**Лабораторная работа №1**

**Знакомство с CMake и Clang**

**Цель работы:** Получение практических навыков в работе с инструментом автоматической сборки проектов CMake и транслятором Clang. А также закрепление теоретических знаний об этапах разработки программ.

**Постановка задачи:**

**Приложение Lab-01C:**

Разработать приложение на языке C содержащее решение для задачи из индивидуального задания\*. При чём код должен быть кроссплатформенным (т.е. компилироваться для Windows и Linux/macOS без внесения дополнительных изменений).

**Приложение Lab-01CS:**

Разработать приложение аналогичное первому, но на языке C#.

Подготовить исполняемые файлы, полученные двумя разными способами, а именно:

* с использованием транслятора Clang и компилятора С# напрямую;
* с использованием CMake.

**Данный пункт касается только приложения Lab-01C**!!! При использовании Clang напрямую должны быть получены промежуточные файлы сборки, т.е. файлы, которые получаются на выходе каждого этапа сборки (после препроцессинга, после компиляции, после компоновки).

При использовании CMake должно быть выполнено следующее:

* Проект для сборки должен включать в себя два приложения на C и C#;
* На ОС семейства Linux/macOS проект С# должен игнорироваться при сборке (должно быть прописано в виде условия);
* Сборка должна быть «out-of-source»;
* Для Windows использовать генератор «Visual Studio» (для тех, у кого нет VS проще использовать генератор «[Ninja](https://ninja-build.org/)»), для Linux – «Unix Makefiles», для macOS – «XCode»;
* В конфигурации CMake (CMakeLists.txt) должны быть использованы переменные для определения текущей ОС, а также вывод сообщения в логи CMake сообщения вида «Hello, I am [название ОС]!».

Разработанные приложения должны быть скомпилированы и протестированы как на Windows, так и на ОС из семейства Linux или macOS (для владельцев MacBook’ов).

**Дополнительно (совершенно не обязательно, но огромный плюс):**

Продемонстрировать кросс-компиляцию проекта (в любом направлении: Windows->Linux/macOS или Linux/macOS->Windows).

Собрать проект на C# через CMake на Linux/macOS.

**Индивидуальные задания:**

Варианты заданий

1. Дано вещественное число x. Пользуясь только умножением, сложением и вычитанием, вычислить:

2x4 – 3x3 + 4x2 – 5x + 6.

Разрешается использовать не более четырех умножений и четырех сложений и вычитаний.

2. Целой переменной k присвоить значение, равное третьей от конца цифре в записи целого положительного числа x.

3. Целой переменной k присвоить значение, равное сумме цифр в записи целого положительного трехзначного числа x.

4. Целой переменной k присвоить значение, равное первой цифре дробной части в записи вещественного положительного числа x.

5. Определить число, полученное выписыванием в обратном порядке цифр заданного целого трехзначного числа.

6. Идет n-ная секунда суток. Определить, сколько полных часов и полных минут прошло к этому моменту.

7. Дано вещественное число x. Не пользуясь никакими операциями, кроме умножения, получить x21 за шесть операций.

8. Дано вещественное число x. Не пользуясь никакими операциями, кроме умножения, получить x3 и x10 за четыре операции.

9. Дано вещественное число x. Не пользуясь никакими операциями, кроме умножения, получить x5 и x13 за пять операций.

10. Дано вещественное число x. Не пользуясь никакими операциями, кроме умножения, получить x2, x5 и x17 за шесть операций.

11. Дано вещественное число x. Не пользуясь никакими операциями, кроме умножения, получить x4, x12 и x28 за шесть операций.

12. «Упаковать» четыре символа в беззнаковое целое. Длина беззнакового целого равна 4.

13. «Распаковать» беззнаковое целое число в четыре символа. Длина беззнакового целого равна 4.

14. Заменить в целочисленной переменной x n бит, начиная с позиции p, n старшими инвертированными битами целочисленной переменной y.

15. Циклически сдвинуть значение целочисленной величины на n позиций вправо.

16. Циклически сдвинуть значение целочисленной величины на n позиций влево.