**Лабораторная работа №10**

Вариант A

1. Ввести строки из файла, записать в список. Вывести строки в файл в обрат­ном порядке.
2. Ввести число, занести его цифры в стек. Вывести число, у которого цифры идут в обратном порядке.
3. Создать список из элементов каталога и его подкаталогов.
4. Занести стихотворения одного автора в список. Провести сортировку по возрастанию длин строк.
5. Задать два стека, поменять информацию местами.
6. Определить множество на основе множества целых чисел. Создать методы для определения пересечения и объединения множеств.
7. Списки, стеки или очереди T(1..n) и U(1..n) содержат результаты n-измере­ний тока и напряжения на неизвестном сопротивлении R. Найти прибли­женное число R методом наименьших квадратов.
8. С использованием множества выполнить попарное суммирование произ­вольного конечного ряда чисел по следующим правилам: на первом этапе суммируются попарно рядом стоящие числа, на втором этапе суммируются результаты первого этапа и т.д. до тех пор, пока не останется одно число.
9. Сложить два многочлена заданной степени, если коэффициенты многочле­нов хранятся в объекте **HashMap**.
10. Умножить два многочлена заданной степени, если коэффициенты многоч­ленов хранятся в различных списках.
11. Не используя вспомогательных объектов, переставить отрицательные эле­менты данного списка в конец, а положительные — в начало списка.
12. Ввести строки из файла, записать в список **ArrayList**. Выполнить сорти­ровку строк, используя метод **sort()** из класса **Collections**.
13. Задана строка, состоящая из символов «(», «)», «[», «]», «{», «}». Проверить правильность расстановки скобок. Использовать стек.
14. Задан файл с текстом на английском языке. Выделить все различные слова. Слова, отличающиеся только регистром букв, считать одинаковыми. Использовать класс **HashSet**.
15. Задан файл с текстом на английском языке. Выделить все различные слова. Для каждого слова подсчитать частоту его встречаемости. Слова, отличаю­щиеся регистром букв, считать различными. Использовать класс **HashMap**.
16. Заполнить **HashMap** 10 объектами <**Integer**, **String**>. Найти строки у кото­рых *ключ*>5. Если *ключ* = 0, вывести строки через запятую. Перемножить все ключи, где длина строки>5.
17. Написать функцию, которая получала бы итераторы на начало и конец от­сортированного **List** и заданный символ. Возвращать функция должна на­чало и конец диапазона, строки в котором начинаются с заданного символа.

Вариант B

1. В кругу стоят N человек, пронумерованных от 1 до N. При ведении счета по кругу вычеркивается каждый второй человек, пока не останется один. Составить две программы, моделирующие процесс. Одна из программ должна использовать класс **ArrayList**, а вторая — **LinkedList**. Какая из двух программ работает быстрее? Почему?
2. Задан список целых чисел и некоторое число X. Не используя вспомога­тельных объектов и методов сортировки и не изменяя размера списка, пе­реставить элементы списка так, чтобы сначала шли числа, не превосходя­щие X, а затем числа, больше X.
3. Написать программу, осуществляющую сжатие английского текста. Построить для каждого слова в тексте оптимальный префиксный код по алгоритму Хаффмена. Использовать класс **PriorityQueue**.
4. Реализовать класс **Graph**, представляющий собой неориентированный граф. В конструкторе класса передается количество вершин в графе. Методы должны поддерживать быстрое добавление и удаление ребер.
5. На базе коллекций реализовать структуру хранения чисел с поддержкой следующих операций:

* добавление/удаление числа;
* поиск числа, наиболее близкого к заданному (т.е. модуль разницы мини­мален).

1. Реализовать класс, моделирующий работу N-местной автостоянки. Машина подъезжает к определенному месту и едет вправо, пока не встретится сво­бодное место. Класс должен поддерживать методы, обслуживающие приезд и отъезд машины.
2. Во входном файле хранятся две разреженные матрицы — А и В. Построить циклически связанные списки СА и СВ, содержащие ненулевые элементы соответственно матриц А и В. Просматривая списки, вычислить: а) сумму S = A + B; б) произведение P = A х B.
3. Во входном файле хранятся наименования некоторых объектов. Построить список Z, элементы которого содержат наименования и шифры данных объектов, причем элементы списка должны быть упорядочены по возраста­нию шифров. Затем «сжать» список Z, удаляя дублирующие наименования объектов.
4. Во входном файле расположены два набора положительных чисел; между наборами стоит отрицательное число. Построить два списка C1 и С2, эле­менты которых содержат соответственно числа 1-го и 2-го набора таким образом, чтобы внутри одного списка числа были упорядочены по возра­станию. Затем объединить списки C1 и С2 в один упорядоченный список, изменяя только значения полей ссылочного типа.
5. Во входном файле хранится информация о системе главных автодорог, свя­зывающих г. Полоцк с другими городами Беларуси. Используя эту инфор­мацию, построить дерево, отображающее систему дорог республики, а за­тем, продвигаясь по дереву, определить минимальный по длине путь из г. Полоцка в другой заданный город. Предусмотреть возможность сохране­ния дерева в виртуальной памяти.
6. Один из способов шифрования данных, называемый «двойным шифрова­нием», заключается в том, что исходные данные при помощи некоторого преобразования последовательно шифруются на некоторые два ключа — K1 и K2. Разработать и реализовать эффективный алгоритм, позволяющий находить ключи K1 и K2 по исходной строке и ее зашифрованному вариан­ту. Проверить, оказался ли разработанный способ действительно эффек­тивным, протестировав программу для случая, когда оба ключа являются 20-битными (время ее работы не должно превосходить одной минуты).
7. На плоскости задано *N* точек. Вывести в файл описания всех прямых, кото­рые проходят более чем через одну точку из заданных. Для каждой прямой указать, через сколько точек она проходит. Использовать класс **HashMap**.
8. На клетчатой бумаге нарисован круг. Вывести в файл описания всех клеток, целиком лежащих внутри круга, в порядке возрастания расстояния от клет­ки до центра круга. Использовать класс **PriorityQueue**.
9. На плоскости задано *N* отрезков. Найти точку пересечения двух отрезков, имеющую минимальную абсциссу. Использовать класс **TreeMap**.
10. На клетчатом листе бумаги закрашена часть клеток. Выделить все различ­ные фигуры, которые образовались при этом. Фигурой считается набор за­крашенных клеток, достижимых друг из друга при движении в четырех направлениях. Две фигуры являются различными, если их нельзя совме­стить поворотом на угол, кратный 90 градусам, и параллельным перено­сом. Используйте класс **HashSet**.
11. Дана матрица из целых чисел. Найти в ней прямоугольную подматрицу, состоящую из максимального количества одинаковых элементов. Исполь­зовать класс **ArrayDeque**.
12. Реализовать структуру «черный ящик», хранящую множество чисел и имею­щую внутренний счетчик K, изначально равный нулю. Структура должна поддерживать операции добавления числа в множество и возвращение *K*-го по минимальности числа из множества.
13. На прямой гоночной трассе стоит *N* автомобилей, для каждого из которых известны начальное положение и скорость. Определить, сколько произой­дет обгонов.
14. На прямой гоночной трассе стоит *N* автомобилей, для каждого из которых известны начальное положение и скорость. Вывести первые *K* обгонов.