

Лабораторная работа №2 «Работа со стеком, очередью, двусторонней очередью»

Необходимо спроектировать и разработать на языке C библиотеку, реализующую логическую структуру данных «стек», «дек» или «очередь», и прикладную программу, которая её использует.

Данные для обработки программой вводятся из входного потока. Каждая строка, вводимая пользователем, должна обрабатываться отдельно. Завершение ввода (завершение работы программы) определяется концом файла.

В ходе выполнения задачи должны быть разработаны:

1. Библиотека, реализующая логическую структуру данных «стек», «дек» или «очередь» на основе вектора и на основе списка. Т.е. должно быть реализовано два варианта решения, если не указано иное.
2. Прикладная программа, использующая разработанную библиотеку.

Примечания:

1. Библиотека должна состоять из нескольких файлов, среди которых обязательно должен быть заголовочный файл, определяющий интерфейс взаимодействия с разрабатываемой структурой данных. Данный интерфейс должен быть общим для обеих реализаций.
2. Использование глобальных переменных не допускается.
3. Исходные коды должны быть логичным образом разбиты на несколько файлов (необходимо использовать как *.c-файлы, так и *.h-файлы).
4. Выбор конкретной реализации структуры данных должен осуществляться на этапе сборки путём указания соответствующих флагов.
5. Программа должна корректным образом обрабатывать все данные, которые может ввести пользователь.
6. Реализация логической структуры данных «стек», «дек» или «очередь» на основе вектора должна обеспечивать корректную работу в рамках области памяти переменного размера, которая единожды выделяется на этапе создания экземпляра структуры, т.е. при добавлении новых записей может произойти переполнение, данная ситуация должна обрабатываться штатным образом, если в задании не оговорено иное.
7. Использование массивов переменной длины (VLA — variable length arrays) не допускается.
8. Реализация логической структуры «стек», «дек» или «очередь» на основе списка должна обеспечивать корректную работу с произвольным количеством записей, т.е. при добавлении новых записей переполнение невозможно.
9. При реализации на основе списка должно отдаваться предпочтение списочным структурам, использующим меньшее количество памяти — односвязные предпочтительнее двусвязных, кольцевые предпочтительнее линейных, если это целесообразно в контексте задачи.

10. Программа должна корректным образом работать с памятью, для проверки необходимо использовать соответствующие программные средства, например: `valgrind`, санитайзеры, встроенные в IDE средства и т.д.