

## Задание № 1

**Срок сдачи:** не позднее 17 марта.

**Логические схемы.** Напишите программу на языке C++, которая будет моделировать работу и рисовать на экране одну из следующих логических схем:

- а) 4-х разрядный декодер;
- б) 4-х разрядная схема сдвига;
- в) 4-х разрядный компаратор;
- г) 4-х разрядный полный сумматор;
- д)\* 4-х разрядный регистр на основе D-триггеров.

Обеспечить ввод входных и вывод выходных сигналов (консоль, файл или GUI), а также перерисовку схемы при изменении входных сигналов.

*Совет:* используйте материал осеннего семестра.

**Конечные автоматы.** Постройте конечный автомат, который получает на вход двоичную запись натурального числа  $x$ , начиная с младшего разряда, и выводит двоичную запись числа:

- а)  $x + 3$ ;
- б)  $3 \cdot x$ ;
- в)  $3x + 2$ ;
- г)  $x \cdot (x \bmod 3)$ .

**Машины Тьюринга.** Обозначим как  $N_0$  множество всех неотрицательных чисел. Опишите машины Тьюринга, которые реализуют:

1. Сложение двух чисел из  $N_0$ , данных в двоичной системе счисления.
2. Умножение двух чисел из множества  $N_0$ , записанных на ленте в виде последовательности единиц, а именно:

$0 \rightarrow 0, 1 \rightarrow 01, 2 \rightarrow 011, 3 \rightarrow 0111, 4 \rightarrow 01111, \dots$

Назовём эту запись *единичной записью* числа. Числа записаны на ленте подряд.

**Алгоритмы Маркова.** Записать алгоритмы Маркова, которые реализуют:

1. см. машину Тьюринга **1**.
2. Удвоение числа заданного:
  - а) в виде единичной записи,
  - б) в двоичной системе счисления.

## Задание № 2

**Срок сдачи:** не позднее 12 мая.

**Взаимодействие C/C++ и языка ассемблера.** Дан текст (последовательность символов), содержащий не более 100 элементов. Признаком конца текста считается символ с кодом '\0'. Требуется:

- ввести текст с клавиатуры и записать его в память ЭВМ;
- определить, обладает ли этот текст заданным свойством, указанным в вашем варианте задания;
- преобразовать текст по *правилу 1* вашего задания, если он обладает заданным свойством, и по *правилу 2* в противном случае;
- вывести на экран исходный и преобразованный тексты, а также номер и формулировку применённого правила.

### Требования к программе

1. Ввод и вывод, выделение памяти, тестовая оснастка и логика выбора действий реализуются на языке C/C++.
2. Действия: проверка свойства, применение правила №1 и правила №2 преобразования текста, — реализуются на языке ассемблера для архитектуры Intel x86\_64 в синтаксисе AT&T. Каждое действие следует разместить в отдельном модуле (файле). Реализация действия может состоять из нескольких функций (процедур). Формулировки свойства текста и правил его преобразования для вашего варианта следует разместить в комментариях соответствующих модулей (в начале файла).
3. Вывод исходного текста должен быть выполнен сразу после ввода, до анализа и преобразования.
4. Вывод преобразованного текста должен быть выполнен только после завершения преобразования.
5. Программа должна сохранять работоспособность при любых входных данных.
6. Необходимо разработать «тестовую оснастку» для выполнения автоматического тестирования.

**Варианты** **выдаются преподавателем.**

### Свойство текста (пример)

1. Текст оканчивается заглавной латинской буквой, которая больше не встречается в тексте.
2. ...

### Правило № 1 (пример)

1. Заменить каждую заглавную латинскую букву следующей по алфавиту ( $A \rightarrow B, B \rightarrow C, \dots, Z \rightarrow A$ ).
2. ...

### Правило № 2 (пример)

1. Перенести в начало текста все входящие в него цифры с сохранением порядка их следования.
2. ...