## Практическая работа №44 Чтение структур из файла

Постройте программу, которая работает с базой данных в виде типизированного файла. Если нужно, за основу можно принять файл-заготовку database.pas.

Ваша СУБД (система управления базой данных) должна уметь выполнять такие действия:

1. просмотр записей;
2. добавление записей;
3. удаление записей.

Предусмотрите обработку ошибочных ситуаций:

1. для новой записи нет места в массиве;
2. введен неправильный номер удаляемой записи.

Содержание базы данных можно выбрать на свой вкус. Например, в ней могут храниться

* сведения об аудио- или видеозаписях;
* каталог библиотеки;
* каталог художественных фильмов;
* данные об автомашинах;
* данные о членах спортклуба;
* данные о животных в зоопарке и т.п.

*Дополнительные задания.*

1. Предусмотрите какой-нибудь вариант шифрования данных, так чтобы их нельзя было просмотреть в текстовом редакторе.
2. Добавьте к возможностям вашей СУБД экспорт данных в тестовый файл в формате XML.

## Сортировка структур с помощью указателей

Добавьте в программу СУБД, которую вы составили в предыдущей работе, возможность сортировки записей по разным полям. Используйте сортировку с помощью указателей.

## Динамические массивы

1. Введите с клавиатуры число N и вычислите все простые числа в диапазоне от 2 до N, используя решето Эратосфена.
2. Введите с клавиатуры число N и запишите в массив первые N простых чисел.
3. Введите с клавиатуры число N и запишите в массив первые N чисел Фибоначчи (напомним, что они задаются рекуррентной формулой,).
4. Напишите функцию, которая находит максимальный элемент переданного ей динамического массива.
5. Напишите подпрограмму, которая находит максимальный и минимальный элементы переданного ей динамического массива (используйте изменяемые параметры).
6. Напишите рекурсивную функцию, которая считает сумму элементов переданного ей динамического массива.
7. Напишите функцию, которая сортирует значения переданного ей динамического массива, используя алгоритм «быстрой сортировки» (см. учебник 10 класса).

## Расширяющиеся динамические массивы

1. В файле записано неизвестное количество целых чисел. Вывести эти числа в порядке возрастания в другой файл. Начальный размер массива – 100 элементов.
2. Доработайте программы из практической работы 43 (*Ввод и вывод структур*) так, чтобы они работали для любого количества записей. Начальный размер массива – 100 элементов.
3. Доработайте программу из практической работы 44 (*Чтение структур из файла*) так, чтобы она работали для любого количества записей (с возможностью расширения). Начальный размер массива – 10 элементов.

## Алфавитно-частотный словарь

1. Постройте программу, которая составляет алфавитно-частотный словарь для заданного файла со списком слов (см. § 41 учебника).
2. \*В предыдущей программе измените функцию *Find* так, чтобы в ней использовался двоичный поиск.
3. В предыдущей программе объедините функции *Find* и *FindPlace*, заменив их на одну функцию. Если слово найдено в списке, функция работает так же, как *Find*: возвращает номер слова в списке. Если слово не найдено, функция должна вернуть отрицательное число: номер элемента массива, перед которым нужно вставить слово, со знаком минус.
4. \*В предыдущей задаче вывести все найденные слова в файл в порядке убывания частоты, то есть в начале списка должны стоять слова, которые встречаются в файле чаще всех.

## Модули

1. В программе, которая составляет алфавитно-частотный словарь (см. предыдущую практическую работу), вынесите все операции со списком в отдельный модуль.
2. Доработайте программу из практической работы 44 (*Чтение структур из файла*) так, чтобы все функции и процедуры, работающие с базой данных, были вынесены в отдельный модуль.

## Вычисление арифметических выражений

1. Напишите программу, которая вычисляет с помощью стека значение арифметического выражения, записанного в постфиксной форме. Выражение вводится с клавиатуры в виде символьной строки.
2. \*Найдите в литературе или в Интернете алгоритм перевода арифметического выражения из инфиксной формы в постфиксную, и напишите программу, которая решает эту задачу.

## Проверка скобочных выражений

1. Напишите программу, которая проверяет правильность скобочного выражения с четырьмя видами скобок: (), [], {} и <>. Все операции со стеком вынесите в отдельный модуль.

## Заливка области

1. Напишите программу, которая выполняет заливку одноцветной области заданным цветом. Матрица, содержащая цвета пикселей, вводится из файла. Затем с клавиатуры вводятся координаты точки заливки и цвет заливки. На экран нужно вывести матрицу, которая получилась после заливки. Все операции с очередью вынесите в отдельный модуль.
2. \*Перепишите предыдущую программу – используйте статический массив для организации очереди. Считайте, что в очереди может быть не более 100 элементов. Предусмотрите обработку ошибки «очередь переполнена».
3. \*Напишите решение задачи о заливке области, в котором точки, добавленные в очередь, как-то помечаются, чтобы не добавлять их повторно. В чём преимущества и недостатки такого алгоритма?
4. \*\*Найдите в литературе или в Интернете описание формата BMP. Напишите программу, которая выполняет заливку одноцветной области рисунка в формате заданным цветом и сохраняет получившийся файл на диске. Координаты точки заливки и цвет заливки вводятся с клавиатуры. Все операции с очередью и рисунками в формате BMP вынесите в отдельные модули.

## Вычисление арифметических выражений

1. Напишите программу, которая вводит и вычисляет арифметическое выражение без скобок. Все операции с деревом вынесите в отдельный модуль.
2. Добавьте в предыдущую программу процедуры обхода построенного дерева так, чтобы получить префиксную и постфиксную запись введенного выражения.
3. \*Добавьте в предыдущую программу процедуру обхода дерева в ширину.
4. \*Усовершенствуйте программу (см. задачу 1), чтобы она могла вычислять выражения со скобками.
5. \*Включите в вашу программу обработку некоторых ошибок (например, два знака операций подряд). Поработайте в парах: обменяйтесь программами с соседом и попробуйте найти выражение, при котором его программа завершится аварийно (не выдаст сообщение об ошибке).

## Хранение двоичного дерева в массиве

1. Измените программу для вычисления арифметического выражения (см. предыдущую практическую работу), так чтобы дерево хранилось в виде массива. Все операции с деревом вынесите в отдельный модуль.

## Алгоритм Прима-Крускала

1. Напишите программу, которая вводит из файла весовую матрицу графа и строит для него минимальное остовное дерево.
2. Оцените асимптотическую сложность алгоритма Крускала.

*Ответ*:

## Алгоритм Дейкстры

1. Напишите программу, которая вводит из файла весовую матрицу графа, затем вводит с клавиатуры номера начальной и конечной вершин и определяет оптимальный маршрут.

Оцените асимптотическую сложность алгоритма Дейкстры.

*Ответ*:

## Алгоритм Флойда-Уоршелла

1. Напишите программу, которая вводит из файла весовую матрицу графа и определяет длины всех оптимальных маршрутов с помощью алгоритма Флойда-Уоршелла.

Оцените асимптотическую сложность алгоритма Флойда-Уоршелла.

*Ответ*:

1. \*Напишите программу, которая решает задачу 5, приведённую в конце § 44 учебника (задача о размещении школы). Для определения кратчайших путей используйте алгоритм Флойда-Уоршелла. Весовую матрицу графа вводите из файла.

## Числа Фибоначчи

1. Напишите программу для вычисления N-ого числа Фибоначчи, использующую рекурсию. Проверьте ее работу при различных значениях N и постройте (например, с помощью электронных таблиц) график зависимости времени счёта от N.

Оцените асимптотическую сложность этого алгоритма.

*Ответ*:

1. Напишите программу для вычисления N-ого числа Фибоначчи, использующую динамическое программирование. Проверьте ее работу при различных значениях N и постройте (например, с помощью электронных таблиц) график зависимости времени счёта от N. Сравните эти данные с рекурсивным вариантом программы:

|  |  |  |
| --- | --- | --- |
| N | Время счёта | |
| рекурсия | динамическое программирование |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Оцените асимптотическую сложность алгоритма, использующего динамическое программирование.

*Ответ*:

## Задача о куче

1. Напишите программу, которая решает задачу о куче (задачу 3, рассмотренную в тексте § 45). Данные вводятся из текстового файла в следующем формате:

* в первой строке записано количество камней N и требуемый вес кучи W*;*
* во второй строке перечислены веса камней pi (i = 1, ..., N).

Считайте, что N ≤ 100 и W ≤ 100.

Результаты работы программы выводятся в текстовый файл в следующем формате:

* в первой строке выводится количество используемых камней и вес полученной кучи;
* во второй строке выводятся веса выбранных камней.

1. \*Измените программу так, чтобы память под используемые массивы выделялась динамически. Допущение о том, что N ≤ 100 и W ≤ 100 не используйте.
2. \*Напишите программу, которая решает задачу о ранце (см. задачу 3 после § 45).

## Количество программ

1. Напишите программу, которая определяет количество программ для преобразования числа A в число B для исполнителя *Утроитель* (см. задачу 4, рассмотренную в тексте § 45). Числа A и B вводятся с клавиатуры. Память под массивы должна выделяться динамически.
2. Напишите программу программ для преобразования числа A в число B для исполнителя *Калькулятор*. У него есть три команды, которым присвоены номера:

1. прибавь 1

2. умножь на 2

3. умножь на 3

Числа A и B вводятся с клавиатуры.

1. Напишите программу программ для преобразования числа A в число B для исполнителя *Калькулятор*. У него есть три команды, которым присвоены номера:

1. прибавь 1

2. умножь на 2

3. возведи в квадрат

Числа A и B вводятся с клавиатуры.

1. Напишите программу программ для преобразования числа A в число B для исполнителя *Калькулятор*. У него есть две команды, которым присвоены номера:

1. прибавь 1

2. увеличь вторую с конца цифру на 1

Первая из них увеличивает число на экране на 1, вторая – увеличивает на 1 число десятков. Если перед выполнением команды 2 вторая с конца цифра равна 9, она не изменяется. Числа A и B вводятся с клавиатуры.

1. Напишите программу программ для преобразования числа A в число B для исполнителя *Калькулятор*. У него есть две команды, которым присвоены номера:

1. прибавь 1

2. увеличь каждый разряд числа на 1

Например, число 23 с помощью команды 2 превратится в 34, а 29 в 39 (так как младший разряд нельзя увеличить). Числа A и B вводятся с клавиатуры.

## Размен монет

1. Напишите программу, которая решает задачу о размене монет (задачу 5, рассмотренную в тексте § 45). Данные вводятся из текстового файла в следующем формате:

* в первой строке записано количество типов монет N и требуемая сумма W*;*
* во второй строке перечислены достоинства монет pi (i = 1, ..., N).

Считайте, что N ≤ 10 и W ≤ 100 и в наборе монет есть монета достоинством pi = 1.

Программа должна вывести на экран количество вариантов сдачи.

1. \*Измените программу так, чтобы память под используемые массивы выделялась динамически. Допущение о том, что N ≤ 10 и W ≤ 100 не используйте.
2. \*Измените программу так, чтобы она работала и в том случае, если в наборе монет нет монеты достоинством pi = 1. Допущение о том, что N ≤ 10 и W ≤ 100 не используйте.