

Лабораторная работа №4 «Обработка одномерных массивов»

Необходимо спроектировать и реализовать на языке С программу, осуществляющую по запросам пользователя ввод, обработку и вывод последовательности данных, которая представляется в виде массива.

Программа должна иметь следующую функциональность:

1. Ввод данных:

- Чтение из стандартного потока ввода («с клавиатуры»).
- Чтение из текстового файла.
- Генерация случайных данных.

2. Вывод данных:

- Запись в стандартный поток вывода («на экран»).
- Запись в текстовый файл.

3. Обработка данных:

- Вставка нового элемента в массив по индексу.
- Вставка нового элемента в упорядоченный массив без нарушения его упорядоченности.
- Удаление заданного количества элементов, начиная с некоторого индекса.
- Сортировка массива.

4. Таймирование.

Примечания:

1. Взаимодействие программы с пользователем должно быть выстроено при помощи диалогового меню. Штатный способ завершения работы программы — через соответствующий пункт диалогового меню.
2. Программа должна осуществлять проверку корректности вводимых данных и, в случае ошибок, выдавать соответствующие сообщения, после чего продолжать работу.
3. Программа должна предоставлять пользователю возможность ввода *любых* необходимых данных: имен файлов, индексов, способов сортировки, параметров таймирования и т.д.
4. Обрабатываемые последовательности должны быть представлены в виде массива элементов, которые имеют подходящий тип данных. В том случае, если обрабатываются последовательности данных, представленных в виде сложных объектов, то для их описания необходимо разработать собственный составной тип данных — структуру.

5. Работа с текстовыми файлами должна осуществляться при помощи функций стандартной библиотеки `fopen()`, `fclose()`, `fprintf()`, `fscanf()`.
6. Реализуемые функции сортировки должны иметь тот же интерфейс, что и функция `qsort()` из состава стандартной библиотеки.
7. Логически законченные части алгоритма решения задачи должны быть оформлены в виде отдельных функций с параметрами. Использование глобальных переменных не допускается.
8. Исходные коды программы должны быть логичным образом разбиты на несколько файлов.
9. Для сборки программы необходимо разработать `Makefile` и использовать утилиту `make`.
10. Программа должна корректным образом работать с памятью, для проверки необходимо использовать соответствующие программные средства, например: `valgrind` (при тестировании и отладке программы ее необходимо запускать командой вида `valgrind ./lab4`, а при анализе производительности — `./lab4`).

Отчётность по выполнению лабораторной работы:

1. Блок-схемы алгоритмов работы основных функций обработки массива.
2. Исходные коды программы.
3. Тестовые наборы для программы.
4. Результаты таймирования, содержащие таблицы, графики зависимости времени выполнения сортировок от количества сортируемых элементов и выводы об оценке сложности рассмотренных алгоритмов сортировки и её совпадении с теоретическими ожиданиями.
5. Сравнительный анализ времени, потраченного на решение задачи программами п. 1 и п. 2.