

Индивидуальные варианты заданий к лабораторной работе «Программирование на языке C++ с использованием структур и перечислений»

Цель работы

Изучить особенности написания программ на языке C++ с использованием структур и перечислений.

Указания к выполнению работы

Для всех индивидуальных вариантов, имеющих ссылку на тестирующую систему, обязательным условием является успешное выполнение всех тестов¹.

При решении задач 1-2 необходимо определить пользовательский структурированный тип данных и описать как минимум по одной функции для каждой задачи, использующей в качестве параметра пользовательский структурированный тип.

При решении задачи 2 необходимо определить массив данных структурированного типа (можно использовать вектор), заполнить его значениями, введенными пользователем и обработать данные из массива соответствующим образом. При решении задач, требующих удаления или добавления структурированных объектов, необходимо перераспределить память под массив структурированного типа.

При решении задачи 3 необходимо определить перечисляемый тип (не менее чем из 3-х элементов), описать функции преобразования строки в заданный перечисляемый тип и обратно. Продемонстрировать работу функций на примере нескольких (более 2-х) наборов строковых и целочисленных данных, вводимых пользователем.

Индивидуальные варианты заданий

Задача 1

1. Описать тип *Price* - с полями целого типа *rub* (рубли), *cop* (копейки) и функцию *incPrice(p, t)*, с параметром *p* типа *Price* и целочисленным *t*, которая вычисляет произведение цены на положительное целое число *t*. Аргумент *p* является входным и выходным параметром. Применить функцию *incPrice* к пяти данным ценам.
2. Описать тип *Price* - с полями целого типа *rub* (рубли), *cop* (копейки) и функцию *priceCmp(p1, p2)* целого типа, с параметром *p1* и *p2* типа *Price*, которая

¹ В этом случае прочие указания к выполнению можно не выполнять

сравнивает цены $p1$ и $p2$ и возвращает 0, если они совпадают, -1, если $p1 < p2$ и 1, если $p1 > p2$. Применить функцию *priceCmp* для пяти пар цен.

3. Описать тип *Complex* - с полями вещественного типа *re* (действительная часть), *im* (мнимая часть) и функцию *mulComplex(c1, c2)*, с параметром *c1* и *c2* типа *Complex*, которая вычисляет и возвращает произведение двух комплексных чисел. Применить функцию *subComplex* для вычисления произведения пяти заданных комплексных чисел.

4. Описать тип *Frac* - с полями целого типа *num* (числитель), *denom* (знаменатель) (для представления правильной обыкновенной дроби) и функцию *sumFrac(f1, f2)*, с параметром *f1* и *f2* типа *Frac*, которая вычисляет и возвращает сумму двух дробей. Применить функцию *sumFrac* для вычисления суммы пяти данных дробей.

5. Описать тип *Frac* - с полями целого типа *num* (числитель), *denom* (знаменатель) (для представления правильной дроби) и функцию *divFrac(f1, f2)*, с параметром *f1* и *f2* типа *Frac*, которая вычисляет и возвращает частное от деления двух дробей. Применить функцию *divFrac* для вычисления отношения пяти пар дробей.

6. Описать тип *Date* - с полями целого типа *day* (день), *month* (месяц) и *year* (год) - и функцию *leapYear(d)* логического типа с параметром типа *Date*, которая возвращает *true*, если год в дате *d* является високосным, и *false* в противном случае. Вывести значение функции *leapYear* для пяти данных дат (предполагается, что все даты являются правильными).

7. Описать тип *Polar* - с полями вещественного типа *r* (радиус), *angle* (угол) и функцию *sumPolar(p1, p2)*, с параметром *p1* и *p2* типа *Polar*, которая вычисляет и возвращает полярную координату путем сложения соответствующих значений переданных параметров. Применить функцию *sumPolar* для вычисления суммы пяти данных координат.

8. Описать тип *Polar* - с полями вещественного типа *r* (радиус), *angle* (угол). Описать тип *Point* - с полями вещественного типа *x* и *y* (координаты точки на плоскости). Описать функцию *polarToRect(p)*, с параметром *p* типа *Polar*, которая преобразует координаты точки, записанные в полярной системе координат в запись в прямоугольной (декартовой) системе координат и возвращает точку типа *Point*. Применить функцию *polarToRect* к пяти заданным точкам в полярной системе координат.

9. Описать тип *Time* - с полями целого типа *h* (часы), *m* (минуты), *s* (секунды) и функцию *timeTo12(t)*, с параметром *t* типа *Time*, которая преобразует время суток из 24-часового к 12-часовому формату. Аргумент *t* является входным и выходным параметром.

Применить функцию *timeTo12* для пяти временных параметров, заданных в 24-часовом формате.

10. Описать тип *Time* - с полями целого типа *h* (часы), *m* (минуты), *s* (секунды) и функцию *timeCmp(t1, t2)* целого типа, с параметром *t1* и *t2* типа *Time*, которая сравнивает время *t1* и *t2* и возвращает 0, если время совпадает, -1, если $t1 < t2$ и 1, если $t1 > t2$. Применить функцию *timeCmp* для пяти пар временных интервалов.

11. Описать тип *Date* - с полями целого типа *day* (день), *month* (месяц) и *year* (год). Описать функцию *leapYear(d)* логического типа с параметром типа *Date*, которая возвращает *true*, если год в дате *d* является високосным, и *false* в противном случае. Описать функцию *daysInMonth(d)* целого типа с параметром типа *Date*, которая возвращает количество дней для месяца, указанного в дате *d*. Вывести значение функции *daysInMonth* для пяти данных дат (предполагается, что все даты являются правильными).

12. Описать тип *Point* - с полями вещественного типа *x* и *y* (координаты точки на плоскости) - и функцию *leng(a, b)* вещественного типа, находящую длину отрезка *AB* на плоскости по координатам его концов (*a* и *b* - параметры типа *Point*). Описать тип *Triangle* - с полями *a*, *b*, *c* типа *Point* (вершины треугольника) - и функцию *perim(t)* вещественного типа, находящую периметр треугольника *t* (*t* - параметр типа *Triangle*). С помощью этой функции найти периметры треугольников *ABC*, *ABD*, *ACD*, если даны координаты точек *A*, *B*, *C*, *D*.

13. Описать тип *Circle* - с полями целого типа *x* (*x* - координата центра окружности), *y* (*y* - координата центра окружности), *r* (радиус окружности). Описать тип *Point* - с полями вещественного типа *x* и *y* (координаты точки на плоскости). Описать функцию *circleXY(c, px, py)*, с параметром *c* типа *Circle*, параметрами *px* и *py* типа *Point*, которая находит точки пересечения окружности с осями *x* и *y*. Считается, что окружность не проходит через начало координат. Аргументы *px* и *py* являются выходными параметрами. Применить функцию *circleXY* для пяти разных окружностей.

14. Описать тип *QEquation* - с полями вещественного типа *a* (коэффициент при степени 2), *b* (коэффициент при степени 1) и *c* (свободный член) для представления квадратного уравнения. Описать функцию *qEqSolve(QE, x1, x2)* логического типа, с параметром *qe* типа *QEquation* и *x1*, *x2* вещественного типа, которая находит и возвращает рациональные корни квадратного уравнения. В случае отсутствия рациональных корней функция возвращает *false* (при наличии решения - *true*). Аргументы *x1* и *x2* являются выходными параметрами. Применить функцию *qEqSolve* для решения квадратных уравнений, заданных всеми возможными сочетаниями коэффициентов *a*, *b* и *c*.

15. Описать тип *Bit* - с полем *len* (число разрядов) целого типа и полем *num* (набор бит) – строка, состоящая из нулей и единиц и функцию *bitAnd(b1, b2)*, с параметром *b1* и *b2* типа *Bit*, которая вычисляет и возвращает результат выполнения операции поразрядного «и» к двум числам *b1* и *b2*. Предполагается, что в функцию посылаются только числа с одинаковым числом разрядов. Применить функцию *bitAnd* для вычисления поразрядного «и» для пяти данных чисел.

16. Описать тип *Bit* - с полем *len* (число разрядов) целого типа и полем *num* (набор бит) – строка, состоящая из нулей и единиц и функцию *bitLSShift(x)*, с параметром *b* типа *Bit*, которая сдвигает биты числа влево с дополнением нулями и потерей старшего бита. Аргумент *b* является входным и выходным параметром. Применить функцию *bitLSShift* для сдвига влево числа *b* на *x* разрядов.

17. Программа получает на вход набор точек на плоскости. Сначала задано количество точек *n*, затем идет последовательность из *n* строк, каждая из которых содержит два числа: координаты точки. Величина *n* не превосходит 100, все исходные координаты – целые числа, не превосходящие 1000 по абсолютной величине. Выведите координаты точки, наиболее удаленной от начала координат.

18. Выведите координаты центра тяжести данного множества точек. Для этого создайте структуру *Point* и сохраните исходные данные в массиве структур *Point*. Программа получает на вход набор точек на плоскости. Сначала задано количество точек *n*, затем идет последовательность из *n* строк, каждая из которых содержит два числа: координаты точки. Величина *n* не превосходит 100, все исходные координаты – целые числа, не превосходящие 1000. Выведите координаты центра тяжести данного множества точек. Ответ необходимо вывести с точностью в 15 значащих цифр.

19. Выведите диаметр данного множества – максимальное расстояние между любыми двумя точками. Создайте структуру *Point* и сохраните исходные данные в массиве структур *Point*. Программа получает на вход набор точек на плоскости. Сначала задано количество точек *n*, затем идет последовательность из *n* строк, каждая из которых содержит два числа: координаты точки. Величина *n* не превосходит 100, все исходные координаты – целые числа, не превосходящие 1000. Необходимо вывести диаметр данного множества с точностью в 15 значащих цифр.

20. Выведите все исходные точки в порядке возрастания их расстояний от начала координат. Создайте структуру *Point* и сохраните исходные данные в массиве структур *Point*. Программа получает на вход набор точек на плоскости. Сначала задано количество точек *n*, затем идет последовательность из *n* строк, каждая из которых содержит два числа: координаты точки. Величина *n* не превосходит 100, все исходные

координаты – целые числа, не превосходящие 1000. Программа выводит только координаты точек, их количество выводить не надо.

21. Описать тип *Date* - с полями целого типа *day* (день), *month* (месяц) и *year* (год). Описать функцию *leapYear(d)* логического типа с параметром типа *Date*, которая возвращает *true*, если год в дате *d* является високосным, и *false* в противном случае. Описать функцию *daysInMonth(d)* целого типа с параметром типа *Date*, которая возвращает количество дней для месяца, указанного в дате *d*. Описать функцию *checkDate(d)* целого типа с параметром типа *Date*, которая проверяет правильность даты, указанной в параметре *d*. Если дата *d* является правильной, то функция возвращает 0; если в дате указан неверный номер месяца, то функция возвращает 1; если в дате указан неверный день для данного месяца, то возвращается 2. Описать функцию *nextDate(d)* с параметром типа *Date*, которая преобразует дату *d* к следующей дате (если дата *d* является неправильной, то она не изменяется). Аргумент *d* является входным и выходным параметром. Применить функцию *nextDate* к пяти данным датам.

22. Среди исходных точек найдите три, образующие треугольник с максимальным периметром. Выведите данный периметр. Описать тип *Point* - с полями вещественного типа *x* и *y* (координаты точки на плоскости) - и функцию *leng(a, b)* вещественного типа, находящую длину отрезка *AB* на плоскости по координатам его концов (*a* и *b* - параметры типа *Point*). Описать тип *Triangle* - с полями *a, b, c* типа *Point* (вершины треугольника) - и функцию *perim(t)* вещественного типа, находящую периметр треугольника *t* (*t* - параметр типа *Triangle*). Программа получает на вход набор точек на плоскости. Сначала задано количество точек *n* ($2 < n < 101$), затем идет последовательность из *n* строк, каждая из которых содержит два числа: координаты точки. Все исходные координаты – целые числа, не превосходящие 1000. Необходимо вывести найденный периметр с точностью до 15 значащих цифр.

23. Среди исходных точек найдите три, образующие треугольник максимальной площади. Выведите данную площадь. Описать тип *Point* - с полями вещественного типа *x* и *y* (координаты точки на плоскости) - и функцию *leng(a, b)* вещественного типа, находящую длину отрезка *AB* на плоскости по координатам его концов (*a* и *b* - параметры типа *Point*). Описать тип *Triangle* - с полями *a, b, c* типа *Point* (вершины треугольника) - и функцию *perim(t)* вещественного типа, находящую периметр треугольника *t* (*t* - параметр типа *Triangle*). Описать функцию *area(t)* вещественного типа, находящую площадь треугольника *t* (*t* - параметр типа *Triangle*) по формуле Герона. Программа получает на вход набор точек на плоскости. Сначала задано количество точек *n* ($2 < n < 101$), затем

идет последовательность из n строк, каждая из которых содержит два числа: координаты точки. Все исходные координаты – целые числа, не превосходящие 1000. Необходимо вывести найденную площадь с точностью в 15 значащих цифр.

24. Описать тип *Point* – с полями вещественного типа x и y (координаты точки на плоскости) – и функцию $leng(a, b)$ вещественного типа, находящую длину отрезка AB на плоскости по координатам его концов (a и b – параметры типа *Point*). Описать тип *Triangle* – с полями a, b, c типа *Point* (вершины треугольника) – и функцию $perim(t)$ вещественного типа, находящую периметр треугольника t (t – параметр типа *Triangle*). Описать функцию $area(t)$ вещественного типа, находящую площадь треугольника t (t – параметр типа *Triangle*) по формуле Герона. Описать функцию $dist(p, a, b)$ вещественного типа (p, a, b – параметры типа *Point*), находящую расстояние $dist(p, ab)$ от точки P до прямой AB . Описать функцию $altitudes(t, h1, h2, h3)$, находящую высоты $h1, h2, h3$ треугольника t (t – входной параметр типа *Triangle*, $h1, h2, h3$ – выходные вещественные параметры), проведенные соответственно из вершин $t.a, t.b, t.c$. С помощью этой функции найти высоты треугольников ABC, ABD, ACD , если даны координаты точек A, B, C, D .

25. Описать тип *Rectangle* – с полями целого типа x и y (координаты левого верхнего угла прямоугольника), a и b (длина и ширина прямоугольника) для задания прямоугольника со сторонами a и b параллельными осям x и y соответственно. Описать функцию $isIntersect(r1, r2)$ логического типа, с аргументами $r1$ и $r2$ типа *Rectangle*, возвращающую *true*, если прямоугольники $r1$ и $r2$ имеют общие точки и *false* в противном случае. Описать функцию $intersectRect(r1, r2)$ типа *Rectangle*, находящую и возвращающую прямоугольник, являющийся пересечением прямоугольников $r1$ и $r2$, или возвращающей прямоугольник с нулевыми сторонами и координатами, если $r1$ и $r2$ не имеют общих точек.

26. Описать тип *Point* – с полями вещественного типа x и y (координаты точки на плоскости) – и функцию $leng(a, b)$ вещественного типа, находящую длину отрезка AB на плоскости по координатам его концов (a и b – параметры типа *Point*). Описать тип *Triangle* – с полями a, b, c типа *Point* (вершины треугольника). Описать функцию $exist(t)$ логического типа, проверяющую возможность существования треугольника t , функцию $perim(t)$ вещественного типа, находящую периметр треугольника t , функцию $area(t)$ вещественного типа, находящую площадь треугольника t по формуле Герона (t – параметр типа *Triangle*). Описать функцию $areaN(p, n)$ вещественного типа, находящую площадь выпуклого n -угольника, вершины которого (в порядке их обхода) передаются в массиве p размера n (> 2) с элементами типа *Point*. С помощью этой функции найти площади трех многоугольников, если дано число их сторон и координаты их вершин.

27. Описать тип *Solve* - с полями вещественного типа x и y (пара значений). Описать тип *Equation* - с полями вещественного типа: a (коэффициент при x), b (коэффициент при y) и z (свободный член) для представления линейного уравнения с двумя неизвестными $a \cdot x + b \cdot y = z$. Описать функцию *sysSolve*($e1$, $e2$) типа *Solve* с параметрами $e1$, $e2$ типа *Equation* для решения системы из двух линейных уравнений с двумя неизвестными и возвращающую пару значений x и y . С помощью этой функции найти решения пяти пар заданных линейных уравнений.

28. Информация о расписании занятий содержит записи: день недели, время начала и конца пары, название предмета, фамилия преподавателя. Дан массив из n «занятий» (n - целое, положительное). В массиве названия предметов и фамилии преподавателей могут повторяться, причем один и тот же предмет могут вести разные преподаватели. Вывести полную информацию о занятиях, относящихся к заданному предмету, упорядоченную по фамилии преподавателей.

29. Описать тип *Rectangle* - с полями целого типа x и y (координаты левого верхнего угла прямоугольника), a и b (длина и ширина прямоугольника) для задания прямоугольника со сторонами a и b не параллельными осям x и y . Описать функцию *isIntersect*($r1$, $r2$) логического типа, с аргументами $r1$ и $r2$ типа *Rectangle*, возвращающую *true*, если прямоугольники $r1$ и $r2$ имеют общие точки и *false* в противном случае. Описать функцию *intersectRect*($r1$, $r2$) типа *Rectangle*, находящую и возвращающую прямоугольник, являющийся пересечением прямоугольников $r1$ и $r2$, или возвращающей прямоугольник с нулевыми сторонами и координатами, если $r1$ и $r2$ не имеют общих точек.

30. Задан массив n квадратных трехчленов, где каждый трехчлен имеет комплексные коэффициенты. Сформировать массив корней трехчленов. Сформировать массив значений трехчленов в точке $x = c + di$. Квадратный трехчлен необходимо представить в виде структурированного типа данных, поля которого хранят коэффициенты трехчлена.

31. В одном из больших городов нашей страны хорошо развит рынок сотовой связи. На рынке присутствуют несколько операторов, каждому из которых принадлежит некоторое количество базовых станций, с помощью которых организуется связь. Каждая базовая станция характеризуется своими координатами (для простоты размерами базовых станций пренебрегаем) и радиусом надежной связи (если абонент находится на расстоянии, не превосходящем этот радиус, от базовой станции, то она может использоваться для работы с ним - передачи ему сигналов и приема сигналов от него). Известный производитель сотовых телефонов *Mokea* планирует оснастить свою новую

модель сотового телефона функцией определения базовых станций, с которыми может работать абонент. Вам же предстоит написать программу, которая по местоположению абонента для каждого оператора определит, сколько базовых станций этого оператора могут работать с абонентом.

Входные данные - первая строка содержит целое число n ($1 \leq n \leq 10000$) - количество базовых станций в городе. Далее идут описания этих базовых станций. Каждое описание занимает две строки. На первой расположено название оператора сотовой связи, которому принадлежит эта базовая станция, а на второй - три целых числа x, y, r ($-10000 \leq x, y \leq 10000, 1 \leq r \leq 10000$) - соответственно ее координаты и радиус надежной связи. Последняя строка входного файла содержит два целых числа x_a, y_a ($-10000 \leq x_a, y_a \leq 10000$) - координаты абонента. Все координаты во входном файле даны в одной и той же декартовой прямоугольной системе координат. Названия операторов - это непустые строки длиной не более 50 символов, состоящие из цифр, строчных и прописных букв английского алфавита. Прописные и строчные буквы английского алфавита различаются (например, *MPS* и *mps* - два разных оператора).

Выходные данные - в первой строке выведите число k - количество операторов сотовой связи, работающих в городе (различаются два оператора считаются разными, если их названия не совпадают). Далее выведите k строк. Каждая из этих строк должна содержать название оператора и количество базовых станций этого оператора, доступных абоненту. Первым должно идти название оператора, число базовых станций должно быть отделено от него одним пробелом. В этом списке операторы должны быть перечислены в том же порядке, в каком они встречаются во входном файле. Гарантируется, что $k \leq 100$.

Тесты: https://acmp.ru/index.asp?main=task&id_task=596

32. Ежегодно в Санкт-Петербурге, Барнауле и некоторых городах ближнего зарубежья проходят соревнования по программированию. Эти соревнования проходят в рамках студенческого чемпионата мира по программированию, организованного одной из самых авторитетных ассоциаций *ACM* (*Association for Computing Machinery*). На этих соревнованиях проходит отбор команд с Северо-Восточного Европейского Региона *NEERC* (*North-Eastern European Regional Contest*). Ежегодно перед организаторами соревнований встает проблема определения команд, которые будут приглашены к участию в финале чемпионата мира по программированию. По новым правилам в финал проходят не более N команд, представляющих *NEERC*. Кроме этого, от одного университета не могут проходить более K команд. При этом из всех таких множеств выбирается то, в котором сумма мест, занятых этими командами в полуфинальных соревнованиях минимально возможная. Ваша задача по итоговому протоколу

полуфинальных соревнований и числам N и K определить, какие команды будут приглашены к участию в финале чемпионата мира.

Входные данные - в первой строке находится три натуральных числа: P – количество команд, принявших участие в полуфинале, N и K ($N, K \leq P \leq 10^5$). В следующих P строках, по одному в строке перечислены названия университетов, команды которых заняли соответствующие места. Название университета содержит строчные и прописные английские буквы и пробелы. Длина названия университета не превышает 30 символов. В следующей строке перечислены номера команд соответствующих университетов. Таким образом, если название университета записано в i -ой строке ($2 \leq i \leq P+1$), то эта команда заняла $i-1$ место на полуфинале и имеет номер, записанный на $i-1$ месте в $P+2$ строке.

Выходные данные - выведите названия команд, приглашенных к участию в финале чемпионата мира по программированию, упорядоченных по месту, занятому на полуфинале. В качестве названия команды выведите название университета и через пробел «#ID», где ID – номер команды.

Тесты: https://acmp.ru/index.asp?main=test&id_task=984

33. Время от времени каждому школьнику, изучающему английский язык, приходится сдавать учителю сочинение на английском языке. Учителя английского языка бывают разные. Когда школьник использует в сочинении слова, которые на уроках еще не проходили, одни восхищаются юным талантом, другие багровеют от злости и ставят двойку непослушному ученику, осмелившемуся кичиться своими знаниями. К сожалению, Ваша учительница из других. Она не потерпит ни малейшего отступления от использования словарного запаса. В этот раз еще одна беда обрушилась на Вашу голову. Сочинение, заданное на завтра контрольное сочинение по выученным словам. А это значит, что все слова, которые Вы выучили на уроках, должны присутствовать в сочинении хотя бы по одному разу. Таким образом, перед сдачей сочинения Вам необходимо проверить, что каждое слово в тексте сочинения встречается в словаре, и каждое слово из словаря встречается в тексте.

Входные данные - в первой строке находится два числа N и M ($1 \leq N \leq 10^3$, $1 \leq M \leq 10^5$). В следующих N строках находится по одному слову из словаря. Все слова состоят из строчных английских букв. Длина каждого слова не превышает 20. Каждое слово состоит хотя бы из одного символа. Лишних пробелов перед словом и после него нет. В следующих M строках находится текст сочинения. Текст состоит из заглавных и строчных английских букв, пробелов и знаков препинания: точек (.), запятых (,), двоеточий (:), точек с запятыми (;), тире (-), апострофов ('), кавычек ("), восклицательных (!) и вопросительных

(?) знаков. Общая длина текста не превосходит 10^4 символов. В данной задаче большие и маленькие буквы в словах не различаются.

Выходные данные - выведите «*Everything is going to be OK.*», если с сочинением все в порядке. Если не все слова из текста встречаются в словаре, выведите «*Some words from the text are unknown.*». Если же предыдущее неверно, но некоторые слова из словаря не встречаются в тексте, выведите «*The usage of the vocabulary is not perfect.*».

Тесты: https://acmp.ru/index.asp?main=task&id_task=997

34. Мир наш развивается, строятся города, люди улетают в космос, изменяется система аттестации студентов в КГУ. Но вот проблема - система аттестации студентов изменили, а программное обеспечение, которое поставлено в деканатах для контроля успеваемости, оставили прежним. Поэтому Вам срочно требуется внедрить во всех деканатах новую программу поиска студентов с заданным баллом!

Входные данные - в первой строке содержатся натуральные числа N и K ($N, K \leq 200\,000$) – соответственно количество студентов, подлежащих аттестации, и число запросов декана об успеваемости студентов. Во второй строке находятся N целых чисел a_i , упорядоченных по возрастанию. Эти числа - аттестационные баллы студентов. В третьей строке располагаются K целых чисел b_i , определяющие искомый балл. ($0 \leq a_i, b_i \leq 2^{32}$)

Выходные данные - выведите для каждого из K запросов через пробел слово «*YES*», если студент с таким баллом есть, и «*NO*» в противном случае.

Тесты: https://acmp.ru/index.asp?main=task&id_task=945

35. Разработка новой поисковой системы, которая ведется группой компаний *Giggle*, проходит под лозунгом «Ничего и не терялось». Вы работаете в отделе передовых разработок, и на сегодняшний день вашей задачей является разработка тестовой версии поискового «движка». Тестовая версия реализует лишь часть функциональности полной версии. В частности, отсутствуют такие функции, как использование логических выражений в запросах, перевод найденных страниц с одного языка на другой и т.д. Возможности тестовой версии поисковой системы ограничиваются обработкой трех видов запросов: запросов на добавление, на удаление и на поиск. Система работает следующим образом. В любой момент времени существует множество известных системе сайтов, причем для каждого сайта известно множество ключевых слов, встречающихся на нем. Запрос на добавление содержит ключевое слово и название сайта. При его выполнении ключевое слово добавляется в множество ключевых слов, присутствующих на данном сайте. Если этого слова в соответствующем множестве еще нет, то результатом запроса является «*OK*», в противном случае – «*Already exists*». Запрос на удаление содержит ключевое слово и название сайта. При его выполнении ключевое слово исключается из

множества ключевых слов, присутствующих на данном сайте. Если этого слова в соответствующем множестве нет, то результатом запроса является «*Not found*», в противном случае – «*OK*». Запрос на поиск содержит только ключевое слово. Результатом запроса является лексикографически отсортированный список сайтов, содержащих данное ключевое слово. При этом в результат выводятся только первые 10 сайтов из этого списка. Задан список запросов. Необходимо вывести результат их последовательного выполнения.

Входные данные - первая строка содержит целое число n – количество запросов, которые необходимо обработать ($0 \leq n \leq 2500$). Каждая из последующих n строк содержит запрос. Запрос на добавление имеет следующий формат:

Add keyword < keyword > to < site >, где < keyword > – ключевое слово, – название сайта, на который добавляется это ключевое слово.

Запрос на удаление имеет следующий формат:

Remove keyword < keyword > from < site >, где < keyword > – ключевое слово, < site > – название сайта, на котором удаляется это ключевое слово.

Запрос на поиск имеет следующий формат.

Search < keyword >, где < keyword > – ключевое слово.

Все ключевые слова (< keyword >) состоят из строчных букв английского алфавита. Длины ключевых слов не превосходят 60 символов.

Все названия сайтов (< site >) состоят из строчных букв английского алфавита, символов «косая черта» («/») и точки («.»). Длины названий сайтов не превосходят 100 символов.

Выходные данные - вывести результат для каждого запроса. При этом придерживайтесь формата, приведенного в примерах. Не забудьте обратить внимание на второй пример. Результаты запросов разделяйте строкой из пяти символов «равно» («=*=*»).

Тесты: https://amp.ru/index.asp?main=task&id_task=964

Задача 2

1. Известна следующая информация о 17 книгах: название, автор, число страниц, цена. Определить число книг с ценой выше заданной.
2. Известны данные о 18 сотрудниках фирмы: фамилия, возраст, должность и отношение в воинской службе (военнообязанный или нет). Напечатать фамилии всех военнообязанных сотрудников.
3. Даны названия 26 городов и стран, в которых они находятся. Среди них есть города, находящиеся в Италии. Напечатать их названия.

4. Известна следующая информация о 15 торговых центрах в городе: название торговой сети, номер магазина, адрес, торговая площадь. Определить сумму торговых площадей заданной торговой сети.

5. Известны данные о мощности двигателя (в л.с.) и стоимости 30 легковых автомобилей. Определить общую стоимость автомобилей, у которых мощности двигателя превышают 100 л.с.

6. Известна следующая информация о 17 студентах: ФИО, номер группы, оценки за экзамены по дисциплинам – Информатика, Алгебра и теория чисел, История, Введение в математический анализ. Определить и вывести на экран средний балл каждого студента.

7. Известны рост и пол каждого из 22 человек. Найти средний рост мужчин.

8. Известна следующая информация о 18 различных товарах: наименование товара, вес, цена. Найти общую стоимость товаров, вес которых не превышает заданный.

9. Известна следующая информация о 12 группах студентов: номер группы, название специальности, число студентов в группе, число подгрупп. Определить номер группы с наибольшим средним числом студентов в подгруппе.

10. Имеется информация о количестве осадков, выпавших за каждый день месяца, и о температуре воздуха в эти дни. Определить, суммарное количество осадков, выпавшее в виде снега, в виде дождя (считать, что идет дождь, если температура воздуха выше 0°).

11. Имеется информация о результатах сдачи экзаменов студентами Вашей группы в зимнюю сессию в виде: фамилия, имя, название дисциплины, оценка. Программа должна вычислять абсолютную успеваемость (число положительных оценок в процентном отношении к общему числу оценок), качество знаний (число отличных и хороших оценок в процентном отношении к общему числу оценок), а также выводить список отличников группы.

12. Известна следующая информация о расписании 12 поездов: номер поезда, назначение (например, «Курск-Москва»), дата и время прибытия, дата и время отправления. По заданной дате и времени определить, какие поезда в данный момент находятся на станции.

13. Известна следующая информация о 12 компьютерных аудиториях: число рабочих мест, число компьютеров, наличие сети, имя ответственного лаборанта. Определить имя лаборанта, ответственного за аудиторию с максимальным числом компьютеров, подключенных к сети.

14. Известна следующая информация о каждом из 15 фильмов: название, страна производства, год и рейтинг зрителей. Определить страну производства фильмов с наивысшим рейтингом зрителей по заданному году.

15. Известна следующая информация о 12 видах кофейных напитков: наименование, сорт, ингредиенты, цена. Найти среднюю цену для каждого сорта кофе.

16. Известна следующая информация о багаже 20 пассажиров: количество вещей и общий вес багажа. Найти число пассажиров, количество вещей которых не больше среднего числа вещей всех пассажиров.

17. Известна следующая информация об n книгах библиотеки (n – натуральное число): название, автор, количество экземпляров, расположение (читальный зал или абонемент). Книгу из абонемента с наименьшим числом экземпляров переместить в читальный зал.

18. Известна следующая информация об n автобусах автобусного парка (n – натуральное число): номер автобуса, фамилия и инициалы водителя, номер маршрута, признак того, где находится автобус – на маршруте или в парке. По заданному номеру маршрута отправить в рейс все автобусы, находящиеся в парке.

19. Известна следующая информация о 18 видах товаров в кондитерской лавке: наименование, цена, вес, калорийность. Среди n (n – натуральное число ≤ 18) самых дешевых товаров найти наиболее калорийный.

20. Известна следующая информация об успеваемости n студентов (n – натуральное число): фамилия студента, название дисциплины, оценка. Название дисциплины может повторяться в разных записях. Найти названия дисциплин, по которым все студенты имеют только положительные отметки, для каждой из этих дисциплин вывести фамилии студентов.

21. Известна следующая информация об n видах канцелярских принадлежностей в магазине (n – натуральное число): вид, цена, количество экземпляров, имеющихся в наличии. После изменения экономической ситуации в стране, магазин поднял цены на все товары на 5%, кроме товаров, количество которых в остатке было меньше 5 штук. Отобразить новую таблицу характеристик канцелярских принадлежностей в магазине.

22. Известна следующая информация об n клиентах спортивного клуба (n – натуральное число): фамилия, номер дисконтной карты, срок абонемента (1, 3, 6 или 12 месяцев), общий срок пользования услугами клуба (количество месяцев). Добавить столбец, содержащий информацию о персональных скидках на услуги каждого из клиентов. Процент скидки определяется в зависимости от общего срока пользования

услугами клуба (например, общий срок пользования услугами клуба менее 12 месяцев – нет скидки, от 12 до 18 месяцев – 2% скидки, от 18 до 24 месяцев – 4% и т.д.)

23. Известна следующая информация об n товарах продуктового магазина (n – натуральное число): штрих-код товара, наименование, цена. Чтобы увеличить продажи, магазин решил провести акции – 10% скидки на товары, стоимостью более 500 рублей. Добавить столбец, содержащий новую цену для каждого товара. Вычислить сумму выручки магазина от полной продажи товаров при условии проведения акции.

24. Известна следующая информация об n работниках фирмы (n – натуральное число): фамилия, должность, сколько дней работает, количество опозданий. Руководство фирмы решило поощрить премией работников, выполняющих режим работы (менее 5% опозданий) и наказать штрафом работников, нарушающих режим (более 20% опозданий). Добавить столбец, содержащий запись «премия», «штраф» или «без изменений» для каждого из работников.

25. Известна следующая информация об n клиентах таксопарка (n – натуральное число): номер карты клиента, количество поездок, общая сумма поездок. Для клиентов таксопарка существует следующая система скидок: при общей сумме поездок от 1000 до 5000 рублей – 2% скидки; от 5000 до 10000 рублей – 5% скидки, сумма свыше 10000 рублей – 10% скидки. Добавить столбец, содержащий процент скидки для каждого из клиентов таксопарка. По заданному номеру карты и текущей стоимости поездки определить стоимость поездки со скидкой.

26. Известна следующая информация об n заявках в сервисном центре по ремонту компьютерной техники (n – натуральное число): наименование товара, дата поступления, описание неисправности, статус заявки (напр, «рассматривается» или «обработано»). Удалить все записи со статусом заявки «обработано».

27. Известна следующая информация из n записей о задолженностях студентов (n – натуральное число): фамилия, номер группы, название предмета. Удалить всех студентов, имеющих более 3-х хвостов. Учитывать, что в разных группах могут встречаться однофамильцы.

28. Известна следующая информация об n заявках на авиабилеты (n – натуральное число): пункт назначения, номер рейса, фамилию и инициалы пассажира, желаемую дату вылета. Удалить все заявки, зарегистрированные на указанную дату вылета для заданного номера рейса.

29. Известны данные о росте n юношей класса, упорядоченные по убыванию (n – натуральное число). Нет ни одной пары учеников, имеющих одинаковый рост. В начале учебного года в класс поступил новый ученик (известно, что его рост не совпадает с

ростом ни одного из учеников класса, превышает рост самого низкого ученика и меньше роста самого высокого). Получить новый список фамилий учеников (с учетом фамилии "новенького"), в котором фамилии расположены в порядке убывания роста.

30. Известна следующая информация об n квартирах, зарегистрированных в риэлтерской фирме (n – натуральное число): количество комнат, этаж, площадь, адрес. По данным, полученным от пользователя, организовать поиск в картотеке подходящего варианта: при равенстве количества комнат и этажа, и различии площади в пределах 10% выводится соответствующая информация и запись о квартире удаляется из картотеки.

31. Определите средний балл всех учащихся по каждому предмету.

Входные данные

Заданы сначала количество учащихся n , затем n строк, каждая из которых содержит фамилию, имя и три числа (оценки по трем предметам: математике, физике, информатике). Данные в строке разделены одним пробелом. Оценки принимают значение от 1 до 5.

Выходные данные

Выведите три действительных числа: средний балл всех учащихся по математике, по физике, по информатике.

Тесты: <https://informatics.msk.ru/mod/statements/view.php?chapterid=328>

32. Выведите фамилии и имена учащихся, не имеющих троек (а также двоек и четверок).

Входные данные

Заданы сначала количество учащихся n , затем n строк, каждая из которых содержит фамилию, имя и три числа (оценки по трем предметам: математике, физике, информатике). Данные в строке разделены одним пробелом. Оценки принимают значение от 1 до 5.

Выходные данные

Необходимо вывести пары фамилия-имя по одной на строке, разделяя фамилию и имя одним пробелом. Выводить оценки не нужно. Порядок вывода должен быть таким же, как в исходных данных.

Тесты: <https://informatics.msk.ru/mod/statements/view.php?chapterid=329>

33. Определите трех учащихся с наилучшим средним баллом по трем предметам. Выведите фамилии и имена этих учащихся. Если при этом у нескольких учащихся средний балл совпадает со средним баллом учащегося, "занявшего 3-е место", то необходимо вывести их всех.

Входные данные

Заданы сначала количество учащихся n , затем n строк, каждая из которых содержит фамилию, имя и три числа (оценки по трем предметам: математике, физике, информатике). Данные в строке разделены одним пробелом. Оценки принимают значение от 1 до 5.

Выходные данные

Необходимо вывести пары фамилия-имя по одной на строке, разделяя фамилию и имя одним пробелом. Выводить оценки не нужно. Порядок вывода должен быть таким же, как в исходных данных.

Тесты <https://informatics.msk.ru/mod/statements/view.php?chapterid=331>

34. Выведите фамилии и имена учащихся в порядке убывания их среднего балла.

Входные данные

Заданы сначала количество учащихся n , затем n строк, каждая из которых содержит фамилию, имя и три числа (оценки по трем предметам: математике, физике, информатике). Данные в строке разделены одним пробелом. Оценки принимают значение от 1 до 5. Общее число учащихся не превосходит 100001.

Выходные данные

Необходимо вывести пары фамилия-имя по одной на строке, разделяя фамилию и имя одним пробелом. Выводить оценки не нужно. Если несколько учащихся имеют одинаковые средние баллы, то их нужно выводить в порядке, заданном во входных данных.

Тесты <https://informatics.msk.ru/mod/statements/view.php?chapterid=332>

35. Однажды, неслучая секретарша перепутала личные дела учащихся. Теперь их снова необходимо упорядочить сначала по классам, а внутри класса по фамилиям

Входные данные

В первой строке дано число n ($1 \leq n \leq 1000$) – количество личных дел. Далее для каждого из n учащихся следующие данные (каждое в своей строке): фамилия и имя, класс, дата рождения. Фамилия и имя – строки не более чем из 20 символов, класс – строка состоящая из числа (от 1 до 11) и латинской буквы (от "A" до "Z"), дата рождения – дата в формате "ДД.ММ.ГГ". Гарантируется, что внутри одного класса нет однофамильцев.

Выходные данные

В выходной файл требуется вывести n строк, в каждой из которых записаны данные по одному учащемуся. Строки должны быть упорядочены сначала по классам, а затем по фамилиям.

Тест <https://informatics.msk.ru/mod/statements/view.php?id=279&chapterid=1404>

Задача 3

1. Создайте перечисляемый тип, описывающий цвета светофора.
2. Создайте перечисляемый тип, описывающий времена года.
3. Создайте перечисляемый тип, описывающий оценки по пятибалльной шкале.
4. Создайте перечисляемый тип, описывающий цвета радуги.
5. Создайте перечисляемый тип, описывающий музыкальные ноты.
6. Создайте перечисляемый тип, описывающий дни недели.
7. Создайте перечисляемый тип, описывающий месяцы года.
8. Создайте перечисляемый тип, описывающий жанры кино.
9. Создайте перечисляемый тип, описывающий масти игровых карт.
10. Создайте перечисляемый тип, описывающий ранги игровых карт.
11. Создайте перечисляемый тип, описывающий литературные жанры.
12. Создайте перечисляемый тип, описывающий студентов вашей подгруппы.
13. Создайте перечисляемый тип, описывающий ваших преподавателей.
14. Создайте перечисляемый тип, описывающий дисциплины вашей последней сессии.
15. Создайте перечисляемый тип, описывающий виды занятий в вузе.
16. Создайте перечисляемый тип, описывающий виды отчетных документов студента.
17. Создайте перечисляемый тип, описывающий операционные системы для ПК.
18. Создайте перечисляемый тип, описывающий типы продуктовых товаров.
19. Создайте перечисляемый тип, описывающий должности преподавателей вузов.
20. Создайте перечисляемый тип, описывающий научные степени.
21. Создайте перечисляемый тип, описывающий военные звания.
22. Создайте перечисляемый тип, описывающий знаки зодиака по месяцам.
23. Создайте перечисляемый тип, описывающий знаки зодиака по годам.
24. Создайте перечисляемый тип, описывающий печатных изданий.
25. Создайте перечисляемый тип, описывающий континенты.
26. Создайте перечисляемый тип, описывающий виды канцелярских принадлежностей.
27. Создайте перечисляемый тип, описывающий виды коммунальных платежей.

28. Создайте перечисляемый тип, описывающий производителей смартфонов.
29. Создайте перечисляемый тип, описывающий производителей ноутбуков.
30. Создайте перечисляемый тип, описывающий языки программирования.
31. Создайте перечисляемый тип, описывающий бутики торгового центра.
32. Создайте перечисляемый тип, описывающий породы собак.
33. Создайте перечисляемый тип, описывающий типы вердиктов тестирующей системы.
34. Создайте перечисляемый тип, описывающий классы программного обеспечения.
35. Создайте перечисляемый тип, описывающий виды задач по олимпиадному программированию.