



ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ»

**ПРАКТИКУМ
ПО КУРСУ «СТРУКТУРЫ И АЛГОРИТМЫ
КОМПЬЮТЕРНОЙ ОБРАБОТКИ ДАННЫХ»**

Составители:
И.Е. Воронина
Н.В. Огаркова

Издательско-полиграфический центр
Воронежского государственного университета
2009

Содержание

Задания на тему «Модульное программирование»	4
Задания на тему «Стеки. Очереди»	9
Задания на тему «Объектно-ориентированное программирование»	12
Задания на тему «Двоичные деревья»	16
Задания на тему «Хеширование»	19
Задания на тему «Сильно ветвящиеся деревья»	21
Задания на тему «Алгоритмы с возвратом»	23
Задания на тему «Внутренние сортировки»	26
Задания на тему «Внешние сортировки»	30
Список литературы	35

Утверждено научно-методическим советом факультета ПММ 26 декабря 2008 г., протокол № 4

Рецензент канд. физ.-мат. наук, доц. кафедры МО ЭВМ факультета ПММ
О.Д. Горбенко

Пособие подготовлено сотрудниками кафедры программного обеспечения и администрирования информационных систем факультета ПММ Воронежского государственного университета.

Рекомендовано для студентов 2 курса дневного отделения, обучающихся по специальности 010503 – Математическое обеспечение и администрирование информационных систем.

Для специальности 010503 – Математическое обеспечение и администрирование информационных систем

Задания на тему «Модульное программирование»

1. Составить модуль, реализующий следующие операции над векторами:

- ввод и вывод компонент вектора;
- определение порядкового номера минимальной (максимальной) компоненты вектора;
- определение минимальной (максимальной) компоненты вектора;
- вычисление скалярного произведения двух векторов.

А. Даны два вектора X и Y размерности $n = 6$. Используя составленный модуль, вычислить

$$\frac{n \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{(k \sum x_i^2 - (\sum x_i)^2)(p \sum y_i^2 - (\sum y_i)^2)}}$$

где k – порядковый номер максимальной компоненты вектора X ;
 p – порядковый номер минимальной компоненты вектора Y .

В. Даны два вектора X и Y размерности $n = 6$. Используя составленный модуль, вычислить

$$\frac{n \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{(n \sum x_i^2 - (\sum x_i)^2)(n \sum y_i^2 - (\sum y_i)^2)}}$$

С. Даны три вектора X, Y, Z , каждый размерности $n = 6$. Используя составленный модуль, вычислить

$$k * (A, A) - p * (B, C),$$

где A – тот из данных трех векторов, минимальный элемент которого имеет самый большой номер (считать, что такой вектор единственный); B и C – два других вектора; k – номер минимального элемента вектора Z ; p – номер максимального элемента вектора Y .

2. Составить модуль, реализующий следующие операции над векторами:

- ввод и вывод компонент вектора;
- определение минимальной (максимальной) компоненты вектора;
- вычисление скалярного произведения двух векторов;
- проверку на упорядоченность по возрастанию или убыванию последовательности компонент.

Даны три вектора X, Y, Z , каждый размерности $n = 6$. Используя составленный модуль, вычислить

$$\min X * (A, A) + \max Y * (B, C),$$

где A – тот из данных трех векторов, компоненты которого упорядочены по возрастанию (считать, что такой вектор единственный); B и C – два других вектора; $\min X$ – минимальная компонента вектора X ;
 $\max Y$ – максимальная компонента вектора Y .

3. Разработать модуль для реализации операций с комплексными числами.

Вычислить значение квадратного трехчлена $ax^2 + bx + c$ с комплексными коэффициентами в комплексной точке x .

4. Разработать модуль для реализации операций с комплексными числами.

Решить квадратное уравнение $ax^2 + bx + c = 0$ с комплексными коэффициентами.

5. Разработать модуль для реализации операций с комплексными числами.

Вычислить значение комплексной функции в заданной комплексной точке z при заданном значении n

$$e^z = 1 + \frac{z}{1!} + \frac{z^2}{2!} + \dots + \frac{z^n}{n!}.$$

6. Разработать модуль для реализации операций с комплексными числами.

Вычислить с заданной точностью ε значение комплексной функции в заданной комплексной точке z

$$\cos(z) = 1 - \frac{z^2}{2!} + \frac{z^4}{4!} - \frac{z^6}{6!} + \dots$$

7. Реализовать модуль для работы с множеством целых чисел, число элементов в котором больше, чем 256. Модуль должен содержать процедуры инициализации множества, включения, исключения элемента, проверки принадлежности элемента множеству; объединения, пересечения и разности множеств. Написать программу, иллюстрирующую работу данного модуля. Множество должно удовлетворять следующему описанию:

a)

```
Const
  MaxDigit = 10000;
  ArraySize = MaxDigit div (High(byte)+1);
Type
  TSet = set of byte;
  TBigSet = array [0..ArraySize] of TSet;
```

b)

```
Const
  MaxDigit = 10000;
  ArraySize = MaxDigit div (High(byte)+1);
Type
  TSet = set of byte;
  PSet = ^TSet;
  TBigSet = array [0..ArraySize] of PSet;
```

c)

```
Type
  TSet = set of byte;
```

```

TList = ^TNode;
TNode = record
  Index : integer;
  S      : TSet;
  Next  : TList;
end;

TBigSet = TList;

```

8. Создать модуль для работы с длинными строками. Длинная строка – это линейный список из символов. Для типа данных *длинная строка* должны быть реализованы следующие операции:

- функция, возвращающая длину строки (аналог функции Length для типа **string**);
- поиск позиции первого вхождения подстроки в строку (аналог функции Pos для типа **string**);
- копирование подстроки из строки, начиная с заданной позиции, заданной длины (аналог функции Copy для типа **string**);
- удаление из строки заданного количества символов, начиная с заданной позиции (аналог процедуры Delete для типа **string**);
- вставка подстроки в строку, начиная с заданной позиции (аналог процедуры Insert для типа **string**);
- слияние двух строк в одну (аналог функции Concat);
- преобразование строки символов в число и, наоборот, числа – в строку (аналоги функций IntToStr и StrToInt (или TryStrToInt));
- удаление начальных, конечных, а также и начальных, и конечных пробелов в строке (аналоги функции TrimLeft, TrimRight, Trim).

Используя составленный модуль, решить задачи.

- A. Дана строка текста, состоящая из слов, разделенных одним или несколькими пробелами. Заменить все слова, совпадающие с первым, на слово, заданное пользователем.
- B. Дана строка текста, состоящая из слов, разделенных одним или несколькими пробелами. Удалить слова, длина которых совпадает с длиной первого слова.
- C. Дана строка текста, состоящая из слов, разделенных одним или несколькими пробелами. Если слово состоит только из цифр и его длина не более четырех символов, преобразовать его в целое число и напечатать. Найти сумму таких чисел.
- D. Заданы две строки *str1* и *str2*. Выясните, можно ли из *str1* путем перестановки литер получить строку *str2*.

E. Заданы две строки *str1* и *str2*. Подсчитать, сколько раз подстрока *str1* встречается в строке *str2*.

9. Создать модуль, содержащий описание типа «Дата» и операции над ним:

- ввод даты (день, месяц, год);
- вывод даты;
- проверка корректности совокупности полей (день, месяц, год);
- увеличение даты на заданное число дней;
- уменьшение даты на заданное число дней;
- преобразование типа «строка» → «дата»;
- преобразование типа «дата» → «строка».

Написать программу, иллюстрирующую работу полученного модуля.

10. Создать модуль, реализующий представление длинного целого числа в виде строки символов, а также операции сложения, вычитания, умножения, деления нацело и нахождения остатка от деления нацело. Написать программу, иллюстрирующую работу полученного модуля.

11. Создать модуль, реализующий представление полиномов, а также следующие операции над ними: сложение, вычитание, умножение на число, умножение двух полиномов. Написать программу, иллюстрирующую работу полученного модуля.

12. Реализовать модуль для работы с линейным списком из целых чисел:

- a) без заглавного звена;
- b) с заглавным звеном.

В модуле должны быть реализованы следующие операции: инициализация, проверка на пустоту, добавление звена в начало списка, добавление звена в конец списка, добавление звена за заданным звеном, удаление первого звена, удаление последнего звена, удаление звена, следующего за заданным, поиск заданного элемента в списке, построение упорядоченного списка, печать элементов списка. Написать программу, иллюстрирующую работу полученного модуля.

13. Разработать модуль для работы с двунаправленным списком, в котором должны быть реализованы следующие операции: инициализация списка, проверка на пустоту, печать элементов списка, удаление (очистка) списка; добавление звена (в начало списка, в конец списка, за заданным звеном, перед заданным звеном); удаление звена (первого, последнего, текущего); перемещение по списку (в начало, в конец, к предыдущему звену, к последующему звену). Используя составленный модуль, вычислить:

$$a) S = (x_1 + x_2 + 2x_n)(x_2 + x_3 + 2x_{n-1}) \dots (x_{n-1} + x_{n-2} + 2x_2);$$

$$b) S = (x_1^2 - x_n^2)(x_2^2 - x_{n-1}^2) \dots (x_n^2 - x_1^2);$$

$$c) S = x_1x_n + x_2x_{n-1} + \dots + x_nx_1.$$

14. Разработать модуль для работы с кольцевым списком. Используя составленный модуль, решить задачу «Считалка»: N ребят располагаются по кругу. Начав отсчет от первого, удаляют каждого k -ого, смыкая круг после каждого удаления. Определите порядок удаления ребят из круга.
15. Текстовый файл содержит словарь, состоящий из слов, разделенных на составляющие (морфемы): приставки, корни, суффиксы и окончания. Известно, что в слове не более трех приставок, всегда присутствуют корень и окончание (если окончание пустое, то оно обозначается символом «#»). Кроме окончания, в слове может присутствовать еще один компонент – энклитика «СЯ». После каждого слова задана частота его употребления в тексте. Информация по каждому слову расположена на отдельной строке. Составляющие слова разделяются символом «/». Даже если какая-либо составляющая отсутствует, ее позиция отделена символом «/». Если приставок или суффиксов в слове несколько, то между собой они разделены символом «|». Например, «/ЗА/БАВ/Л|Я|ТЬ/#/СЯ». Реализовать модуль, позволяющий по заданному текстовому файлу или файлам формировать файлы, состоящие из последовательностей морфем, упорядоченных по алфавиту, формировать частотные словари как слов, так и морфем, добавлять в файлы морфем новые компоненты (без нарушения лексикографического порядка), удалять или заменять морфемы.

Задания на тему «Стеки. Очереди»

Указание: в заданиях с нечетными номерами стек или очередь должны быть реализованы на основе цепочного представления, а в заданиях с четными номерами – на основе сплошного представления.

1. Дан текстовый файл. Занести в другой файл символы исходного файла в обратном порядке.
2. Дан текстовый файл (строка). Обратить каждое слово текста.
3. Дана последовательность чисел, ограниченная нулем. Распечатать числа последовательности в обратном порядке.
4. Для последовательности, ограниченной нулем, распечатать в обратном порядке отдельно четные и нечетные элементы.
5. Для последовательности, ограниченной нулем, распечатать в обратном порядке сначала положительные, а затем отрицательные элементы.
6. Напечатать строки файла в обратном порядке.
7. Напечатать обращенные строки файла в обратном порядке.
8. На двух стержнях перемешаны кольца двух цветов. Используя третий стержень, переместить на разные стержни кольца одинаковые по цвету.
9. Даны три стержня. На двух из них – кольца, упорядоченные по убыванию их размера. Переместить кольца на третий стержень так, чтобы кольца располагались по возрастанию их размера.
10. Даны два стека, содержащие целые числа (считать, что все числа различны). Напечатать максимальное и минимальное числа, восстановив при этом содержимое стеков. Не использовать дополнительные структуры, кроме набора переменных.
11. Даны два стека, содержащие целые числа (считать, что все числа различны). Поменять местами максимальное и минимальное числа, не используя дополнительных структур данных, кроме набора переменных.
12. В строке записан текст, сбалансированный по круглым скобкам. Требуется для каждой пары соответствующих открывающей и закрывающей скобок напечатать номера их позиций в тексте, упорядочив пары номеров в порядке возрастания номеров позиций закрывающих скобок.
13. В строке записан текст, сбалансированный по круглым скобкам. Требуется для каждой пары соответствующих открывающей и закрывающей скобок напечатать номера их позиций в тексте, упорядочив пары номеров в порядке возрастания номеров позиций открывающих скобок.

14. Даны два стека. Подсчитать общее количество элементов в них, не используя дополнительных структур данных, кроме набора переменных, и восстановив при этом содержимое стеков.
15. В текстовом файле записана без ошибок формула следующего вида:
 $\langle \text{формула} \rangle ::= \langle \text{цифра} \rangle \mid M(\langle \text{формула} \rangle, \langle \text{формула} \rangle) \mid m(\langle \text{формула} \rangle, \langle \text{формула} \rangle)$
 $\langle \text{цифра} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$
 M обозначает функцию \max , а m — \min . Вычислить значение данной формулы (например, $M(5, m(6, 8)) = 6$).
16. Преобразовать арифметическое выражение в постфиксную форму. В выражение входят:
 - переменные;
 - целые числа;
 - знаки '+', '-', '*', '/';
 - скобки '(', ')
17. Вычислить значение арифметического выражения, хранящегося в постфиксной форме. В выражение входят:
 - целые числа;
 - знаки '+', '-', '*', '/', '~' (так обозначается унарный минус).
18. Преобразовать выражение из постфиксной формы в инфиксную.
19. Распечатать возрастающие серии последовательности целых чисел в обратном порядке (серия — упорядоченная подпоследовательность максимальной длины).
 А. Серии в прямом порядке, а числа внутри них — в обратном.
 В. Серии в обратном порядке, а числа внутри них — в прямом.
20. Проверить правильность расстановки скобок 3-х типов в выражении. Виды скобок: '{', '}', '(', ')', '[', ']'.
21. Дана строка текста. Напечатать гласные и согласные буквы отдельно с сохранением порядка.
22. Задана последовательность, ограниченная нулем. Напечатать положительные и отрицательные числа отдельно с сохранением порядка.
23. Даны две очереди, содержащие целые числа. Напечатать сначала все четные, а потом все нечетные, не используя дополнительных структур, кроме набора переменных базовых типов.
24. Задан текстовый файл. Распечатать все слова максимальной длины (с использованием очереди).
25. Дана очередь из слов (все слова различны). Удалить из нее все слова, содержащие больше гласных, чем согласных. Не использовать дополнительные структуры данных, кроме набора переменных.

26. Даны две очереди из чисел. Разместить в одной очереди положительные элементы, а в другой — отрицательные, не используя дополнительных структур данных, кроме набора переменных.
27. Даны две очереди из слов. Разместить в одной очереди слова с четной длиной, а в другой — с нечетной, не используя дополнительных структур данных, кроме набора переменных.
28. Даны две очереди. Подсчитать общее количество элементов в них, не используя дополнительных структур данных, кроме набора переменных, и восстановив при этом содержимое очередей.
29. Даны две очереди X и Y , содержащие вещественные числа. Из каждой очереди одновременно извлекается по одному числу x и y соответственно. Если $x < y$, то число $(x + y)$ помещается в конец очереди X , иначе число $(x - y)$ помещается в конец очереди Y . Необходимо определить число шагов, через которое одна из очередей станет пустой.
30. Дан текстовый файл A . Переписать его содержимое в файл B , перенося при этом в конец каждой строки все входящие в нее знаки препинания.
31. Задан массив $Name [1..N]$ (элементы имеют тип $array [1..15] \text{ of } \text{char}$), содержащий имена людей, и массив $Children [1..N, 1..N]$ (элементы имеют тип $boolean$), в котором $Children [i, j] = true$, если человек под номером j является ребенком человека под номером i . Для человека с заданным номером k напечатать: сначала — имена всех его детей, затем — всех его внуков, затем — всех правнуков и т. д.

Задания на тему «Объектно-ориентированное программирование»

1. На верхнем уровне иерархии классов находится линейный список, тип элементов которого `Pointer`. Для него определены следующие методы: добавить в начало, добавить в конец, добавить за заданным звеном; удалить звено из начала списка, удалить из конца, удалить за заданным звеном; найти звено с заданным элементом, удалить звено с заданным элементом, вставить звено с заданным элементом в упорядоченный список и другие. Реализовать класс *Линейный список* таким образом, чтобы можно было породить любого из потомков: список из чисел, список из строк, список из записей и т.д.
 - А. Реализовать потомка – список из строк и проиллюстрировать работу программы.
 - В. Реализовать потомка – список из целых чисел и проиллюстрировать работу программы.
 - С. Реализовать потомка – список из записей (любых) и проиллюстрировать работу программы.
2. На вершине иерархии расположен класс – стек, элементами которого являются данные типа `Pointer`. Наследники данного класса – стек из целых чисел и стек из строк. Реализовать указанную иерархию объектов и решить следующую задачу: даны два стека с некоторыми элементами – числами или строками (по выбору пользователя). Поместить элементы, удовлетворяющие заданному критерию в один стек, а не удовлетворяющие – в другой. Критерий отбора для каждого из потомков определить самостоятельно.
3. На вершине иерархии классов – класс *Очередь*, элементами которого являются данные типа `Pointer`. Потомки данного класса – очередь из целых чисел и очередь из строк. Реализовать указанную иерархию объектов и решить следующую задачу: даны две очереди с некоторыми элементами – числами или строками (по выбору пользователя). Напечатать сначала элементы, удовлетворяющие критерию, а потом – не удовлетворяющие. Критерий отбора для каждого из потомков определить самостоятельно.
4. Реализовать класс «Список», построенный на основе динамического массива, элементами которого являются данные типа `Pointer` (аналог класса `TList` в Delphi). На вершине иерархии классов – класс «Стек указателей» с элементами типа `Pointer`, полем которого является «Список». Наследником класса «Стек указателей» является класс «Стек чисел». Решить следующую задачу: задана строка текста, распечатать в обратном порядке отдельно положительные и отрицательные числа.

5. Реализовать класс «Список», построенный на основе динамического массива, элементами которого являются данные типа `Pointer` (аналог класса `TList` в Delphi). На вершине иерархии классов – класс «Очередь указателей» с элементами типа `Pointer`, полем которого является «Список». Наследником класса «Очередь указателей» является класс «Очередь строк». Решить следующую задачу: задана строка текста, распечатать сначала слова четной длины, а потом – нечетной.
6. Реализовать класс «Линейный список из целых чисел». Класс должен содержать следующие методы:
 - добавить элемент в начало списка;
 - добавить элемент в конец списка;
 - вставить элемент после заданного звена;
 - удалить звено из начала списка;
 - удалить звено из конца списка;
 - удалить элемент после заданного звена;
 - найти звено с заданным элементом;
 - удалить звено с заданным элементом;
 - вставить звено с заданным элементом в упорядоченный список и другие.

На вершине иерархии – класс «Структура данных», полем которого является «Линейный список из целых чисел», реализующий методы:

- добавить элемент;
- удалить элемент;
- очистить структуру данных;
- заполнение структуры данных из строки текста;
- преобразование данных в строку текста.

Часть из представленных методов являются абстрактными. Потомками класса «Структура данных» должны быть стек из чисел, очередь из чисел, упорядоченный список из чисел. Пользователь выбирает одного из потомков (того, с которым он хочет работать) и вводит последовательность чисел, которые затем выводятся в соответствии с порядком, определенным структурой данных.

7. На вершине иерархии расположен абстрактный стек из строк и в нем же определен метод, который позволяет по заданной строке текста распечатать все наиболее длинные слова в обратном порядке. Реализовать потомков – стек на основе линейного списка, а также стек на основе массива. Пользователь определяет, с каким из потомков он хочет работать при решении задачи.
8. На вершине иерархии расположена абстрактная очередь из целых чисел. В этом же классе определен метод, который позволяет по заданной строке текста распечатать числа в прямом порядке. Реализовать потом-

ков – очередь на основе линейного списка, а также очередь на основе массива. Пользователь определяет, с каким из потомков он хочет работать при решении задачи.

9. Пусть элемент списка представлен следующим образом:

```
TElem = class
private
    Left, Right : TElem;
protected
    // Здесь, возможно, определены методы.
public
    // Здесь, возможно, определены методы.
end;
```

Построить двунаправленный список таким образом, чтобы, определив потомков класса TElem и не переопределяя сам список, можно было решить следующую задачу: по заданной строке текста сформировать список из элементов, соответствующих типу потомка, и вывести его на экран. Решение задачи должно быть представлено как метод класса *Список*:

- A. Определить потомка TIntElem (звено, содержащее целое число).
 - B. Определить потомка TStrElem (звено, содержащее строку). Выбираются только те слова, которые не могут быть трактованы как целые числа или как логические значения.
 - C. Определить потомка TBoolElem (звено, содержащее логическое значение).
10. Задан некоторый родительский класс. От него должно быть определено несколько потомков. Определить класс *Список элементов*, который может одновременно содержать разных потомков базового класса. Реализовать операции добавления элемента в список, удаление элемента из списка, поиск элемента в списке и другие. Решить задачи.
- A. «Публикации». Реализовать следующую иерархию классов: любая публикация имеет название, автора (авторов) и год издания. Для книги всегда задано количество страниц. Для статьи указываются номера страниц (диапазон) и название сборника, в котором опубликована данная статья. Задан список публикаций. Сделать выборку публикаций заданного автора, вышедших в течение определенного периода времени.
 - B. «Детали компьютера». Любая деталь компьютера, хранящаяся на складе, имеет идентификационный номер – целое число. Детали бывают разными, например, память, процессор, монитор и т.д. Как следствие, детали обладают разными характеристиками. На складе имеются различные детали, для каждой из которых известно количество в штуках и цена. Предложить покупателю разные варианты

нужной ему детали с заданными характеристиками на определенную сумму.

- C. «Владельцы транспортных средств». О каждом владельце транспортного средства известны ФИО, год рождения, телефон и т. п. Транспортные средства бывают разными: для перевозки грузов должны быть известны грузоподъемность, открытый верх или закрытый и т. п.; для транспорта, предназначенного для перевозки людей, – количество посадочных мест, модель, цвет; мотоциклы могут применяться и для перевозки людей, и для перевозки пакетов или документов. В некотором агентстве имеется список владельцев транспортных средств. Поступает некоторый заказ. Необходимо подобрать те машины, которые могут обслужить данный заказ.
- D. «Фигуры». Любая геометрическая фигура характеризуется точкой привязки и цветом. В зависимости от того, какая это фигура, известны ее остальные характеристики, например, для окружности – это радиус; для прямоугольника, у которого стороны параллельны осям координат, – еще одна точка; для треугольника – две другие вершины. Известен список геометрических фигур. Определите суммарную площадь всех фигур; определите, лежит ли хотя бы один круг внутри какого-нибудь прямоугольника.
- E. «Почта». Любое почтовое отправление имеет отправителя, получателя и дату отправки. Рассмотрим почтовые отправления трех видов: письмо, посылка и перевод. У каждого письма есть текст и признак того, является ли оно заказным. Для посылки известен ее вес (вещественное число), а для почтового перевода – сумма перевода (целое число). Имеется список почтовых отправлений. Определите, какое количество переводов отправил гражданин N в течение заданного периода времени.

Задания на тему «Двоичные деревья»

1. Подсчитать число узлов в заданном двоичном дереве.
2. Подсчитать число узлов на k -ом уровне заданного двоичного дерева (корень считать узлом 1-го уровня).
3. Заданы два текстовых файла A и B . Максимальная длина слова — 20 символов. Занести в файл C те слова файла A , которых нет в файле B . Для хранения слов файла B и ускорения поиска среди них воспользуйтесь деревом двоичного поиска.
4. Описать логическую функцию, проверяющую на равенство два заданных двоичных дерева.
5. Описать процедуру или функцию, которая вычисляет среднее арифметическое всех элементов дерева.
6. Описать процедуру или функцию, которая находит в непустом дереве t длину (число ветвей) пути от корня до ближайшей вершины с заданным элементом E ; если E не входит в дерево t , за ответ принять (-1).
7. Описать процедуру или функцию, которая определяет число вхождений заданного элемента в дерево.
8. Описать процедуру или функцию, которая вычисляет сумму элементов дерева.
9. Описать логическую функцию, определяющую, есть ли в заданном двоичном дереве хотя бы два одинаковых элемента.
10. Описать процедуру, которая для заданного N строит двоичное дерево с количеством уровней N , где на каждом уровне i располагаются узлы, информационные части которых равны i .
11. Подсчитать число листьев (терминальных элементов) в заданном двоичном дереве.
12. Определить высоту заданного двоичного дерева.
13. В заданном двоичном дереве подсчитать число элементов, равных максимальному.
14. Заменить в заданном двоичном дереве все отрицательные элементы на их абсолютные величины.
15. Поменять местами максимальный и минимальный элементы заданного двоичного дерева, все элементы которого различны.
16. Для дерева A была сделана копия — дерево B , и из дерева B были удалены некоторые из ветвей. Необходимо сформировать список указателей на удаленные ветви дерева A .

17. Проверить, является ли заданное двоичное дерево сбалансированным.
18. Проверить, является ли заданное двоичное дерево идеально сбалансированным.
19. Описать процедуру, которая для заданного N строит дерево следующего вида: корнем является узел, информационная часть которого равна N ; второй уровень содержит узлы со значениями $N-1$ и $N-2$; третий уровень — $N-3$, $N-4$, $N-5$, $N-6$ и т. д. Последний уровень может быть неполным, он содержит узел с конечным значением — 1.
20. Формулу вида
 $\langle \text{формула} \rangle ::= \langle \text{терминал} \rangle | (\langle \text{формула} \rangle \langle \text{знак} \rangle \langle \text{формула} \rangle)$
 $\langle \text{знак} \rangle ::= + | - | * | /$
 $\langle \text{терминал} \rangle ::= \langle \text{переменная} \rangle | \langle \text{цифра} \rangle$
 $\langle \text{переменная} \rangle ::= a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z$
 $\langle \text{цифра} \rangle ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$
можно представить в виде двоичного дерева по следующим правилам: формула из одного терминального символа (цифры или переменной) представляется деревом, состоящим из одной вершины, содержащей этот символ, а формула вида $(f_1 s f_2)$ — деревом, в котором корень — это знак s , а левое и правое поддеревья — это соответствующие представления формул f_1 и f_2 . Описать подпрограмму, которая:
 - a) вычисляет значение дерева-формулы (считать, что значения переменных заданы массивом `Value: array ['a'..'z'] of real`);
 - b) по формуле, содержащейся в текстовом файле, строит соответствующее дерево-формулу;
 - c) печатает дерево-формулу в виде соответствующей формулы;
 - d) проверяет, является ли заданное двоичное дерево деревом-формулой;
 - e) упрощает дерево-формулу, заменяя в нем:
 - все поддеревья, соответствующие формулам $(f+0)$, $(0+f)$, $(f-0)$, $(f*1)$ и $(1*f)$, на поддеревья, соответствующие формуле f ;
 - поддеревья, соответствующие формулам $(f*0)$ и $(0*f)$, — на вершину с 0;
 - поддеревья $(a+b)$, $(a-b)$, $(a*b)$, (a/b) (где a и b — константы) — на вершину c , равную $a+b$, $a-b$, $a*b$, a/b соответственно;
 - f) преобразует дерево-формулу, заменяя в нем все поддеревья, соответствующие формулам $((f1 \pm f2) * f3)$ и $(f1 * (f2 \pm f3))$, на поддеревья, соответствующие формулам $((f1 * f3) \pm (f2 * f3))$ и $((f1 * f2) \pm (f1 * f3))$;
 - g) находит производную дерева-формулы по заданной переменной.

21. Для заданного двоичного дерева подсчитать длину внутреннего пути (сумма длин путей для всех его вершин).
22. Для заданного двоичного дерева подсчитать среднюю длину пути (сумма длин путей для всех его вершин, деленная на их количество).
23. Для заданного двоичного дерева подсчитать длину внешнего пути (сумма длин путей для всех его вершин, включая специальные. Специальная вершина – это вершина дерева, где отсутствует поддереву (nil)).
24. Проверить, является ли данное двоичное дерево деревом поиска.
25. Дан текстовый файл. Слова содержат не более 20 символов. Определить частоту использования каждого слова в тексте. Результат оформить в виде таблицы, содержащей слова в лексикографическом порядке.
26. В заданном дереве найти поддерево двоичного поиска с максимальным количеством элементов.
27. Подсчитать количество узлов в дереве, левое и правое поддеревья которых равны.
28. Подсчитать количество узлов в дереве, у которых каждый элемент левого поддерева можно найти среди элементов правого поддерева.
29. Подсчитать количество узлов в дереве, у которых хотя бы один элемент левого поддерева можно найти среди элементов правого поддерева.
30. Подсчитать количество узлов в дереве, у которых среди элементов правого поддерева нельзя найти элементы левого поддерева.
31. Проверить, что все элементы данного дерева встречаются в нем еще хотя бы один раз.
32. Описать процедуру или функцию, которая бы меняла местами самый правый узел дерева и его корень.
33. Подсчитать количество узлов в дереве, у которых все элементы левого поддерева различны.
34. Подсчитать количество листьев, у которых узлы, из которых они выходят, меньше по значению, чем сами листья.
35. Подсчитать количество деревьев, у которых все листья одинаковы.
36. Найти в дереве лист с кратчайшим расстоянием от вершины и напечатать этот путь.

Задания на тему «Хеширование»

Указание: в каждой из задач метод разрешения коллизий определяется одним из следующих пунктов:

- a) внутренние цепочки;
- b) внешние цепочки;
- c) метод отрытой адресации (линейное опробование);
- d) метод отрытой адресации (квадратичное опробование);
- e) метод отрытой адресации (двойное хеширование).

Требования:

1. Хеш-таблица должна быть реализована в виде класса.
2. В каждой из задач (4–9) должны быть реализованы следующие возможности:
 - загрузка данных из файла (текстового или типизированного);
 - сохранение данных в файл (текстовый или типизированный);
 - добавление данных;
 - удаление данных;
 - поиск данных;
 - выполнение основной задачи, указанной в упражнении.
1. Известен список служебных слов на языке Паскаль. По тексту некоторой программы определить частоту встречаемости этих слов.
2. Задан текстовый файл. Сформировать список слов, употребляющихся в тексте более пяти раз.
3. Задан текстовый файл. Сформировать набор словосочетаний из двух слов, встречающихся в тексте не менее двух раз.
4. Задан набор записей следующей структуры: табельный номер, ФИО, заработная плата. По табельному номеру найти остальные сведения.
5. Задан набор записей следующей структуры: номер телефона, ФИО, адрес. По номеру телефона найти сведения о ФИО владельца и его адресе.
6. Задан набор записей следующей структуры: название кинофильма, режиссер, список актеров, краткое содержание. По заданному названию фильма найти остальные сведения.
7. Задан набор записей следующей структуры: номер автомобиля (запись), его марка и ФИО владельца. По номеру автомобиля найти остальные сведения.

8. Задан набор записей следующей структуры: паспортные данные (запись), ФИО владельца, адрес прописки. По паспортным данным определить ФИО владельца и адрес.
9. Задан набор записей следующей структуры: ФИО студента, номер студенческого билета, результаты последней сессии. По номеру студенческого билета вывести ФИО студента и количество несданных зачетов и экзаменов.
10. Задан список имен людей. Определить частоту использования каждого имени в некотором тексте.
11. Создать модуль, реализующий методы для работы с хеш-таблицей: инициализация, добавление элемента, удаление элемента, поиск. Кроме того, при заполнении хеш-таблицы выше заданного уровня размер хеш-таблицы должен автоматически увеличиваться.
12. Исследовать зависимость числа коллизий от коэффициента заполненности хеш-таблицы. Результаты должны быть представлены в виде таблицы и в виде графика.
13. Разработать несколько хеш-функций для случая, когда ключевое поле имеет тип `string` и исследовать качество предложенных хеш-функций. Результаты должны быть представлены в виде графиков.
14. Из файла, содержащего текст на русском языке, слова помещаются в хеш-таблицу. Определить среднее число коллизий для нескольких заданных хеш-функций.

Задания на тему «Сильно ветвящиеся деревья»

1. Задано Trie-дерево. Подсчитать количество слов, начинающихся с определенной последовательности символов.
2. Дан текстовый файл, состоящий из слов, разделенных пробелами. Построить Trie-дерево, которое содержит перевернутые слова из файла. Найти все слова, имеющие заданное окончание.
3. В Trie-дереве посчитать количество слов, имеющих одинаковое количество гласных и согласных букв.
4. Из Trie-дерева удалить все слова, которые содержат заданную букву.
5. В Trie-дереве найти и напечатать все слова палиндромы.
6. В Trie-дереве определить количество слов, содержащих букву «А».
7. Из Trie-дерева удалить все слова с длиной большей чем заданная.
8. В Trie-дереве подсчитать количество слов, содержащих определенное количество согласных.
9. Дан текстовый файл, который содержит повторяющиеся слова. Распечатать их в алфавитном порядке, указав число вхождений каждого слова в текст.
10. В Trie-дереве подсчитать количество слов-палиндромов.
11. Из Trie-дерева удалить все слова, начинающиеся с заданной последовательности букв.
12. Задано Trie-дерево и некоторое слово. Вывести все формы данного слова, содержащиеся в дереве. Формой слова считается некоторое другое слово, отличающееся от заданного не более чем на n последних символов, где n вводит пользователь.
13. Из Trie-дерева удалить все слова, которые содержат указанную подстроку.
14. Дано Trie-дерево. Посчитать количество слов с нечетной длиной.
15. В Trie-дереве найти все слова, содержащие только символы заданного множества.
16. Для заданного Trie-дерева построить Trie-дерево, в котором все слова расположены в обратном порядке.
17. Из Trie-дерева удалить все слова с четной длиной.
18. Дано Trie-дерево. Напечатать все слова, длина которых меньше чем длина заданного слова.

19. Посчитать в Trie-дереве количество слов, оканчивающихся на согласную букву.
20. Написать процедуру, реализующую вставку элемента в B-дерево, при этом все дерево должно храниться во внешней памяти (например в файле из целых чисел).
21. Написать процедуру, реализующую удаление элемента из B-дерева, при этом все дерево должно храниться во внешней памяти (например в файле из целых чисел).
22. Написать процедуру, реализующую вставку элемента в (2-3)-дерево, при этом все дерево должно храниться во внешней памяти (например в файле из целых чисел).
23. Написать процедуру, реализующую удаление элемента из (2-3)-дерева, при этом все дерево должно храниться во внешней памяти (например в файле из целых чисел).
24. Написать процедуру, реализующую вставку элемента в B+ -дерево, при этом все дерево должно храниться во внешней памяти (например в файле из целых чисел).
25. Написать процедуру, реализующую удаление элемента из B+ -дерева, при этом все дерево должно храниться во внешней памяти (например в файле из целых чисел).

Задания на тему «Алгоритмы с возвратом»

1. Найти все вершины графа, недостижимые из заданной вершины. *Указание:* использовать рекурсивную процедуру проверки доступности одной вершины из другой.
2. Раскрасить граф минимальным количеством цветов. Каждая вершина должна быть помечена цветом, отличным от цвета смежных вершин.
3. Определить, является ли связным заданный граф. *Указание:* использовать рекурсивную процедуру проверки доступности одной вершины из другой.
4. Найти длину кратчайшего цикла в графе.
5. Найти вершину графа, которая принадлежит каждому пути между двумя заданными вершинами.
6. Задана система односторонних дорог. Найти путь, соединяющий города *A* и *B*, не проходящий через заданное множество вершин.
7. Задана система двусторонних дорог. Найти замкнутый путь, проходящий через каждую вершину и длиной не более 100 км.
8. Найти в системе двусторонних дорог город, для которого сумма расстояний до других городов минимальна.
9. По системе двусторонних дорог определить, есть ли в ней город, из которого можно добраться в любой другой, проехав меньше 100 км. Разрешается построить дополнительно три дороги.
10. По системе двусторонних дорог определить, можно ли, закрыв какие-либо три из них, добиться того, чтобы из города *A* нельзя было попасть в город *B*.
11. Задана система двусторонних дорог. *N*-периферией называется множество городов, расстояние от которых до выделенного города больше *N*. Определить *N*-периферию для заданного *N*.
12. Построить такой многоугольник (не обязательно выпуклый) с вершинами в заданном множестве, периметр которого максимален.
13. Найти минимальное множество прямых, на которых можно разместить все точки заданного множества.
14. В трехмерном пространстве задано множество точек. Найти разбиение этого множества на два непустых непересекающихся множества, чтобы их центры тяжести находились как можно ближе друг к другу.
15. В условии предыдущей задачи найти множество, содержащее ровно *N* точек, центр тяжести которого находится как можно ближе к началу координат.

16. Имеется n костей домино. Построить из них последовательность максимальной длины.
17. Дана последовательность костей домино. Определить последовательность ходов двух игроков, которые ведут к «рыбе» (ситуация, в которой у двух игроков есть кости, но ни один не может сделать ход).
18. Для шахматного поля размером $N \times N$, на котором расставлены черные и белые фигуры, найти наименьшее количество ферзей и их расстановку, при которой все поля доски, занятые фигурами противоположного цвета, находятся под ударом. Решить аналогичную задачу для следующих шахматных фигур:
- конь;
 - слон;
 - ладья.
19. Найти кратчайший по количеству ходов путь ладьи, позволяющий попасть из одной клетки в другую. Известно расположение белых и черных фигур на доске. Фигуры противоположного цвета можно «бить».
20. Задана квадратная матрица размером $N \times N$, которая содержит целые числа от 0 до 6. Костью домино можно накрыть две соседние ячейки матрицы (горизонтальные или вертикальные), если числа на кости совпадают с числами в ячейках. Проверить, можно ли заданную матрицу накрыть одним комплектом костей домино.
21. Какое минимальное количество комплектов домино необходимо, чтобы построить заданную матрицу?
22. Есть набор костей домино, часть из которых выложена на столе, а остальные находятся у двух игроков. Необходимо определить, существует ли последовательность ходов, при которой второй игрок вынужден пропустить ход из-за отсутствия необходимых костей. Игроки ходят по очереди. Игру начинает первый игрок.
23. В игре на ориентированном графе два игрока поочередно накрывают белыми и черными фишками его вершины. Ход считается допустимым, если в заданную вершину ведет дуга из той вершины, в которой располагается фишка противоположного цвета. Первым ходом белые накрывают любую вершину. Проигрывает тот, кто не может сделать очередного хода. Определить, является ли начальная комбинация выигрышной для белых.
24. Дан набор слов. Составить из них цепочку максимальной длины по количеству слов (или по количеству букв). Цепочка образуется в том случае, если первая буква следующего слова совпадает с последней буквой предыдущего слова. Повторно использовать слова нельзя.
25. Найти максимальную длину кольца, построенного из слов, содержащих одинаковое количество гласных и согласных букв.
26. Дана схема лабиринта. Необходимо найти все возможные варианты путей выхода из лабиринта без пересечений.
27. По двум конвейерам двигаются молочные бутылки. Для каждой бутылки известно время заполнения и закупоривания. Найти расстановку бутылок, при которой время обработки минимально.
28. Некто, стоя на остановке, в течение часа отмечал интервалы прибытия автобусов. Какое минимальное количество автобусов могло работать на линии в это время и каков интервал их движения? Например, если отмеченные интервалы – $\{3, 10, 10, 10, 10, 10\}$, это означает, что работал один автобус, который в первый раз появился через три минуты после того, как некто начал отмечать интервалы и далее появлялся с интервалом в 10 минут, т. е. время его появления на остановке следующее – $\{3, 13, 23, 33, 43, 53\}$. Если отмеченные интервалы соответствуют набору чисел $\{2, 10, 5, 15, 3, 12, 11\}$, это означает, что работало два автобуса. Первый из них появился через две минуты от начала отсчета и далее ходил с интервалом в 15 минут, т. е. время его появления на остановке – $\{2, 17, 32, 47\}$. Второй автобус впервые появился в 12 минут, а интервал его движения – 23 минуты, т. е. он появлялся в следующие моменты времени: $\{12, 35, 58\}$. Следовательно, общее время появления двух автобусов – $\{2, 12, 17, 32, 35, 47, 58\}$, что как раз соответствует интервалам $\{2, 10, 5, 15, 3, 12, 11\}$.
29. Задача о пробирках. Даны три пробирки объемом 100 мл. Две из них имеют шкалу рисок. Уровни рисок указываются пользователем. Из начального состояния $\{100 \text{ мл}, 0 \text{ мл}, 0 \text{ мл}\}$ (в первой пробирке 100 мл, а две другие пусты) определить, возможен ли переход путем переливания жидкости из одной пробирки в другую в состояние $\{_, _, 1 \text{ мл}\}$ (в пробирках с рисками — произвольное количество миллилитров, а в пробирке без рисок — 1 мл).
30. Дан граф. Определите последовательность вершин, через которые необходимо пройти, чтобы нарисовать этот граф. Нельзя отрывать карандаш от бумаги и проходить по одной и той же дуге несколько раз.
31. Задача о рюкзаке. Дано N предметов, у каждого есть вес и цена. Предложить набор предметов максимальной стоимости, помещающихся в рюкзак, рассчитанный на 50 кг.

Задания на тему «Внутренние сортировки»

1. Задан массив записей. Поле *key* каждой записи – целое число. Написать программу, которая наглядно демонстрирует сортировку массива по ключу *key*:
 - а) методом простого слияния;
 - б) методом естественного слияния.Количество элементов массива таково, что все элементы отображаются на экране. В данных сортировках используется дополнительный массив.
2. Исследовать сортировки простым и естественным слиянием, построить графики зависимости количества сравнений и количества перестановок от количества элементов массива ($n = 1 \dots 100$) для этих двух сортировок. Использовать функцию, которая по заданному массиву и количеству элементов в этом массиве подсчитывает, сколько потребуется сравнений (перестановок).
3. Задан массив записей. Поле *key* каждой записи – целое число. Реализовать пирамидальную сортировку по ключу *key*. Визуализировать процесс сортировки.
 - А. Массив изображен в виде последовательности элементов.
 - Б. При построении пирамиды на экране массив отображается не только в виде последовательности, но и в виде построенной пирамиды.
4. Написать программу, которая наглядно иллюстрирует работу обменной поразрядной сортировки для следующих типов данных:
 - а) целого;
 - б) символьного;
 - в) строкового (короткого).
5. Написать программу, иллюстрирующую работу сортировки Хоара:
 - а) реализовать рекурсивным методом;
 - б) реализовать нерекурсивным методом;
 - в) реализовать любым из методов, но учитывать, что для сортировки массива маленького размера лучше применять какой-нибудь другой метод сортировки (например, простыми вставками, метод «пузырька»...).
6. Проиллюстрировать работу сортировки Бэтчера.
Указание: обратите внимание на то, что эта сортировка параллельная, т. е. сравнение или перестановка элементов должны выполняться одновременно (параллельно).
7. Написать программу, которая наглядно иллюстрирует работу следующих методов сортировки:

- а) «пузырек»;
- б) шейкерная.

Провести сравнение сортировок методом «пузырька» и шейкерной по количеству сравнений, по количеству обменов. Для этого построить графики зависимостей данных величин от количества элементов массива.

8. Дан текстовый файл, состоящий из слов, разделенных пробелами. Отсортировать слова в этом файле методом вставки в список с вычислением адреса.
9. Написать программу, которая иллюстрирует работу метода Шелла с одной из формул вычисления шага сортировки:
 - а) $h[k-1] = 3h[k] + 1, h[1] = 1, t = \lceil \log_3 n \rceil - 1$;
 - б) $h[k-1] = 2h[k] + 1, h[1] = 1, t = \lceil \log_2 n \rceil - 1$;
 - в) $2^k - 1$;
 - г) $2^k + 1$;
 - д) $(2^k - (-1)^k) / 3$;
 - е) $(3^k - 1) / 2$;
 - ж) числа Фибоначчи.
10. Написать программу, которая исследует зависимость количества сравнений (количества перестановок) в методе Шелла от выбора разных формул для вычисления шага. Изобразить соответствующие графики функций.
11. Реализовать сортировку массива целых чисел методом двухпутевых вставок при использовании следующих дополнительных структур данных:
 - а) массива;
 - б) двунаправленного списка.Визуализировать работу данного алгоритма.
12. Написать программу, которая наглядно иллюстрирует работу сортировок простыми и бинарными вставками.
13. Исследовать зависимость количества сравнений в сортировках простыми и бинарными вставками от количества элементов в этих массивах. При этом отобразить необходимое количество перестановок.
14. Дан массив записей, поле *key* которого – целые числа. Написать программу, которая иллюстрирует работу сортировки по ключу *key* методом подсчета. При этом должны отображаться не только сравниваемые элементы, но и вспомогательный массив, в котором хранятся результаты сравнений. Перестановки элементов осуществляются в исходном массиве.

15. Написать программу, которая иллюстрирует сортировку массива распределяющим подсчетом. Элементом массива является запись следующего типа:
- ```
Record
 Ch : char;
 Key : integer;
End;
```
- Указание:* ключом сортировки является поле целого типа.
16. Для каждого из пунктов обмена валюты известны курс покупки и курс продажи доллара. Вывести список пунктов в порядке:
- уменьшения выгоды продажи;
  - уменьшения выгоды покупки;
  - убывания среднеквадратического отклонения от курса доллара на ММВБ.
17. В магазине строительных материалов в продаже имеются стеновые панели, для которых известны ширина, длина, количество штук, цена за  $1 \text{ м}^2$ . Вывести в порядке возрастания цены сведения о тех стеновых панелях, общая площадь которых не меньше заданной.
18. Дан список морфем, в котором указана сама морфема и ее тип (приставка, корень, суффикс или окончание). В списке могут встречаться повторяющиеся морфемы. Написать программу, результатом работы которой должен быть список морфем, разбитый на группы для каждого типа. Внутри этих групп морфемы должны быть расположены в лексикографическом порядке. В группе корней одна и та же морфема может встречаться несколько раз, а в других группах морфемы должны быть уникальными.
19. Есть некий измерительный прибор, работа которого зависит от входных параметров  $\alpha$  и  $x$ , а результат определяется следующей формулой  $y = \alpha \cdot \sin(\alpha x) \cdot \cos^2(x/\alpha)$ . Проводится серия опытов для значений  $x_1, x_2, \dots, x_n$ ,  $\alpha = \text{const}$ . Вывести результат в виде таблицы, упорядоченной по убыванию значений показаний прибора, полученных в ходе опытов.
20. В массиве произвольно хранятся сведения о жильцах (ФИО, адрес, телефон). Упорядочить эти данные по полю ФИО в алфавитном порядке. Реализовать следующие возможности:
- добавление сведений о новом жильце;
  - удаление сведений о жильце;
  - редактирование сведений о жильце.
- Эти действия не должны нарушать упорядоченность массива.

*Указание:* использовать алгоритмы, изложенные в методах сортировок простыми или бинарными вставками.

21. Информация агентства по продаже недвижимости содержит следующие сведения о квартирах: район, в котором находится квартира, этаж, количество комнат, общая площадь, цена за  $1 \text{ м}^2$ . Клиент, обращаясь в агентство, имеет возможность указать вес для каждого из критериев (важный критерий имеет большой вес, незначительный — маленький), а агентство, в свою очередь, предлагает ему список квартир, упорядоченный по невозрастанию суммы весов.
22. Дана целочисленная квадратная матрица размера  $n$ . Упорядочить значения так, чтобы  $a_{11} \leq a_{12} \leq \dots \leq a_{1n} \leq a_{21} \leq a_{22} \leq \dots \leq a_{2n} \leq \dots \leq a_{n1} \leq a_{n2} \leq \dots \leq a_{nn}$ .
23. В чемпионате России по футболу принимают участие 16 команд. Для каждой команды известен список игроков, каждый игрок из команды имеет свой рейтинг. Вывести список команд в порядке убывания вероятности победы в чемпионате. Вероятность победы равна рейтингу команды — сумме рейтингов 11 лучших игроков.
24. Дано:
- ```
const n = 20;
type город = (A, B, C, D, E, F, G, H);
города = set of город;
маршрут = record
  гор : города;
  цена : integer;
end;
тур_фирма = array [1..n] of маршрут;
var ТФ : тур_фирма; Цена : integer;
```
- Вывести список маршрутов, не превышающих по цене заданное значение *Цена*, в порядке убывания.

Задания на тему «Внешние сортировки»

Указание: каждая из задач (1–3) должна предоставлять следующие возможности:

- создание нового файла;
 - дополнение существующего файла;
 - просмотр сведений из файла;
 - выполнение основной задачи, указанной в упражнении.
1. Частное предприятие «Петров и К^о» занимается приобретением у изготовителей шоколадных конфет в упаковках (коробках) и их продажей частным лицам и организациям. Для улучшения контроля за приходом и расходом товара предприятие ведет компьютерный учет. Очень часто покупателям данной фирмы требуются те или иные сведения о товаре, например, наиболее свежие конфеты или изготовленные каким-то определенным производителем. Сведения о конфетах включают в себя следующие данные:
- название;
 - вес упаковки;
 - цена упаковки;
 - изготовитель;
 - дата выпуска;
 - срок хранения.
- A. Вывести сведения о конфетах, упорядочив их по названию. Использовать двухпутевое однофазное естественное несбалансированное слияние.
- B. Вывести сведения о конфетах, которые не дороже цены, заданной покупателем, и отсортированы в порядке повышения цены. Использовать многопутевое однофазное простое слияние.
- C. Вывести данные о просроченном товаре, если известна текущая дата, и упорядочить их по полю «изготовитель». Использовать многопутевое двухфазное простое слияние с внутренней сортировкой.
- D. Вывести сведения о конфетах с заданным весом упаковки, упорядочив их по дате выпуска (сначала самые свежие). Использовать двухпутевое двухфазное естественное сбалансированное слияние.
- E. Вывести названия конфет и их изготовителей в порядке увеличения срока хранения конфет. Использовать многопутевое однофазное естественное несбалансированное слияние.
- F. Вывести названия и дату истечения срока годности, упорядочив данные в порядке убывания отношения веса упаковки к ее цене. Использовать двухпутевое двухфазное простое слияние.

- G. Вывести в алфавитном порядке названия конфет, выпускаемых заданным производителем, с указанием даты выпуска и срока хранения конфет. Использовать многопутевое однофазное простое слияние с внутренней сортировкой.
2. Концерн «Видеосервис» предоставляет услуги по прокату и продаже видеофильмов частным лицам и организациям. Сведения о фильмах, имеющихся в наличии, хранятся в электронном виде. Для облегчения выбора заказа покупателю предоставляется возможность сократить каталог просмотра путем задания даты выхода фильма или, например, киностудии, выпустившей этот фильм. Сведения о фильмах включают в себя следующие данные:
- название;
 - год выпуска;
 - киностудия;
 - режиссер;
 - длительность фильма;
 - наличие приза;
 - три главных героя.
- A. Вывести в алфавитном порядке сведения о видеофильмах с указанием киностудий, их выпустивших в заданный период времени (для периода задан год начала и год конца). Использовать многопутевое двухфазное естественное несбалансированное слияние.
- B. Распечатать названия фильмов, в которых снимался какой-то конкретный актер, в обратном порядке их выхода в свет (сначала самые свежие). Использовать многопутевое двухфазное простое слияние.
- C. Вывести сведения о киностудиях, выпустивших фильмы в 2000 г., и о кинорежиссерах, работающих на этих студиях. Данные о киностудиях должны быть упорядочены в алфавитном порядке, и для каждой киностудии сведения о режиссерах также должны быть упорядочены по алфавиту. Использовать двухпутевое однофазное естественное сбалансированное слияние.
- D. Вывести сведения о фильмах, имеющих награды, упорядочив их по длительности просмотра. Использовать двухпутевое однофазное простое слияние с внутренней сортировкой.
- E. Упорядочить названия фильмов по фамилиям актеров, принимающих в них участие. Использовать двухпутевое двухфазное естественное несбалансированное слияние.
3. В ООН имеется полный перечень всех стран, который включает в себя:
- название;
 - континент;
 - столицу;
 - площадь;

- численность населения;
- государственный строй.

В разных ситуациях (например для оказания помощи некоторой стране) требуется выбрать из всего списка те или иные государства. Так как сведения о странах хранятся в электронном виде, то часто бывает необходимо решать задачи, подобные нижеизложенным.

- A. Указать сведения о государствах заданного континента в порядке возрастания численности населения. Использовать двухпутевое однофазное простое слияние.
 - B. Упорядочить по названиям столиц сведения о государствах с площадью меньше заданной. Использовать двухпутевое двухфазное простое слияние с внутренней сортировкой.
 - C. Указать названия государств с заданным государственным строем в алфавитном порядке. Использовать многопутевое однофазное естественное сбалансированное слияние.
 - D. Вывести названия и столицы государств в порядке убывания плотности населения. Применить многопутевое двухфазное естественное сбалансированное слияние.
4. Написать программу, которая осуществляет сортировку слов текстового файла (одно слово в одной строке) путем слияния данных из файла и некоторой заданной внутренней структуры:
- a) массива;
 - b) линейного списка;
 - c) дерева.

Указание:

1. Исходный текстовый файл генерируется случайным образом: количество слов в нем случайно (до 10000), и сами слова сгенерированы при помощи датчика случайных чисел.
2. Сортировка осуществляется следующим способом: даны два вспомогательных файла и некоторая структура данных, например массив. Сначала первая порция данных из исходного файла упорядоченно перекачивается в заданную внутреннюю структуру. Для массива данные заносятся методом простых или бинарных вставок, для списка — методом вставок в список; для дерева — построением дерева-списка.
3. На следующем этапе из массива (списка, дерева) упорядоченные данные переписываются во вспомогательный файл. Затем очередная порция данных из исходного файла упорядоченно переносится в массив (список, дерево), после чего осуществляется слияние данных из внутренней структуры и ранее заполненного вспомогательного файла. Результат слияния записывается в другой вспомогательный файл. Так продолжается до тех пор, пока все данные не будут отсортированы.

5. Исследовать один из методов сортировки, указанный в задачах с 1 по 3, по заданному критерию (например по количеству сравнений, по количеству проходов, по времени выполнения сортировки). Для этого необходимо:

- a) отсортировать файл целых чисел, сформированный случайным образом, и определить значения критерия;
- b) отсортировать файл целых чисел, заданный в обратном порядке, и определить значения критерия.

Пункты a) и b) нужно выполнить для файлов с различным количеством элементов (их определяет пользователь, например: 1000, 5000, 10000). Результат вывести в виде таблицы.

Таблица. Формат вывода результатов

Количество элементов	Значение критерия	
	Случайный файл	Обратный файл
1000		
5000		
10000		

- a) визуализировать процесс распределения серий по вспомогательным файлам (должно быть отображено количество реальных серий, записываемых в каждый из вспомогательных файлов на данном уровне, и количество фиктивных серий, остающихся во вспомогательных файлах);
 - b) визуализировать процесс слияния серий, причем должны выделяться вспомогательные файлы, участвующие в слиянии, и отображаться изменение количества серий (и реальных, и фиктивных) в этих файлах (не сами элементы файлов).
7. Реализовать алгоритм каскадной сортировки. Кроме того:
- a) визуализировать процесс распределения серий по вспомогательным файлам (должно быть отображено количество реальных серий, записываемых в каждый из вспомогательных файлов на данном уровне, и количество фиктивных серий, остающихся во вспомогательных файлах);
 - b) визуализировать процесс слияния серий, причем должны выделяться вспомогательные файлы, участвующие в слиянии, и отображаться изменение количества серий (и реальных, и фиктивных) в этих файлах (не сами элементы файлов).

8. В файле хранится последовательность русских слов. Упорядочить ее в алфавитном порядке.

Указание:

1. Применить внешнюю сортировку.
2. Учесть, что порядок кодов букв русского алфавита не соответствует порядку букв в алфавите.

Список литературы

1. Ахо А. Структуры данных и алгоритмы : пер. с англ. / А. Ахо, В. Хопкрофт, Д. Ульман. — М. : Издательский дом "Вильямс", 2003. — 384 с.
2. Вирт Н. Алгоритмы и структуры данных / Н. Вирт. — СПб. : Невский диалект, 2001. — 352 с.
3. Задачи по программированию / С. А. Абрамов [и др.]. — М. : Наука. Физ.-мат. лит., 1988. — 224 с.
4. Иванова Г. С. Объектно-ориентированное программирование : учебник для вузов. — 2-е изд., перераб. и доп. / Г.С. Иванова, Т.Н. Ничупкина, Е.К. Пугачев ; под ред. Г.С. Ивановой. — М. : Изд-во МГТУ им. Н.Э. Баумана, 2003. — 368 с.
5. Касьянов В. Н. Сборник заданий по практикуму на ЭВМ / В. Н. Касьянов, В. К. Сабельфельд. — М. : Наука, 1986. — 272 с.
6. Кнут Д. Искусство программирования : пер. с англ. / Д. Кнут. — М. : Вильямс, 2000. — Т. 3: Сортировка и поиск. — 822 с.
7. Мейер Д. Методы программирования / Д. Мейер, К. Бодуэн. — М. : Мир, 1985. — Т. 1, 2.
8. Пильщиков В.И. Сборник упражнений по языку Паскаль : учеб. пособие / В. Н. Пильщиков. — М. : Наука. Физ.-мат. лит., 1989. — 160 с.
9. Программирование алгоритмов обработки данных / О.Ф. Ускова [и др.]. — СПб. : БХВ — Санкт-Петербург, 2003. — 192 с.
10. Программирование на языке Паскаль: задачник / О.Ф. Ускова [и др.] ; под ред. О.Ф. Усковой. — СПб. : Питер, 2002. — 336 с.
11. Сибуя М. Алгоритмы обработки данных / М. Сибуя, Т. Ямамото ; пер. с япон. — М. : Мир, 1986. — 224 с.

Учебное издание

**ПРАКТИКУМ
ПО КУРСУ «СТРУКТУРЫ И АЛГОРИТМЫ
КОМПЬЮТЕРНОЙ ОБРАБОТКИ ДАННЫХ»**

Учебно-методическое пособие

Составители:

**Воронина Ирина Евгеньевна,
Огаркова Наталья Владимировна,**

Подписано в печать 10.03.2009. Формат 60×84/16. Усл. печ. л. 2,0.
Тираж 50 экз. Заказ 300.

Издательско-полиграфический центр
Воронежского государственного университета.
394000, г. Воронеж, пл. им. Ленина, 10. Тел. (факс) +7 (4732) 598-026
<http://www.ppc.vsu.ru>; e-mail: ppc_center@ppc.vsu.ru

Отпечатано в типографии Издательско-полиграфического центра
Воронежского государственного университета.
394000, г. Воронеж, ул. Пушкинская, 3