Задания к работе №1 по кодированию и защите информации.

Все задания выполняются на языке программирования C++ (стандарт C++14 и выше).

Допускается использование библиотеки стандартных шаблонов (STL).

Применение готовых реализаций алгоритмов защиты информации и библиотек/фреймворков/прочих сторонних решений, содержащих такие реализации, не допускается.

Аварийное завершение работы реализованных компонентов и приложений не допускается.

Весь реализованный Public API компонентов должен быть покрыт модульными (unit-) тестами.

Реализация должна быть опубликована в репозиторий на GitHub.

Крайний срок публикации реализаций на GitHub: 19.02.2026 09:00 GMT+3.

1. Реализуйте компонентный метод для выполнения перестановки битов в рамках переданного значения (тип значения - массив байтов). Параметры метода: значение для перестановки его битов, правило перестановки (*P*-блок, заданный в виде массива индексов битов), правила индексирования битов (заданные в виде значений перечисления): порядок индексирования (от младшего бита к старшему или наоборот); индекс отсчёта (*0* или *1*). Продемонстрируйте работу реализованного функционала.

2. Реализуйте два перегруженных компонентных метода для выполнения замены битов в рамках переданного значения (тип значения - массив байтов). Параметры первого метода: значение для замены его битов, правило замены (*S*-блок, заданный в виде ассоциативного контейнера). Параметры второго метода: значение для замены его битов, правило замены (функциональный объект (*std::function*<...>). Продемонстрируйте работу реализованного функционала.

3. Реализовать набор компонентных методов, обеспечивающих:
    a. побитовый циклический сдвиг *n*-битового значения на *k* битов влево;
    b. побитовый циклический сдвиг *n*-битового значения на *k* битов вправо;
    c. применение *k*-битовой маски к *n*-битовому значению
    d. получение значения, состоящего из битов исходного значения, начиная с *i*-го и заканчивая *j*-м включительно оба конца;
    e. обмен местами *i*-го и *j*-го бита *n*-битового значения;
    f. установление *0*/*1* *i*-го бита *n*-битового значения.

    Продемонстрируйте работу реализованного функционала.

4. Реализовать клиент-серверный комплекс приложений, реализующий бизнес-логику по выполнению шифрующего преобразования данных алгоритмом RC4. Клиентское приложение должно иметь возможность передавать ключ шифрования и данные произвольного размера серверу, который выполняет шифрование/дешифрование и предоставляет клиентскому приложению возможность получить результат операции шифрования/дешифрования. Сервер должен иметь возможность одновременно обрабатывать 100 сессий шифрования/дешифрования. Клиент-серверное взаимодействие организуйте с использованием средств синхронизации уровня ядра ОС: разделяемая память, семафоры. Продемонстрируйте работу комплекса приложений, выполнив шифрование и последующее дешифрование различных файлов (текстовых, изображений, видео, аудио, пустого файла и т. д.) различными ключами шифрования, с проверкой на получение оригинальных данных после выполнения над данными операций шифрования и дешифрования (для этого реализуйте функцию сравнения файлов).