4. «Бипалиндромы». Назовём натуральное число k-бипалиндромом, если его записи в десятичной системе u в системе c основанием k являются палиндромами, то есть, одинаково читаются от начала κ концу u от конца κ началу. Например, число $373_{10} = 565_8 = 1421_6$ является 8-бипалиндромом, но не является 6-бипалиндромом (нижний индекс после цифровой записи указывает основание системы счисления, в которой сделана данная запись). Для заданных чисел k u n определите, сколько натуральных чисел, не превосходящих n, являются k-бипалиндромами.

Данная задача позиционируется как имеющая среднюю сложность. В принципе, единственная идея, которая здесь требуется— это алгоритм перевода числа в систему счисления с заданным основанием.

Собственно, решение прямолинейно: перебираем числа от 1 до n и проверяем, является ли очередное число бипалиндромом. Для этого переводим его в спи-

сок цифр в десятичной записи и в записи с основанием k, после чего проверяем, являются ли палиндромами эти записи. Разумно записывать цифры в массив.

Идеи тестов:

Для проверки предлагаются 20 случайных тестов, включающих крайние (n=1 и $n=10^6$). Рассматриваются все значения k от 2 до 9 и некоторые большие 10.

10.5. «Побелка забора». Тому Сойеру нужно побелить забор, состоящий из п досок. Время покраски k-й доски известно и равно t_k . Одновременно Том может красить или одну доску, или две соседние доски (двумя кистями). Во втором случае время покраски равно наибольшему из времён покраски отдельных досок. Кроме того, пока предыдущая операция не будет завершена, Том не может приступать к следующей операции. За какое наименьшее время можно покрасить этот забор?

Данная задача, по мнению программного комитета, является сложно, так как привлекает идеи принципа динамического программирования.

«Лобовое» решение — перебирать все варианты покраски забора — уложится во временные рамки только для небольшого количества досок в заборе. Поэтому требуется более хитрое решение, которое может быть связано с принципом динамического программирования.

Создадим массив Т размера $(n+1)\times 2$ с индексами по первому измерению от 0 до n, а по второму — от 1 до 2. В ячейке T[i,1] будем хранить минимальное время, требующееся для покраски досок забора от 1-й до i-й, при условии, что i-я доска красилась в «индивидуальном» порядке. В ячейке T[i,2] будем хранить минимальное время, требующееся для покраски досок забора от 1-й до i-й, при условии, что i-я доска красилась в паре с (i-1).

Положим T[0,1] = 0, T[0,2] = 0 — если не покрашено ни одной доски, то на это надо потратить время, равное нулю. Также положим, что $T[1,1] = t_1$ и T[1,2] = Inf, где Inf — «бесконечность», время, заведомо большее ответа; например, эту константу можно положить равной сумме всех t_k , увеличенную на 1. Смысл последних присваиваний: в одиночку первую доску можно покрасить за время, требуемое на её покраску, а в паре с предыдущей её покрасить нельзя, так как нет предыдущей доски.

Затем начнём вычислять T[i,1] и T[i,2] для индексов i, бо́льших 1. Пересчёт отражает следующие соображения. Если очередная i-я доска красится в «индивидуальном» порядке, то минимальное время, за которое можно закончить покраску — это есть минимальное время покраски досок до (i-1)-й включительно плюс время покраски i-й доски: $T[i,1] = \min\{T[i-1,1],T[i-1,2]\} + t_i$. Если же мы красим текущую доску вместе с предыдущей, то время покраски есть минимальное время покраски досок до (i-2)-й включительно плюс максимум из времён покраски (i-1)-й и i-й досок:

 $T[i,1] = min\{T[i-2,1],T[i-2,2]\} + max\{t_{i-1},t_i\}.$

26

Соответственно, ответ — это минимум из величин T[n,1] и T[n,2].

- 1. Минимальный тест одна доска.
- 2. Минимальный тест две доски.
- 3. Тест, когда все доски надо красить парами (1-2, 3-4, и т.д.)
- 4. Достаточно много досок, их чётное количество.
- 5. Достаточно много досок, их нечётное количество.
- 6. Максимальный тест малые времена покраски досок, 10000 досок.
- 7. Максимальный тест большие времена покраски досок, 10000 досок.
- 8-20. Случайные тесты.

Фестиваль «Юные интеллектуалы Среднего Урала» Муниципальный этап Всероссийской олимпиады по информатике 2012-2013 учебный год

11 класс

Время выполнения задач — 4 часа Ограничение по времени — 2 секунды на тест Ограничение по памяти — 64 мегабайта

11.1. «Столбы». Когда Петя Торопыжкин ехал в поезде в Москву на Всероссийскую олимпиаду по информатике, он обнаружил, что все столбы, поддерживающие провода контактной сети, пронумерованы натуральными числами от 1 до 10000. Он наблюдал промежуток дороги, на котором стояли столбы с номерами от n до m. Его заинтересовало, а сколько столбов из этого диапазона (включая столбы с номерами n и m) имеют номер, кратный 3. Помогите ему, напишите программу, которая бы вычисляла это количество.

Формат ввода: В первой строке заданы через пробел два числа n и m ($1 \le n \le m \le 10000$).

Формат вывода: Выведите единственное число — количество чисел из диапазона от n до m включительно, делящихся на 3.

Пример

input.txt: output.txt:

1 10 3

11.2. «Идём в поход». Специализированный информатико-математический класс, в котором учится Петя Торопыжкин, собирается в поход. Петю назначили ответственным за продовольствие. У него имеется рюкзак с n карманами; грузоподъёмность k-го кармана равна a_k кг. Сколько банок тушёнки, массой m кг каждая, он сможет взять в поход?

Формат ввода: В первой строке через пробел записаны два целых числа m и n ($1 \le m \le 10, 1 \le n \le 100$). В следующей строке через пробел записаны n чисел a_k — грузоподъёмности карманов рюкзака ($1 \le a_k \le 500$).

Формат вывода: Выведите единственное целое число — количество банок тушёнки, которое Петя сможет погрузить в свой рюкзак.

Пример

input.txt: output.txt:

2 4 4

1 2 3 4

11.3. «Поливаем цветы». На клумбе растут n цветков в точках с координатами (x_i, y_i) $(1 \le i \le n; x_i, y_i \ne 0;$ все точки попарно различны). Поливал-ка представляет собой вращающуюся лейку, находящуюся в начале координат, в

каждый момент времени орошающую какой-то луч с началом там же в начале координат. В начале лейка поливает положительное направление оси Ox, затем начинает поворачиваться против часовой стрелки. Цветок с каким номером будет полит первым? Если два цветка находятся на одном луче, выходящем из начала координат, то первым будет полит тот, который расположен ближе к началу координат.

Формат ввода: В первой строке записано единственное число n — количество цветов ($1 \le n \le 10000$). В следующих n строках через пробел записаны пары целых чисел, по модулю не превосходящих 1000 — координаты растений.

Формат вывода: Выведите единственное целое число — номер цветка (в порядке, указанном во входном файле; отсчёт номеров начинается от 1), который будет полит первым.

Пример

11.4. «Составляем записку». Петя Торопыжкин собирается в субботу надолго задержаться в школе — идёт подготовка к командному соревнованию школьников по программированию. Он хочет известить об этом родителей, но очень необычным способом: он не желает писать записку от руки, а думает склеить её, вырезав целые слова из имеющейся газеты. Напишите программу, которая определит, имея текст записки и имеющийся газетный текст, удастся ли Пете подготовить свою записку.

Формат ввода: В первой строке через пробел записаны два натуральных числа: длина петиного сообщения и длина имеющегося газетного текста (в обоих случаях учтены и пробелы тоже). Во второй строке записано петино сообщение, в третьей строке — имеющийся газетный текст. Текст — последовательность слов, разделённых пробелами; слово — последовательность заглавных латинских букв; первый и последний символы текста являются буквами. Размер каждого текста — не более 10000 символов, включая пробелы; не более 1000 слов.

Формат вывода: В случае, когда сообщение составить можно, выведите сообщение из двух слов — GOOD NOTE. Если же Пете удастся составить своё послание, то выведите первое слово из петиной записки, которое невозможно получить из имеющегося текста.

Пример 1		Пример 2		Пример 3	
<pre>input.txt:</pre>	output.txt:	input.txt:	output.txt:	input.txt:	output.txt:
5 7	GOOD NOTE	5 7	C	5 7	A
A B C		A B C		A B A	
ABCD		ABDE		ABCD	

11.5. «Пропадающие числа». Квадрат $n \times n$ изначально заполнен натуральными числами, при этом каждое число лежит в диапазоне от 1 до n. Затем выполняется следующая процедура, если в каком-то столбце или строке (полной или неполной) имеется число k и ровно k одинаковых чисел, то все эти числа исчезают. Например, $3 \ 9 \ 2 \ 5 \ 3 \ 6 \to 9 \ 5 \ 6$; исчезла двойка и два числа 3. Если таких конструкций в столбце несколько, то одновременно исчезают все; при этом каждое число может участвовать в нескольких конструкциях. Например, $9 \ 5 \ 9 \ 2 \ 9 \ 4 \ 9 \ 6 \ 5 \ 9 \ 4 \to 6$; Исчезают конструкции $2 \ 5 \ 5, \ 2 \ 4 \ 4, \ 5 \ 9 \ 9 \ 9 \ 9$. После исчезновения оставшиеся числа «осыпаются» вниз. Затем процедура повторяется на новой конфигурации до тех пор, пока в получившейся конфигурации не останется конструкций, которые можно удалить по указанному правилу. По заданному начальному заполнения квадрата опишите его конечное состояние.

Формат ввода: В первой строке записано единственное целое число $n\ (1\leqslant 50)$ — размер квадрата. В следующих n строках записано через пробел по n чисел, каждое в диапазоне от 1 до n — элементы начальной таблицы.

Формат вывода: Выведите n строк по n объектов, разделённых пробелами — элементы итогового состояния таблицы. Если в клетке стоит число, следует его вывести; если клетка пустая, выведите символ «.» (точка).

Пример

input.txt:	output.txt
5	
1 2 1 2 5	
5 4 4 3 5	
1 5 4 4 5	5
3 3 3 5 2	4 3 5
3 3 5 4 5	

<u>Примечание к примеру:</u> в случае больших n вывод может быть не таким «красивым» из-за сдвижки столбцов при выводе многозначных чисел.

Последовательность изменений таблицы:

1	2	1	2	5		<u>1</u>	$\underline{2}$	<u>1</u>	$\underline{2}$	<u>5</u>																	
5	4	4	3	5		5	4	4	3	5						5						5					
1	5	4	4	5	\rightarrow	1	<u>5</u>	4	<u>4</u>	5	\rightarrow					5	\rightarrow					5	\rightarrow				
3	3	3	5	2		3	3	3	5	2			4	4	3	2			<u>4</u>	<u>4</u>	3	$\underline{2}$					5
3	3	5	4	5		3	3	<u>5</u>	4	5		3	5	4	5	5		3	<u>5</u>	4	<u>5</u>	<u>5</u>			4	3	5

Жирным шрифтом с подчёркивание отмечены числа, которые исчезнут на очередной итерации.

На первой итерации происходят следующие исчезновения: в первой строке: $1\ 1\ 2\ (2\ \text{раза}),\ 1\ 5;$ в третьей строке $1\ 5;$ в четвёртой строке: $3\ 3\ 3;$ во втором столбце: $2\ 3\ 3;$ в первом столбце: $1\ 1\ 5;$ в третьем столбце: $1\ 3,\ 1\ 5;$ в четвёртом столбце: $2\ 4\ 4.$

На второй итерации происходят следующие исчезновения: в первом столбце: $1\ 3;$ в четвёртой строке: $1\ 2,\ 4\ 4\ 2;$ в пятой строке: $3\ 5\ 5$ 5.

Фестиваль «Юные интеллектуалы Среднего Урала» Муниципальный этап Всероссийской олимпиады по информатике 2012—2013 учебный год 11 класс. Разбор решений и идеи тестов

Далее в разборах используются обозначения типов: short int — 2-байтовый целый тип (integer в TurboPascal/BorlandPascal и int в BorlandC++), int — 4-байтовый целый тип (longint в TurboPascal/BorlandPascal/FreePascal, long int в BorlandC++, integer в Delphi/PascalABC и int в VisualC++/C++ Builder), int64 — 8-байтный целый (отсутствует в BorlandPascal и BorlandC++, но присутствует в системах программирования под Windows).

Задачи оцениваются по 100 баллов за каждую; стало быть, полная стоимость пакета — 500 баллов. В рамках одной задачи все тесты считать равноценными.

11.1. «Столбы». Когда Петя Торопыжкин ехал в поезде в Москву на Всероссийскую олимпиаду по информатике, он обнаружил, что все столбы, поддерживающие провода контактной сети, пронумерованы натуральными числами от 1 до 10000. Он наблюдал промежуток дороги, на котором стояли столбы с номерами от п до т. Его заинтересовало, а сколько столбов из этого диапазона (включая столбы с номерами п и т) имеют номер, кратный 3. Помогите ему, напишите программу, которая бы вычисляла это количество.

Данная задача является весьма лёгкой.

Искомое число — количество чисел, кратных 3, лежащих в отрезке [n,m] — есть разность количеств чисел, кратных 3, на отрезках [1,m] и [1,n-1]. Количество же чисел, кратных 3, лежащих на отрезке от 1 до k есть k div 3.

Впрочем, подойдёт и «лобовое» решение, перебирающее числа от n до m и непосредственно считающее количество чисел, кратных 3.

Идеи тестов:

Для проверки предлагаются 10 случайных тестов.

11.2. «Идём в поход». Специализированный информатико-математический класс, в котором учится Петя Торопыжкин, собирается в поход. Петю назначили ответственным за продовольствие. У него имеется рюкзак с n карманами; грузоподъёмность k-го кармана равна a_k кг. Сколько банок тушёнки, массой m кг кажедая, он сможет взять в поход?

Программному комитету данная задача представляется лёгкой.

Для нахождения ответа нужно просуммировать частные от деления размеров карманов нацело на вес одной банки тушёнки.

Идеи тестов:

Для проверки предлагаются 10 случайных тестов.

11.3. «Поливаем цветы». На клумбе растут п цветков в точках с координатами (x_i, y_i) $(1 \le i \le n; x_i, y_i \ne 0;$ все точки попарно различны). Поливалка представляет собой вращающуюся лейку, находящуюся в начале координат, в каждый момент времени орошающую какой-то луч с началом там же в начале координат. В начале лейка поливает положительное направление оси Ox, затем начинает поворачиваться против часовой стрелки. Цветок с каким номером будет полит первым? Если два цветка находятся на одном луче, выходящем из начала координат, то первым будет полит тот, который расположен ближе к началу координат.

Задача, по мнению программного комитета, имеет среднюю сложность.

Решение — упорядочить все точки по углу, отсчитываемому от оси Ox против часовой стрелки, а при равенстве угла — по величине радиус-вектора. После сортировки надо взять первый элемент в такой последовательности. Размер входных данных требует использовать «хорошую» сортировку, сложность которой менее, чем квадратичная (например, QuickSort, HeapSort или MergeSort).

Также не вполне тривиальным может быть вычисление угла. В стандартной математической библиотеки языка $\mathrm{Cu}/\mathrm{Cu}++$ имеется функция $\mathrm{atan2}$, которая вычисляет угол радиус-вектора точки, выдавая его в диапазоне $[-\pi, +\pi]$. В Паскале надо воспользоваться функцией atan , а потом корректно перевести угол в соответствующий квадрант, используя знаки координат точки.

Идеи тестов:

- 1. Одна точка.
- 2. Две точки.
- 3–5. Одна точка в первом квадранте, много во втором, третьем, четвёртом.
- 6–8. Одна точка во втором квадранте, много в первом, третьем, четвёртом.
- 9–11. Одна точка в третьем квадранте, много в первом, втором, четвёртом.
- 12–14. Одна точка в четвёртом квадранте, много в первом, втором, третьем.
 - 15. Максимальный тест.
- 16-20. Случайные тесты.
- 11.4. «Составляем записку». Петя Торопыжкин собирается в субботу надолго задержаться в школе идёт подготовка к командному соревнованию школьников по программированию. Он хочет известить об этом родителей, но очень необычным способом: он не желает писать записку от руки, а думает склеить её, вырезав целые слова из имеющейся газеты. Напишите программу, которая определит, имея текст записки и имеющийся газетный текст, удастся ли Пете подготовить свою записку.

Данная задача имеет сложность выше среднего из-за того, что требуется нетривиальная обработка строк.

Собственно идея решения крайне несложна: посчитать количество каждого в петиной записке, в газетном тексте, а после проверить, что газетный текст содержит каждое слово из петиной записки в не меньшем количестве. Если это так то записку составить можно, в противном случае — нельзя. Если записку составить нельзя, то надо определить слово из записки, которое не составится. Для этого для каждого слова из записки надо запоминать в каких позициях оно стоит в записке. Соответственно, для каждого «плохого» слова, находить в какой позиции стоит его копия, которую невозможно получить из текста.

В принципе, рассматриваемая задача решаема и на уровне «тривиальных» знаний, но наиболее удобно использовать здесь структуру данных «ассоциативный массив» или, как её ещё называют, «словарь». По сути, это массив, в котором элементы индексированы не целыми числами, а объектами произвольной природы, на которых определено соотношение «больше/меньше». Ключом (индексом) такого массива разумно сделать строку-слово, а элемент — массив позиций этого слова в тексте.

Таким образом, однотипно обрабатываем оба текста, получая соответствующие структуры данных, а потом перебираем слова в структуре петиной записки, игнорируя «хорошие» слова (то есть, слова, которые есть в структуре газетного текста и длина массива позиций которых не короче, чем длина массива для слова из записки), а для «плохих» слов ищем слово с минимальным значением среди тех позиций, которые не обеспечены словами из текста. Например:

из записки: «ABC» \to [1,5,10,15,23], из текста: «ABC» \to [3,18]

В записке слова на позициях 1 и 5 покрываются вырезками из текста, а на позиции 10 и далее – нет. Поэтому «ABC» — «плохое» слово и его номер — 10.

Идеи тестов:

- 1. Размеры записки и текста небольшие. В записке и в тексте по одному слову, ситуация хорошая.
- 2. Размеры записки и текста небольшие. В записке и в тексте по одному слову, ситуация плохая.
- 3. Размеры записки и текста небольшие. В записке одно слово, в тексте много, ситуация хорошая.
- 4. Размеры записки и текста небольшие. В записке одно слово, в тексте много, ситуация плохая.
- 5. Размеры записки и текста небольшие. В записке и в тексте много слов, «плохое» слово одно, оно отсутствует в тексте.
- 6. Размеры записки и текста небольшие. В записке и в тексте много слов, «плохое» слово одно, оно присутствует в тексте в недостаточном количестве
- 7–10. Повторяют тесты 3–6, но большими длинами текста и малой длиной записки.
- 11-14. Повторяют тесты 3-6, но большими длинами текста и записки.

- 15. Максимальный тест, ситуация хорошая.
- 16. Максимальный тест, ситуация плохая.
- 17–20. Случайные тесты с плохой ситуацией.

11.5. «Пропадающие числа». Квадрат $n \times n$ изначально заполнен натуральными числами, при этом каждое число лежит в диапазоне от 1 до n. Затем выполняется следующая процедура, если в каком-то столбце или строке (полной или неполной) имеется число k и ровно k одинаковых чисел, то все эти числа исчезают. Например, $3 \ 9 \ 2 \ 5 \ 3 \ 6 \to 9 \ 5 \ 6$; исчезла двойка и два числа 3. Если таких конструкций в столбце несколько, то одновременно исчезают все; при этом каждое число может участвовать в нескольких конструкциях. Например, $9 \ 5 \ 9 \ 2 \ 9 \ 4 \ 9 \ 6 \ 5 \ 9 \ 4 \to 6$; Исчезают конструкции $2 \ 5 \ 5, \ 2 \ 4 \ 4, \ 5 \ 9 \ 9 \ 9 \ 9$. После исчезновения оставшиеся числа «осыпаются» вниз. Затем процедура повторяется на новой конфигурации до тех пор, пока в получившейся конфигурации не останется конструкций, которые можно удалить по указанному правилу. По заданному начальному заполнения квадрата опишите его конечное состояние.

Данная задача относится к классу «симуляторов»: в ней требуется аккуратно реализовать указанную последовательность действий. Сложность этой задачи достаточно высока, поскольку требуется достаточно аккуратная техника для её реализации.

По мнению программного комитета, процесс преобразования поля должен состоять из двух частей. В первой части производится помечивание тех чисел, которые должны исчезнуть. Во второй части производится собственно преобразование поля: числа убираются и переставляются так, чтобы произошло их «осыпание».

Конкретные реализации процедур и используемых структур данных могут быть весьма различными. Предлагаемая реализация содержит следующие идеи.

Ячейка поля представляется записью, одно поле которой содержит число, стоящее в этой ячейке. Если ячейка пустая, то кодируем эту ситуацию, записывая нуль. Второе поле — логическое — указывает, будет ли это число исчезать на очередной итерации.

Перед обработкой поля делается проход по нему, и все логические флаги сбрасываются в false. При обработке каждая строка и каждый столбец проходятся дважды. На первом проходе собирается информация о числах, на втором происходит помечивание чисел, которые будут удалены.

На первом проходе производятся следующие конкретные действия:

- имеется массив целых, в котором копятся количества отдельных чисел; перед проходом он заполняется нулями;
- имеется логический массив, в котором копятся флаги наличия тех или иных чисел; перед проходом он сбрасывается в false;
- соответственно, при считывании числа из очередной ячейки, если оно не ноль, то добавляется единичка к соответствующей ячейке первого массива, и соответствующая ячейка второго массива выставляется в true.

После того, как первый проход по строке/столбцу закончен, формируется третий логический массив, ячейки которого указывают, будут ли конкретные числа помечиваться на исчезновение по итогам обработки этой строки/столбца. Вначале он скидывается в false. Затем перебираем числа от 1 до n; если количество чисел, равных какому-то значению k равно q_k (это выясняется из первого массива) и имеется число q_k (это выясняется из второго массива), то числа k и q_k будут исчезать, соответствующие ячейки третьего массива выставляются в true.

Когда третий массив сформирован, производится второй проход по строке/столбцу и помечивание чисел на удаление: если в очередной ячейке стоит ненулевое число и соответствующая ячейка третьего массива выставлена в true, то ячейка помечивается на удаление.

После того, как все поле помечено, производится удаление чисел и перестановка оставшихся. Для этого делается проход по каждому столбцу. Считаем, что нумерация строк идёт снизу вверх: нижняя строка имеет номер 1, верхняя — номер n. Заводится два счётчика: счётчик текущей обрабатываемой строки и счётчик последней занятой строки в данном столбце. Вначале оба счётчика равны нулю.

Цикл идёт до тех пор, пока первый счётчик не превзойдёт n или пока ячейка, им адресуемая, не окажется пустой (дальше вверх могут быть только пустые ячейки. их обрабатывать не надо). Если очередная ячейка содержит число, которое должно исчезнуть, заполняем ячейку нулём и увеличиваем первый счётчик — переходим к следующей ячейке. Если очередная ячейка содержит число, которое не исчезает, то

- а) записываем число из ячейки во вспомогательную переменную;
- б) обнуляем ячейку;
- в) увеличиваем второй счётчик;
- г) записываем число из вспомогательной переменной в ячейку, указываемую вторым счётчиком;
- д) увеличиваем первый счётчик переходим к следующей ячейке.

Такова процедура одной итерации. Для проверки необходимости окончания итераций заводим флаг, который перед обработкой поля сбрасывается в false, а при удалении какого-либо числа выставляется в true. Таким образом, цикл по итерациям повторяется, пока этот флаг истинен.

- 1. Минимальный тест: n = 1.
- 2-5. Малые тесты: n = 2.
- 6. n = 10, все поле заполнено числом 10.
- 7. n=10, каждая строка заполнена случайно перемешанной последовательностью 1 2 2 3 3 3 4 4 4 4.
- 8–20. Случайные тесты, включая три максимальных (n=50).