

Задача 1 (Python). Вычислительная геометрия.

Требования:

- В следующих задачах предполагается, что в файле записано несколько пар чисел, которые можно рассматривать как координаты множества точек на плоскости или как координаты множества концов отрезков на прямой.
- Реализованная программная реализация должна:
 - обрабатывать переполнение, возникшее в процессе вычисления;
 - реагировать и корректно обрабатывать введенные числа вне вещественного диапазона;
 - проверять корректность введенных данных;
 - проверять наличие файла;
 - корректно обрабатывать название файла;
- Требования к интерфейсной части не предъявляется и должно быть разработано студентом.

1. Множество точек определяет ломаную. Имеет ли она самопересечения?
2. Множество точек определяет многоугольник. Является ли он выпуклым?
3. Множество точек определяет многоугольник. Для данной точки определить где она расположена относительно этого многоугольника: внутри, снаружи, на границе.
4. Дано множество отрезков на прямой. Принадлежит ли отрезок $[a, b]$ их объединению?
5. Два множества точек задают два многоугольника. Определить расстояние между этими многоугольниками
6. Дано множество точек. Найти центр и радиус минимального круга, который содержит все эти точки.
7. Дано множество точек. Найти из данного множества две различные точки так, чтобы окружности содержали одинаковое число точек. Определить минимальные и максимальные радиусы данных окружностей.
8. Дано множество отрезков на прямой. Выбрать из него и вывести те отрезки, объединение которых дает отрезок наибольшей длины.
9. Даны центры равномерно растущих кругов на плоскости. При столкновении друг с другом столкнувшиеся круги прекращают свой рост. Найти радиусы кругов, когда процесс роста остановится полностью.
10. Дано множество точек на плоскости. Построить выпуклую оболочку этого множества.
11. Множество точек определяет многоугольник. Построить многоугольник, который получится, если линию, задающую каждую сторону, отодвинуть в перпендикулярном ей направлении на величину h .
12. Даны центр круга, его радиус и координаты отрезка на плоскости. Определить количество точек пересечения отрезка и круга (нуль, одну или две точки пересечения).
13. Даны центры кругов и их радиусы. Определить расположение двух окружностей (пересекаются, касаются, не пересекаются).
14. Множество точек определяет многоугольник. Подсчитать количество точек с целочисленными координатами, лежащих внутри него (но не на его границе).

15. Множество точек определяет два многоугольника. Определить минимальное расстояние между ними.
16. Дано множество точек. Найти треугольник с минимальной площадью.
17. Дано множество точек. Определить, сколько из них лежит внутри заданного прямоугольника.
18. Дано множество точек Q . Для каждой точки множества $q \in Q$ определить ближайшую к ней точку из $Q \setminus \{q\}$. Предложенный алгоритм задачи должен иметь вычислительную сложность $O(n \log n)$.
19. Множество точек определяет многоугольник. Разделить заданный многоугольник на два многоугольника таким образом, чтобы получившиеся площади соотносились как n к m . Вывести координаты новых многоугольников.

Задание 2 (C++). Классы.*Требования:*

- В отдельном файле должен быть написан тест на заданный класс (под тестом понимается описанные в файле команды, которые автоматически построчно считываются программной реализацией).
- Дополнительно к заданию в классе должны быть также реализованы следующие методы:
 - создать новый экземпляр класса;
 - отобразить (распечатать) и посчитать число созданных экземпляров класса;
 - искать экземпляр класса по заданным параметрам;
 - удалить экземпляр класса по заданным параметрам;
- Дополнительно к заданию в программной реализации должно быть также продемонстрировано:
 - наследование (производный класс может быть либо взят из соседней задачи, либо придуман самостоятельно; в производном классе также должны быть определены необходимые конструктор, деструктор, операторы и методы);
 - инкапсуляция (public/private члены класса);
 - полиморфизмом (способность функции обрабатывать данные разных типов).
- Программная реализация **не подразумевает** обработку вещественных чисел вне диапазона, однако должна корректно работать с выделенной памятью и адекватно обрабатывать вводимые данные.
- Требования к интерфейсной части не предъявляется и должно быть разработано студентом.
- Размерность запрашивается с клавиатуры.

1. Определить класс `CComplexVector` для работы с векторами комплексных чисел. Длина вектора задается в конструкторе класса. В классе должны быть определены необходимые конструкторы, деструктор, операторы присваивания, сложения, вычитания, скалярного умножения.

2. Определить класс `CVector` для работы с вектором вещественных чисел. Длина вектора задается в конструкторе класса. В классе должны быть определены необходимые конструкторы, деструктор, операторы присваивания, сложения, вычитания, скалярного умножения.

3. Определить класс `CIntN` для работы с целыми знаковыми числами, состоящими из N десятичных цифр, где N задается в конструкторе. В классе должны быть определены необходимые конструкторы, деструктор, операторы присваивания, сложения, вычитания.

4. Определить класс `CRVector` для работы с вектором вещественных чисел произвольной длины. Внутри класса вектор должен быть реализован с помощью указателя на `double` и целой переменной, в которой хранится количество отведенной под вектор памяти. Длина вектора должна изменяться, если происходит присваивание значения элементу вектора. В классе должны быть определены необходимые конструкторы, деструктор, операторы присваивания, сложения, вычитания, скалярного умножения.

5. Определить класс `CRat` для работы с вектором дробей вида p_i/q_i , где p_i – целое, q_i – натуральное. Длина вектора задается в конструкторе класса. В классе должны быть определены необходимые конструкторы, деструктор, операторы присваивания, сложения, вычитания, скалярного умножения.

6. Определить класс CString для работы со строками, длина которых хранится в самом классе (т.е. строки произвольных символов). В классе должны быть определены необходимые конструкторы, операторы присваивания, сложения (слияния строк), присваивания обычной строки переменной типа CString.
7. Определить класс CCoordPl для работы с множеством точек \mathbb{R}^2 . Такое множество хранит координаты (x, y) точек плоскости и позволяет выбирать точки, лежащие в некоторой окрестности заданной точки. Предполагается, что все точки находятся внутри изначально заданного прямоугольника. В классе должны быть определены конструктор, деструктор, операторы присваивания, сложения, вычитания, скалярного умножения.
8. Определить параметризованный класс CMap, который реализует отображение, где ключом является строка, а значением – целое число. Число отображений задается в конструкторе класса. В классе должны быть определены конструктор, деструктор, операторы присваивания, сложения (для ключей – побуквенное сложение строк, для значений – сложение целых чисел), вычитания.
9. Определить класс CTime для работы со временем часовых поясов (UTC). Предполагается, что значащими параметрами времени являются число, месяц, год, час, минута, секунда и часовой пояс. Число используемых отсечек времени задается в конструкторе класса. В классе должны быть определены конструктор, деструктор, операторы присваивания, сложения, вычитания и сравнения.
10. Определить класс CComplexE для работы с комплексными числами $e^{i*\varphi}$. Число используемых комплексных чисел задается в классе. В классе должны быть определены конструктор, деструктор, операторы присваивания, сложения, вычитания, умножения и деления, реализован метод перевода числа $e^{i*\varphi}$ в форму $a+i*b$ по формуле Эйлера и обратно.
11. Определить класс CPVector для работы с вектором с началом в точке $(0,0)$ и координатами конца (x,y) . В классе должны быть определены конструктор, деструктор, операторы присваивания, сложения, вычитания, умножения (на заданное число), реализованы методы определения угла между вектором и осью x (в градусах и радианах) и квадранта (четверти) плоскости. Дружественный класс определяет, находятся ли два вектора в одном квадранте (четверти) плоскости и угол между ними.
12. Определить класс CFrac для работы с простыми дробями a/b , заданной числителем и знаменателем. Метод печати должен быть в виде a/b . В классе должны быть определены конструктор, деструктор, операторы присваивания, сложения, вычитания, деления, умножения и сравнения, метод определения бесконечной периодической дроби (периода), правильности дроби, целой части дроби.
13. Определить класс CMatrix для работы с матрицами. Размер матрицы задается в конструкторе класса. В классе должны быть определены конструктор, деструктор, операторы присваивания, сложения, вычитания, умножения (на заданное число), реализованы методы нахождения определителя матрицы, ранга матрицы, вычисления обратной матрицы.
14. Определить класс CTrackBody для работы с телом, которое начинает свое движение с постоянной скоростью v в заданной точке (x_0, y_0) по окружности произвольного радиуса. Предполагается, что в некоторый произвольный момент времени τ тело начинает двигаться по окружности другого произвольного радиуса (произведя переключение, таких переключений при движении за время T может происходить произвольное число раз). В классе должны быть определены конструктор, деструктор, операторы присваивания, сложения, вычитания, умножения (на заданное число), реализованы методы определения расстояния через заданное время, число произведенных переключений (в том числе

новые радиусы, по которым двигалось тело), квадранта (четверти) плоскости. Дружественный класс определяет расстояние между двумя телами в некоторый момент времени t .

15. Определить класс `SBalTrackBody` для работы с телом, брошенном в пространстве из начальной точки (x_0, y_0) со скоростью v_0 под углом α к горизонту с массой m_0 . В классе должны быть определены конструктор, деструктор, операторы присваивания, сложения (характеристик тел), вычитания, реализованы методы расчёта дальности полета и максимальной высоты подъема. Дружественный класс определяет какое тело улетело дальше.

16. Определить класс `STriangle` для работы с треугольником, определенным координатами вершин. В классе должны быть определены конструктор, деструктор, операторы присваивания, сложения (например, площадей), вычитания, реализованы методы расчёта длин сторон, периметра, углов, типа треугольника и подсчета площади. Дружественный класс определяет подобие треугольников.

17. Определить класс `SQuadratic` для работы с квадратными уравнениями, заданное коэффициентами a , b и c . В классе должны быть определены конструктор, деструктор, операторы присваивания, сложения (коэффициентов), вычитания, реализованы методы расчета значения дискриминанта, решения x_1 и x_2 . Дружественный класс определяет, имеет ли решение уравнение при перестановке коэффициентов между собой.

18. Определить класс `SLagranz` для работы с полиномом Лагранжа, проходящего через 3 точки с координатами (x_1, y_1) , (x_2, y_2) и (x_3, y_3) . В классе должны быть определены конструктор, деструктор, операторы присваивания, сложения (например, по координатно 3х точке и построения нового полинома Лагранжа), вычитания, деления, умножения, реализованы методы распечатки коэффициентов полинома Лагранжа, проходящего через 3 точки, значения полинома в точке x . Дружественный класс проверяет, проходит ли полином через точку с координатами (a, b) .

19. Определить класс `SPolinomN` для работы с полиномом n -ой степени, определенного коэффициентами при степенях x . Степень полинома задается в конструкторе класса. В классе должны быть определены конструктор, деструктор, операторы присваивания, сложения (коэффициентов), вычитания, реализованы методы вычисления значения полинома в точке $x=a$, производной полинома в точке $x=a$. Дружественный класс находит разность значений между двумя полиномами.

Задача 3 (C/C++). Динамические структуры.

Требования:

- Программная реализация **не подразумевает** обработку вещественных чисел вне диапазона и направлена исключительно на работу с памятью;
- Программная реализация может быть исполнена как на C, так и на C++ (предполагается выбор одного из языков программирования);
- Программная реализация всех заданий (задача состоит из реализации 3 подзадач) может быть реализована в рамках единого программного куска кода (если в задании используется язык C) или быть реализована в рамках одного класса (если в задании используется язык C++), однако функциональные части программной реализации должны быть логически понятны;
- Требований к интерфейсной части не предъявляется и должно быть разработано студентом.

Блок А. Односвязные списки.

A1. Найти количество максимальных элементов списка действительных чисел.

A2. Написать функцию, которая по списку L строит два новых списка: L_1 – из положительных элементов и L_2 – из отрицательных элементов списка L.

A3. Определить, является ли список упорядоченным по возрастанию.

A4. Сформировать список целых чисел, вводимых пользователем, в том порядке, в котором вводятся эти числа, но без повторений элементов.

A5. Написать функцию, которая оставляет в списке L только первые вхождения одинаковых элементов.

A6. Определить количество различных элементов списка действительных чисел, если известно, что его элементы образуют возрастающую последовательность.

A7. Имеется список целых чисел. Продублировать в нем все четные числа.

A8. Написать функцию, которая по двум данным линейным спискам формирует новый список, состоящий из элементов, одновременно входящих в оба данных списка.

A9. Написать функцию, которая по двум линейным спискам L_1 и L_2 формирует новый список L, состоящий из элементов, входящих в L_1 , но не входящих в L_2 .

A10. Написать функцию, которая в линейном списке из каждой группы подряд идущих одинаковых элементов оставляет только один.

A11. Написать функцию, которая удаляет из списка элементы, входящие в него только один раз.

A12. Пусть имеется список L_1 действительных чисел. Записать в список L_2 все элементы списка L_1 в порядке возрастания их значений.

A13. Пусть имеется список действительных чисел $a_1 \rightarrow a_2 \rightarrow \dots \rightarrow a_n$. Сформировать новый список $b_1 \rightarrow b_2 \rightarrow \dots \rightarrow b_n$ такой же размерности по следующему правилу: элемент b_k равен сумме элементов исходного списка с номерами от 1 до k.

A14. Написать функцию, которая, получив в качестве параметра указатель q на один из элементов списка и некоторое число x, добавляет новый элемент со значением x после (до) элемента, на который указывает ссылка q.

A15. Написать функцию, которая, получив в качестве параметра указатель q на один из элементов списка, удаляет элемент, расположенный после элемента (сам элемент), на который указывает ссылка q .

A16. Сформировать список действительных чисел. Затем преобразовать его, прибавив к положительным числам максимальный элемент.

A17. Удалить из списка все элементы, встречающиеся более одного раза.

A18. Дан список целых чисел. Продублировать в нем все простые числа.

A19. Определить, есть ли в списке действительных чисел элементы, превосходящие сумму всех элементов списка.

A20. Определить, образуют ли элементы списка действительных чисел геометрическую прогрессию.

A21. Удалить из списка действительных чисел все максимальные элементы.

A22. Пусть имеются два списка, элементы которых упорядочены по возрастанию. Сформировать новый список из элементов первого и второго списка, элементы которого будут упорядочены.

A23. Сформировать список действительных чисел $a_1 \rightarrow a_2 \rightarrow \dots \rightarrow a_n$, вводимых пользователем. Затем сформировать новый список $b_1 \rightarrow b_2 \rightarrow \dots \rightarrow b_n$ такой же размерности по следующему правилу: элемент b_k является максимальным из элементов исходного списка с номерами от 1 до k .

A24. Пусть имеется текстовый файл. Используя линейный список, подсчитать число появлений каждого слова и создать новый текстовый файл, каждая строка которого имеет вид <слово>– <число его появлений>. Слова должны располагаться в лексикографическом порядке.

A25. Написать функцию добавления элемента в циклический список и функцию печати всех элементов списка, начиная с некоторого элемента.

A26. Записать в циклический список n целых чисел, вводимых пользователем. Также написать функцию, которая получает в качестве параметров указатель на один из элементов списка q и целое число $k > 0$. Данная функция пробегает по элементам списка, начиная с элемента q , и останавливается на k -м по счету элементе, который удаляется, и счет начинается со следующего элемента, и так до тех пор, пока в списке не останется один элемент. Функция возвращает значение данного элемента.

Блок Б. Стеки и очереди.

B1. Сформировать стек целых чисел, вводимых пользователем, а затем его содержимое вывести на экран, не удаляя элементы, из которых извлекается информация. Также написать функцию удаления стека.

B2. Расположить элементы целочисленного массива размером n в обратном порядке с использованием стека.

B3. Написать функцию, которая слова в текстовом файле распечатывает в обратном порядке. По файлу можно пройти только один раз.

B4. Элементы целочисленного массива записать в очередь. Написать функцию извлечения элементов из очереди до тех пор, пока первый элемент очереди не станет четным.

B5. Даны две непустые очереди, которые содержат одинаковое количество элементов. Объединить очереди в одну, в которой элементы исходных очередей чередуются.

B6. Даны две непустые очереди. Элементы каждой из очередей упорядочены по возрастанию. Объединить очереди в одну с сохранением упорядоченности элементов.

Б7. Пусть имеется файл действительных чисел и некоторое число C . Используя очередь, напечатать сначала все элементы, меньшие числа C , а затем все остальные элементы.

Блок В. Двусвязные списки.

В1. Пусть имеется односвязный список действительных чисел, каждый элемент которого содержит дополнительное (нереализованное) ссылочное поле `prev`. Преобразовать исходный односвязный список в двусвязный, в котором каждый элемент связан не только с последующим элементом (с помощью поля `next`), но и с предыдущим (с помощью поля `prev`).

В2. Написать функцию, которая по произвольному указателю на один из элементов двусвязного списка подсчитывает количество элементов в этом списке.

В3. Написать функцию, которая, получив в качестве параметра указатель на один из элементов двусвязного списка действительных чисел и два числа, добавляет первое число в начало списка, а второе в его конец.

В4. Дано число x и указатель q на один из элементов непустого двусвязного списка. Вставить после данного элемента списка новый элемент со значением x .

В5. Написать рекурсивную функцию, которая осуществляет вывод на экран всех элементов двусвязного списка в обратном направлении, начиная с последнего элемента.

В6. Имеется двусвязный список действительных чисел. Продублировать в нем все положительные числа.

В7. Дано некоторое число x . Удалить из двусвязного списка все элементы со значением x .

В8. Пусть имеется некоторый двусвязный список действительных чисел и некоторое число C . Требуется переставить значения элементов таким образом, чтобы сначала следовали (в произвольном порядке) элементы, меньшие числа C , а затем элементы, не меньшие числа C .

В9. Определить, расположены ли элементы в двусвязном списке симметричным образом.

В10. Осуществить циклический сдвиг элементов двусвязного списка на k позиций вправо.

Блок Г. Бинарные деревья.

Г1. Найти сумму элементов бинарного дерева.

Г2. Найти вершины, у которых количество потомков в левом поддереве не равно количеству потомков в правом поддереве.

Г3. Найти вершины, для которых высота левого поддерева не равна высоте правого поддерева.

Г4. Написать функцию, которая определяет число вхождений элемента x в бинарное дерево.

Г5. Найти максимальный элемент бинарного дерева и количество повторений максимального элемента в данном дереве.

Г6. Написать функцию, которая определяет, есть ли в бинарном дереве хотя бы два одинаковых элемента.

Г7. Написать функцию, определяющую максимальное количество одинаковых элементов бинарного дерева.

Г8. Написать функцию, которая определяет, является ли бинарное дерево симметричным.

Г9. Написать функцию, которая определяет, является ли бинарное дерево деревом поиска.

Г10. Вывести все листья дерева поиска в порядке возрастания.

Г11. Пусть имеется бинарное дерево T . Сформировать два идеально сбалансированных дерева из отрицательных и неотрицательных элементов дерева T .

Г12. Вывести на экран все пути, ведущие от корня к листьям бинарного дерева, у которых суммарный вес элементов минимальный.

Г13. Найти последний номер из всех уровней бинарного дерева, на которых есть положительные элементы.

Г14. На каждом уровне бинарного дерева найти максимальный элемент.

Г15. На каждом уровне дерева найти количество внутренних вершин и количество листьев.

Г16. Найти суммы элементов всех нечетных уровней.

Г17. Найти минимальный и максимальный пути между листьями бинарного дерева.

Г18. Удалить из бинарного дерева наименьшее количество вершин таким образом, чтобы полученное дерево было строго бинарным.

Г19. Пусть имеется текстовый файл. Используя дерево поиска, создать другой текстовый файл – частотный словарь, содержащий слова и количество появлений каждого слова в исходном файле.

Г20. Написать функцию, которая осуществляет послойный обход бинарного дерева, при котором значения вершин печатаются от уровня к уровню, начиная с корневой вершины. Значения вершин дерева на каждом уровне печатаются слева направо. Для реализации алгоритма использовать структуру очередь следующим образом. На первом шаге вставить в очередь корневую вершину. Затем прописать цикл, работающий по такому принципу.

Пока очередь не пуста:

- берем из очереди первый элемент q ;
- выводим значение элемента q на экран;
- если у вершины q есть левая вершина-потомок в дереве, то добавляем этот потомок в очередь;
- если у вершины q есть правая вершина-потомок в дереве, то добавляем этот потомок в очередь;
- удаляем из очереди элемент q .