МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет информационных технологий и робототехники

Кафедра программного обеспечения информационных систем и технологий

**Отчет по лабораторной работе № 3**

по дисциплине: «Системное программирование»

на тему: «Многофайловые проекты и шаблон MVC»

Выполнила: ст. гр. 10702221

Баяманова А.К.

Проверил: Давыденко Н.В.

Минск 2023

**ЛАБОРАТОРНАЯ РАБОТА # 3**

**Многофайловые проекты и шаблон MVC**

**Цель работы**

Научиться грамотно проектировать и разрабатывать многофайловые проектына языке программирования С/C++ с использованием архитектурного шаблона проектирования и разработки гибких и масштабируемых приложений Model-View-Controller (MVC).

**Требования**

1) Разработать многофайловый консольный проект на С/С++ согласно варианту задания с использованием шаблона (паттерна) проектирования MVC.

2) Размерность матрицы в индивидуальном задании задаётся пользователем на стадии выполнения программы. Для повышения производительности программы использовать динамическое выделение памяти.

3) ЗАПРЕЩАЕТСЯ в программе использовать под любым предлогом ГЛОБАЛЬНЫЕ переменные!

4) Каждое задание оформить в виде отдельной бизнес-функции.

5) Все функции должны быть сгруппированы по соответствующим отдельным файлам.

6) Все функции должны быть самодостаточные, т.е. при их разработке необходимо придерживаться принципа Single Responsibility Principle.

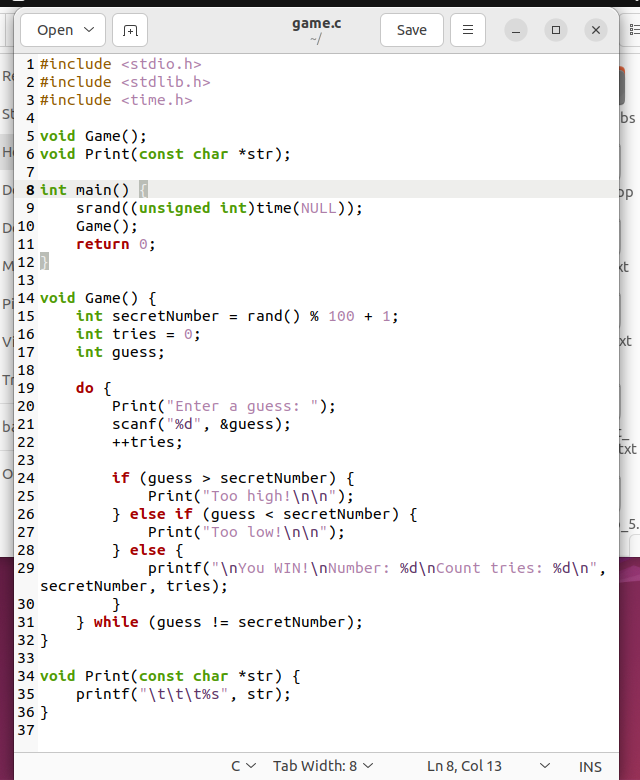
7) При выполнении задания запрещается использовать интегрированные средства разработки (Integrated Development Environment, IDE). Рекомендуется задействовать любой текстовый редактор (к примеру, gedit) и набор компиляторов GNU Compiler Collection (GCC), в частности, компиляторы языков программирования С/C++ gcc/g++, а для отладки – gdb-отладчик.

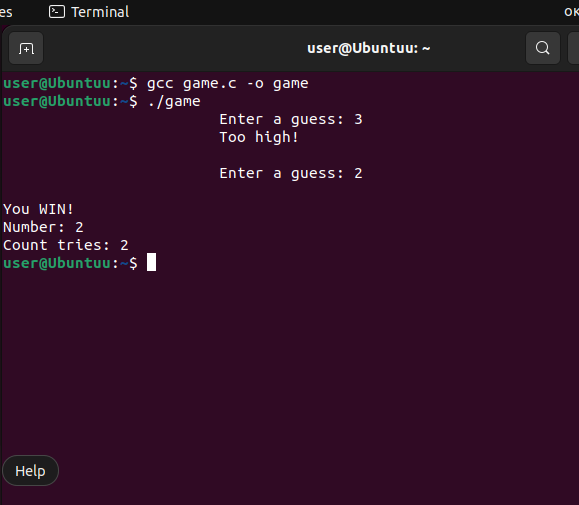
8) При разработке программ необходимо придерживаться соглашения по написанию кода на C/C++ (Code-Convention).

**Основное задание**

Разработать интерактивную программу «Try to Guess the Number» («Попробуй угадать число»), которая эмулирует классическую игру на отгадывание числа. Суть программы (игры) сводиться к следующему: компьютер генерирует случайное число из диапазона, к примеру, от 1 до 100, а пользователь (далее игрок) пытается отгадать число за минимальное количество попыток. При каждой очередной попытке компьютер «подсказывает» игроку, как соизмеряется предложенный вариант игрока с действительным загаданным компьютером числом: загаданное число больше или меньше указанного (higher/lower). Как только игрок отгадывает число, компьютер должен «поздравить» его с выводом на экран угаданного числа и количество затраченных игроком попыток. Далее компьютер может «предложить» повторно сыграть игру или выйти из программы.

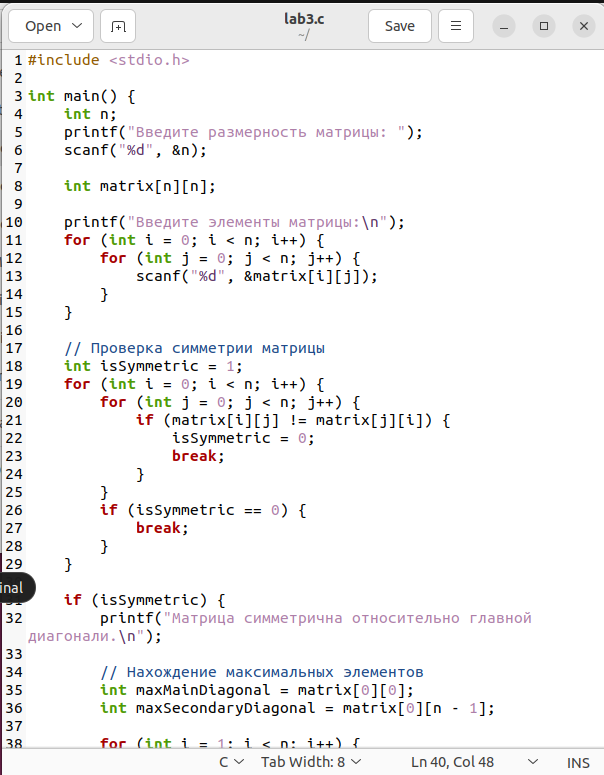
Для универсальности предложенной программы можно добавить возможность выбора диапазона генерирования компьютером случайных чисел, а также задания ограничения на количество попыток. В случае, если игрок не укладывается в заданное количество попыток (т.е. проигрывает), программа должна выводить суровую надпись «Game Over»

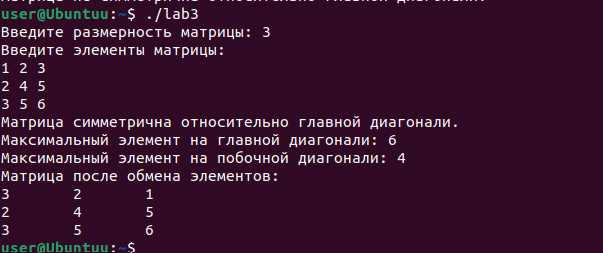




**Индивидуальное задание**

2) Определить, является ли матрица симметричной (относительно главной диагонали). Вывести на экран соответствующее сообщение. Найти максимальный элемент среди стоящих на главной и побочной диагонали и поменять местами с элементом, стоящим на пересечении этих диагоналей.





**Контрольные вопросы**

**2. Какие существенные преимущества имеют многофайловые проекты перед однофайловыми? Что такое сборка?**

Многофайловые проекты (проекты, состоящие из нескольких исходных файлов) обычно имеют несколько существенных преимуществ перед однофайловыми проектами:

Модульность и организация: В многофайловых проектах код разделяется на логически связанные модули, что упрощает организацию и управление проектом. Каждый файл может содержать отдельную функциональность или компонент программы, что делает код более читаемым и поддерживаемым.

Повторное использование: Многие части кода могут быть многократно использованы в разных частях проекта и даже в других проектах. Это способствует повторному использованию кода и сокращению объема работы.

Параллельная разработка: В многофайловых проектах несколько программистов могут работать над разными частями проекта параллельно, что повышает эффективность и скорость разработки.

Тестируемость: Модули можно тестировать независимо друг от друга, что упрощает обнаружение и устранение ошибок.

Удобство работы с библиотеками: Многофайловая структура обеспечивает возможность создания и использования библиотек, которые могут быть использованы в разных проектах.

Что касается понятия "сборка", то оно связано с процессом преобразования исходного кода в исполняемый код или другие формы выходных файлов. Процесс сборки включает в себя компиляцию исходных файлов, связывание объектных файлов и библиотек в исполняемый файл или библиотеку, а также другие этапы, такие как оптимизация и упаковка.

В многих языках программирования и средах разработки сборка происходит автоматически с использованием инструментов сборки, таких как компиляторы, средства управления зависимостями и средства автоматизации сборки (например, Make, CMake, Ant, Maven и т. д.).

Сборка - это необходимый этап для создания исполняемых программ или библиотек, и она позволяет программистам создавать сложные проекты, состоящие из множества файлов, с минимальными усилиями.