Лабораторная работа #5

Работа с динамической памятью. Окружение и аргументы командной строки

[malloc, calloc, realloc, free, new, delete, args]

ЛАБОРАТОРНАЯ РАБОТА # 5
Работа с динамической памятью. Окружение и аргументы командной строки

Цель работы

Закрепить знания и практические навыки эффективного использования памяти при проектировании и реализации интерактивных приложений на языке программирования C/C++.

Требования

1) Разработать многофайловый консольный проект на C/C++ согласно варианту задания с использованием шаблона (паттерна) проектирования MVC.
2) Входные данные для программы должны вводиться в виде набора строк! Если в задании сказано про «*слова*» – это значит, что вводится должна одна строка, которая состоит из несколько слов. Если сказано про «*строки*», то должен вводиться набор строк, каждая из которых состоит минимум из двух трёх слов.
3) Предусмотреть два способа инициализации данных: с помощью аргументов командной строки и пользовательского ввода (при запуске программа должна проверять наличие параметров и использовать их для выполнения своего алгоритма при их наличии, в противном случае, запрашивать соответствующие данные у пользователя).
4) Программа должна выводить на консоль исходные данные и конечный результат.
5) ЗАПРЕЩАЕТСЯ в программе использовать под любым предлогом ГЛОБАЛЬНЫЕ переменные!
6) Для повышения производительности программы и закрепления навыков работы с памятью везде, где это возможно, необходимо использовать ДИНАМИЧЕСКОЕ выделение и освобождение памяти, а также осуществлять работу через УКАЗАТЕЛИ.

7) Каждое задание оформить в виде отдельной бизнес-функции. Все функции должны быть сгруппированы по соответствующим отдельным файлам.
8) Все функции должны быть самодостаточные, т.е. при их разработке необходимо придерживаться принципа ***Single Responsibility Principle***.
9) При выполнении задания разрешается использовать *IDE*, а также задействовать любой текстовый редактор (к примеру, ***gedit***) и набор компиляторов GNU Compiler Collection (GCC), в частности, компиляторы языков программирования C/C++ ***gcc/g++***.
10) Для автоматизации сборки проекта необходимо использовать утилиту-автосборщик – ***GNU make***.

Основное задание

Написать программу «Анаграммы» (anagrams), которая реализует популярную словесную игру. Игра «Анаграммы» очень интересная и необычная игра в которой нет ни картинок ни слов. Суть игры (программы) заключается в следующем: формируется группа слов, компьютер случайным образом выбирает одно из слов и случайным образом переставляет в нём буквы, а затем представляет пользователю (игроку). Цель игрока – угадать выбранное компьютером слово.

Индивидуальное задание

1) Найти самую короткую и самую длинную строки. Вывести найденные строки и их длину.
2) Упорядочить и вывести строки в порядке возрастания (убывания) значений их длины.
3) Вывести на консоль те строки, длина которых меньше (больше) средней, а также длину.
4) Найти слово, в котором число различных символов минимально. Если таких слов несколько, найти первое из них.
5) Найти количество слов, содержащих только символы латинского алфавита, а среди них – количество слов с равным числом гласных и согласных букв.

6) Найти слово, символы в котором идут в строгом порядке возрастания их кодов. Если таких слов несколько, найти первое из них.
7) Найти слово, состоящее только из различных символов. Если таких слов несколько, найти первое из них.

Best of LUCK with it, and remember to HAVE FUN while you're learning :)

Victor Ivanchenko

Контрольные вопросы

1. На какие укрупнённые блоки (области или сегменты) делиться виртуальное адресное пространство, которое выделяется операционной системой для каждой программы?
2. Для чего каждый из блоков нужен, что там хранится и в какой момент определяется его размер?
3. Что такое класс памяти (storage class) переменной и что он определяет?
4. Какие существуют разновидности классов памяти в языках программирования C/C++ и их основные отличия?
5. Какие есть классы памяти у переменных? Какой из классов памяти и при каких условия используется по умолчанию (неявно) при объявлении переменной?
6. Какие есть классы памяти у функций? Какой из классов памяти и при каких условия используется по умолчанию (неявно) при описании функции?
7. Что такое указатели и зачем они нужны при программировании?
8. Опишите основные операторы (операции), которые применяются для работы с указателями.
9. Что такое ссылка и чем указатели отличаются от ссылок в языке программирования C++?
10. Для чего необходимо использовать динамическое распределение памяти?
11. Опишите основные функции стандартной библиотеки языка программирования C и особенность их использования для работы с динамической памятью: *malloc*, *calloc*, *realloc*, *free* и т.д.
12. Опишите основные операторы языка программирования C++ для работы с динамической памятью: *new* и *delete*.
13. Какие бывают ошибки при работе с динамической памятью?
14. Какие существуют способы передачи параметров в функцию в языках программирования C/C++?
15. Что происходит с оригинальной переменной, передаваемой в качестве параметра в функцию, и памятью, выделенной для программы, в момент вызова соответствующей функции?

16. Что такое «утечка памяти» (memory leak)? Когда, где и по чьей вине это может возникнуть?
17. Как создать одномерный массив конкретного типа данных? Как объявить массив, который может хранить элементы различных типов данных?
18. Какие существуют способы для работать с одномерными динамическими массивами?
19. Опишите способы динамического выделения и освобождения памяти для многомерного массива (минимум три)? Какой из способов эффективней и почему?
20. Какие существуют способы для работать с многомерными динамическими массивами?