УТВЕРЖДЕН

ХХХ.ХХХХХХХХ.ХХХХХ-01 90 01-ЛУ

**Лабораторная работа №7**

**Пояснительная записка**

**ХХХ.ХХХХХХХХ.ХХХХХ-01 90 01**

**Листов 15**

2021

СОДЕРЖАНИЕ

[1. Введение 3](#__RefHeading___Toc1126_2197296794)

[1.1. Наименование программы 3](#__RefHeading___Toc1128_2197296794)

[1.2. Краткая характеристика области применения 3](#__RefHeading___Toc1130_2197296794)

[1.3. Постановка задачи 3](#__RefHeading___Toc1132_2197296794)

[2. Основания для разработки 4](#__RefHeading___Toc1134_2197296794)

[3. Назначение разработки 5](#__RefHeading___Toc1136_2197296794)

[3.1. Описание объекта исследования 5](#__RefHeading___Toc1138_2197296794)

[3.2. Функциональное назначение 5](#__RefHeading___Toc1140_2197296794)

[4. Требования к программе или программному изделию 6](#__RefHeading___Toc1142_2197296794)

[4.1. Требования к функциональным характеристикам 6](#__RefHeading___Toc1144_2197296794)

[4.1.1. Требования к составу выполняемых функций 6](#__RefHeading___Toc1146_2197296794)

[4.1.2. Требования к организации входных и выходных данных 6](#__RefHeading___Toc1148_2197296794)

[4.2. Требования к надежности 6](#__RefHeading___Toc1150_2197296794)

[4.3. Условия эксплуатации 6](#__RefHeading___Toc1152_2197296794)

[4.4. Требования к составу и параметрам технических средств 6](#__RefHeading___Toc1154_2197296794)

[4.5. Требования к информационной и программной совместимости 6](#__RefHeading___Toc1156_2197296794)

[4.6. Требования к маркировке и упаковке 6](#__RefHeading___Toc1158_2197296794)

[4.7. Требования к транспортированию и хранению 6](#__RefHeading___Toc1160_2197296794)

[4.8. Специальные требования 6](#__RefHeading___Toc1162_2197296794)

[5. Требования к программной документации 7](#__RefHeading___Toc1164_2197296794)

[6. Технико-экономические показатели 8](#__RefHeading___Toc1166_2197296794)

[6.1. Эффективность и ресурсоёмкость 8](#__RefHeading___Toc1168_2197296794)

[6.2. Стоимость 8](#__RefHeading___Toc1170_2197296794)

[7. Стадии и этапы разработки 9](#__RefHeading___Toc1172_2197296794)

[8. Порядок контроля и приемки 10](#__RefHeading___Toc1174_2197296794)

[9. ОПИСАНИЕ алгоритма 11](#__RefHeading___Toc1176_2197296794)

[9.1. main.c 11](#__RefHeading___Toc1178_2197296794)

[9.2. Вызов Машины Тьюринга 12](#__RefHeading___Toc1180_2197296794)

[9.3. Машина Тьюринга 13](#__RefHeading___Toc1182_2197296794)

[10. Код программы 14](#__RefHeading___Toc2280_2197296794)

[11. Cкриншоты с контрольными примерами 17](#__RefHeading___Toc977_2323423047)

[12. Выводы 18](#__RefHeading___Toc1184_2197296794)

[Перечень принятых сокращений 19](#__RefHeading___Toc1186_2197296794)

1. Введение
   1. Наименование программы

Наименование программы – «Машина Тьюринга»

* 1. Краткая характеристика области применения

Программа будет запущена один раз, во время сдачи лабораторной работы, после чего забудется навсегда.

* 1. Постановка задачи

1. Объединиться в подгруппы по 2-3 студента.

2. Изучить примеры решения задач с помощью МТ в файле «МТ (решенные задачи).pdf».

3. Программно реализовать представленные задачи.

4. Подготовить отчет о ЛР, включающий постановку задачи, блок-схемы алгоритмов, листинг программы, скриншоты с контрольными примерами и выводы по работе.

5. Защитить лабораторную работу.

Cрок выполнения 4 часа – выполнение ЛР, 2 часа защита ЛР.

Задания:

1)Последовательность натуральных чисел (x1, x2, .., xn) задается на ленте машины Тьюринга как слово 01^x1 01^x20...01^xn0, где 1^x обозначает слово 11...1, состоящее из x единиц. Предполагается, что остальные клетки ленты содержат нули. Построить машину Тьюринга, осуществляющню заданное преобарзование. В начале работы головка показывает на 0 перед крайтей левой едеинцей, и машина находится в состоянии q1. (x1, x2) (x1, x2, x1)

2)Построить машину Тьюринга, которая вычисляет модуль разности любых двух натуральных чисел.

3)Задать определения: МТ, правильно вычисляющей предикат; МТ, вычисляющая предикат с восстановлением. Построить МТ для правильного вычисления предиката :

P = u, x - нечетное число

P = л, x - четное число

1. Основания для разработки

Основанием для разработки является задание, выданное преподавателем М.А.Бирюковым, именуемым в дальнейшем Заказчиком, В.Е. Дунаеву, Т.В. Хохлову и А.В. Крылову, именуемым в дальнейшем Исполнителем.

Согласно заданию, Исполнитель обязан разработать и защитить систему не позднее 14.05.2022, предоставить исходные коды и документацию не позднее 14.05.2022

Наименование темы разработки – «Разработка проекта для работы с МТ»

1. Назначение разработки
   1. Описание объекта исследования

Работа некоторых алгоритмов Машины Тьюринга

* 1. Функциональное назначение

Программа демонстрирует результаты выполнения последовательности команд для МТ по данному заданию.

1. Требования к программе или программному изделию
   1. Требования к функциональным характеристикам
      1. Требования к составу выполняемых функций

После запуска программы пользователю отображается сообщение «Введите номер задания [1 — 3] Для завершения работы введите 0». Ожидается, что пользователь введёт целое число в диапазоне от 0 до 3.

Для пользователя программа предоставляет три примера работы МТ:

- Преобразование строки по заданному правилу

- Вычислить модуль разности двух натуральных чисел

- Вычисление предиката

----------------------------------------

* + 1. Требования к организации входных и выходных данных

При вводе номера задания принимаются целые числа в диапазоне от 0 до 3.

* 1. Требования к надежности

Вероятность безотказной работы должна составлять не менее 99.99% при условии исправности компьютера, на котором запускается программа (отсутствие программных и физических неполадок).

* 1. Условия эксплуатации

Программа запускается на компьютере с установленной операционной системой Linux.

* 1. Требования к составу и параметрам технических средств

Состав технических средств:

* + Компьютер оператора, включающий в себя:
  + процессов x86 с тактовой частотой не менее 1 ГГц;
  + оперативную память объёмом не менее 1 Гб;
  + видеокарту, мышь, монитор.
  1. Требования к информационной и программной совместимости

Необходимо наличие файлов «main.c», «funcs.h» и «funcs.c» в одной директории для корректной работы.

* 1. Требования к маркировке и упаковке

Программное изделие передаётся по сети Internet или на съёмном носителе информации.

* 1. Требования к транспортированию и хранению

Специальных требований не предъявляется.

* 1. Специальные требования

Отсутствуют

1. Требования к программной документации

Предварительный состав программной документации:

* + техническое задание (включает описание применения);
  + код программы, блок-схема.

1. Технико-экономические показатели
   1. Эффективность и ресурсоёмкость

Объём занимаемой памяти: не более 1 Мб (файлы «main.c», «funcs.h» и «funcs.c»)

* 1. Стоимость

Разработано в образовательных целях (бесплатно).

1. Стадии и этапы разработки

Разработка должна быть проведена в три стадии:

* + техническое задание;
  + технический проекты;
  + демонстрация.

На стадии «Техническое задание» должен быть выполнен этап разработки и утверждения настоящего технического задания.

На стадии «Технический проект» должны быть выполнены перечисленные ниже этапы работ:

* + разработка программы;
  + разработка программной документации;
  + испытания программы.

На стадии «Демонстрация» должен быть выполнен этап разработки «Подготовка и передача программы».

Содержание работ по этапам:

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

* + постановка задачи;
  + определение и уточнение требований к техническим средствам;
  + определение требований к программе;
  + определение стадий, этапов и сроков разработки программы и документации на нее;
  + утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями ГОСТ 19.101-77.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

* + разработка, согласование и утверждение порядка и методики испытаний;
  + проведение приемо-сдаточных испытаний;
  + корректировка программы и программной документации по результатам испытаний.

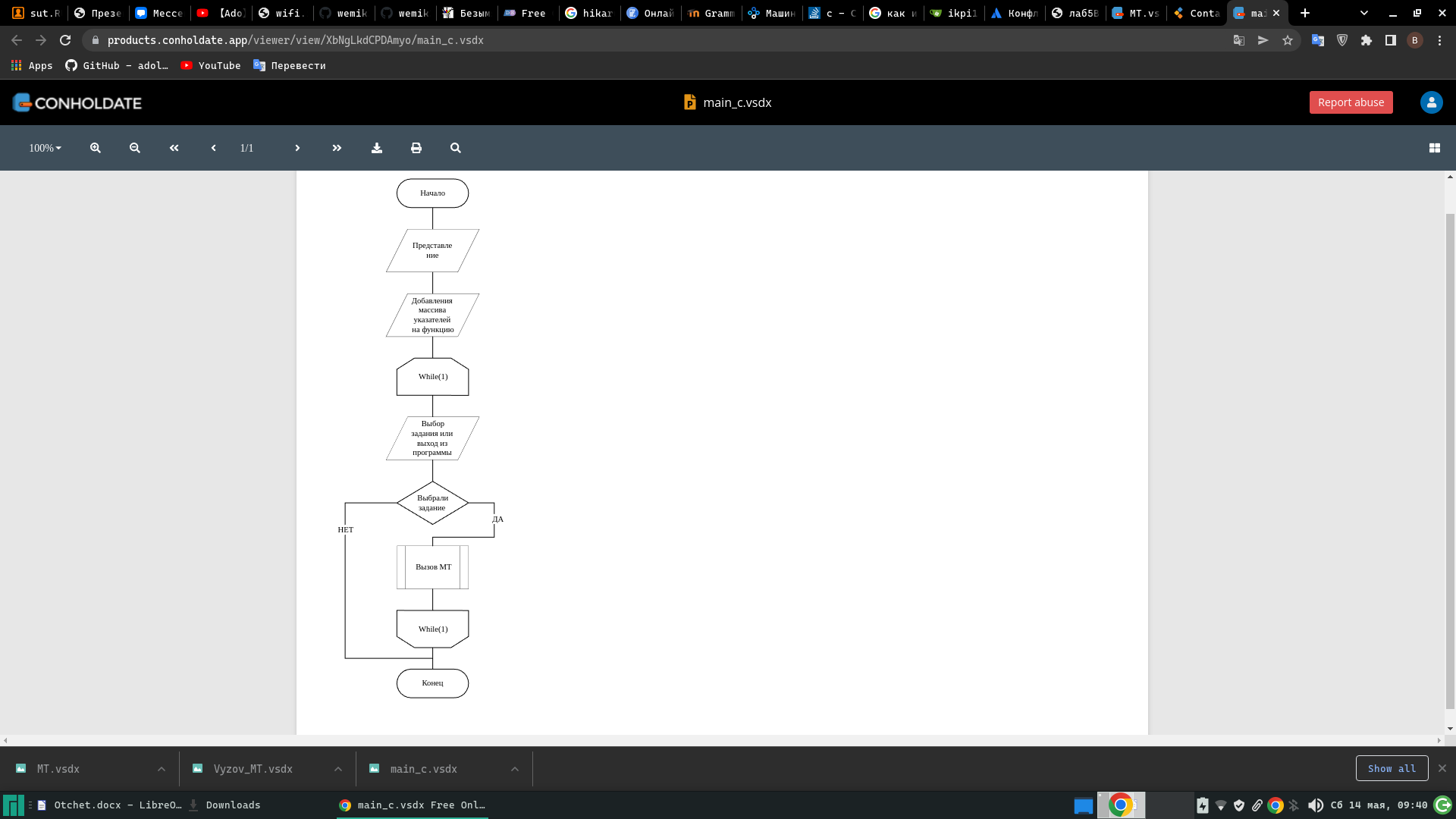
На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию на объектах заказчика.

1. Порядок контроля и приемки

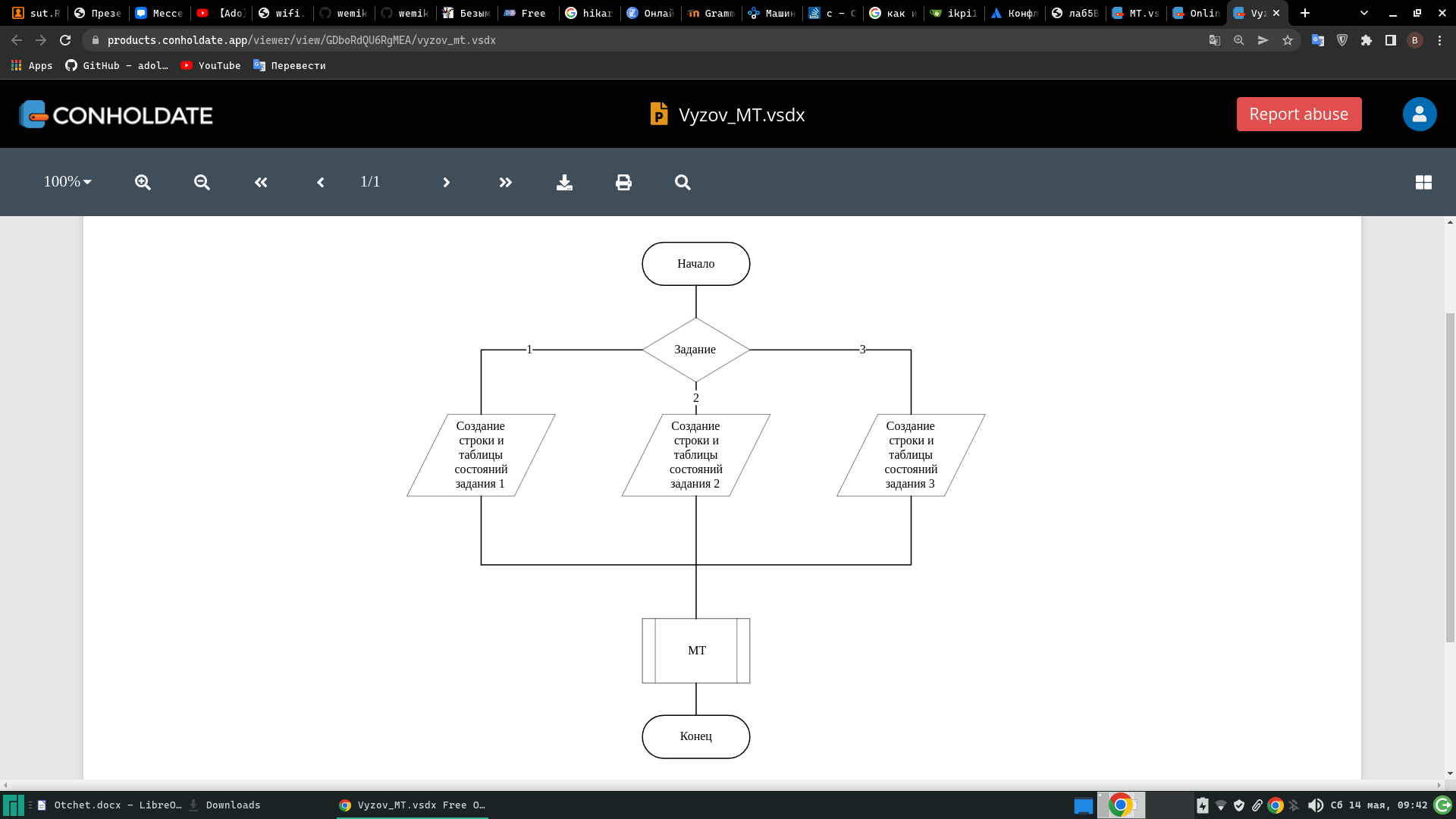
Приемосдаточные испытания программы могут проводиться согласно разработанной исполнителем и согласованной заказчиком «Программы и методики испытаний».

Ход проведения приемо-сдаточных испытаний заказчик и исполнитель могут документировать (например, в протоколе испытаний).

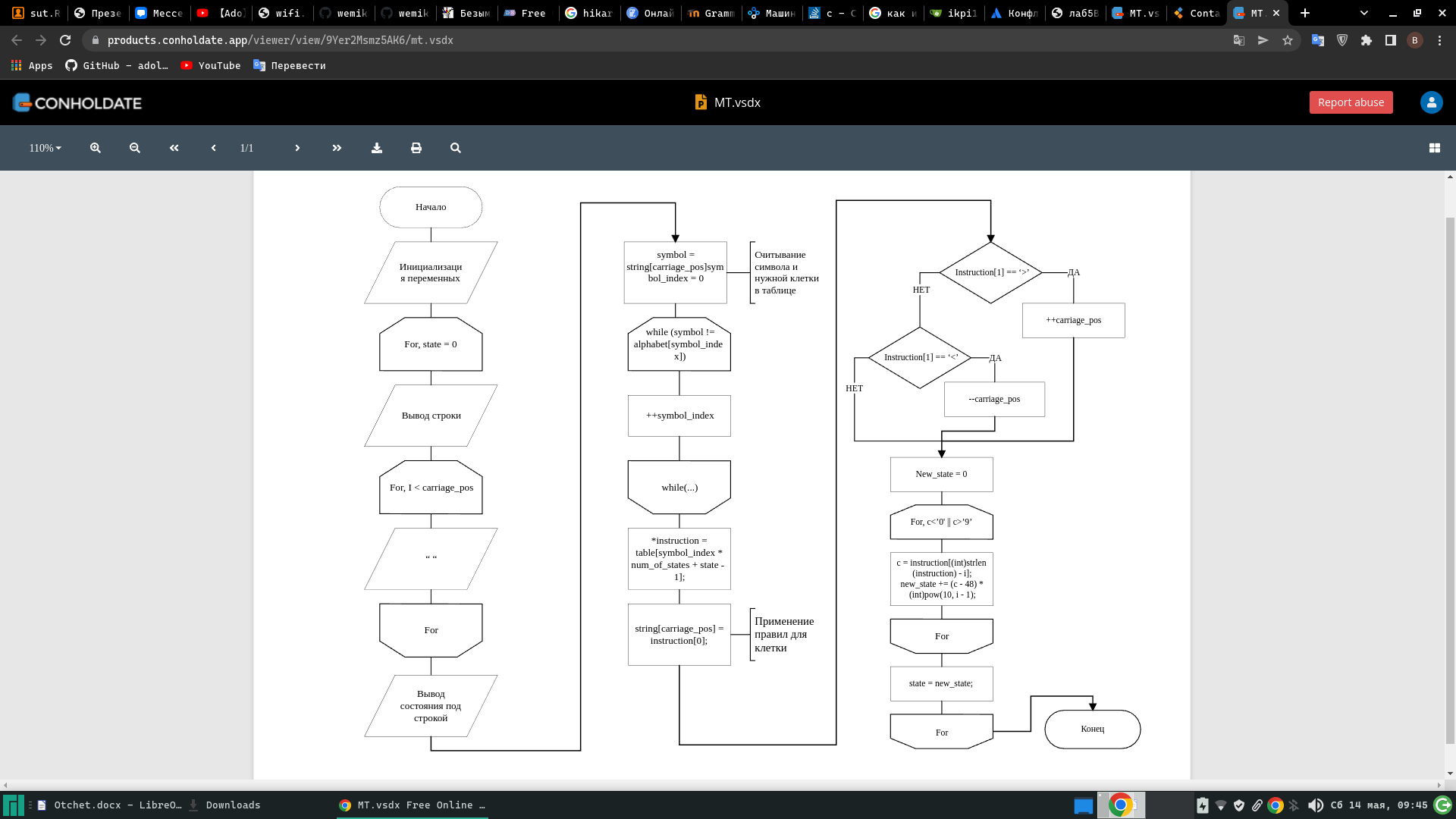
1. ОПИСАНИЕ алгоритма
   1. main.c



* 1. Вызов Машины Тьюринга



* 1. Машина Тьюринга



1. Код программы

**10.1 main.c**

#include <stdio.h>

#include "funcs.h"

int main() {

puts(

"Курсовая работа по АОПИ, ИКПИ-14\n"

"Авторы: Крылов Артём, Хохлов Тихон, Дунаев Владислав\n");

void (\*funcs[])() = {emul\_1, emul\_2, emul\_3};

while (1) {

int n;

puts(

"Введите номер задания [1-3]:\n"

"Для завершения работы введите 0");

scanf("%d", &n);

if (!n) break;

funcs[n - 1]();

puts("");

}

return 0;

}

**10.2 funcs.h**

#pragma once

void emul\_1();

void emul\_2();

void emul\_3();

**10.3 funcs.c**

#include "funcs.h"

#include <math.h>

#include <stdio.h>

#include <string.h>

void emul(char \*alphabet, char \*string, int carriage\_pos, int num\_of\_states,

char \*\*table) {

int state = 1;

for (int count = 0;; ++count) {

// вывод строки

printf("%2d %s\n", count, string);

for (int i = 0; i < carriage\_pos; ++i) printf(" ");

printf(" %d\n", state);

if (!state) break;

// считывание символа и нужной клетки в таблице

char symbol = string[carriage\_pos];

int symbol\_index = 0;

while (symbol != alphabet[symbol\_index]) ++symbol\_index;

char \*instruction = table[symbol\_index \* num\_of\_states + state - 1];

// выполнение содержимого клетки

string[carriage\_pos] = instruction[0];

if (instruction[1] == '>')

++carriage\_pos;

else if (instruction[1] == '<')

--carriage\_pos;

int new\_state = 0; // у Тихона больше 10 состояний :)

for (int i = 1;; ++i) {

char c = instruction[(int)strlen(instruction) - i];

if (c < '0' || '9' < c) break;

new\_state += (c - 48) \* (int)pow(10, i - 1);

}

state = new\_state;

}

}

void emul\_1() {

char s[] = " 11 111 ";

char \*t[] = {"1.2", "2>3", "1>3", "1>4", "1>5", "1<6", "XXX",

"XXX", "XXX", "XXX", "XXX", "XXX", "2>2", "1<7",

" >1", " <7", " >4", " >5", "1<6", " <6", " .0"};

emul("12 ", s, 0, 7, t);

}

void emul\_2() {

char s[] = " 5-9 ";

char \*t[] = {"9<1", "0<2", "9<3", "0>4", "XXX", " >6",

"0<2", "1<2", "9<4", "1>4", "XXX", "XXX",

"1<2", "2<2", "1>4", "2>4", "XXX", "XXX",

"2<2", "3<2", "2>4", "3>4", "XXX", "XXX",

"3<2", "4<2", "3>4", "4>4", "XXX", "XXX",

"4<2", "5<2", "4>4", "5>4", "XXX", "XXX",

"5<2", "6<2", "5>4", "6>4", "XXX", "XXX",

"6<2", "7<2", "6>4", "7>4", "XXX", "XXX",

"7<2", "8<2", "7>4", "8>4", "XXX", "XXX",

"8<2", "9<2", "8>4", "9>4", " <6", "XXX",

"->5", "-<3", "XXX", "->1", " .5", " .5",

" <1", " >5", " >5", " >5", " .0", " >6"};

// каретка всегда на последнем символе строки

emul("0123456789- ", s, (int)strlen(s) - 1, 6, t);

}

void emul\_3() {

char s[] = " ^111^ ";

char \*t[] = {"^.0", "^<5", "^<4", "^>6", "^>0", "1.0", "^<7", "^<8", "^>10", "^>11", "^<12", "^<12", "^<13", "^>14",

"\*.1", "1<2", "1>3", "1>2", "^<5", "XXX", "XXX", "1<2", "1>9", "XXX", "\*.7", "1<12", "1<13", "1>14",

"\*<7", "\*>3", "\*>2", "^<4", "^<5", "XXX", "\*<7", "XXX", "XXX", "XXX", "\*>11", "\*<12", "XXX", "\*.2",

"XXX", " .9", "XXX", "XXX", "XXX", "XXX", " <8", "^<9", "1>9", " >10", "XXX", " <13", "^>14", " >14"};

int car\_pos = 0;

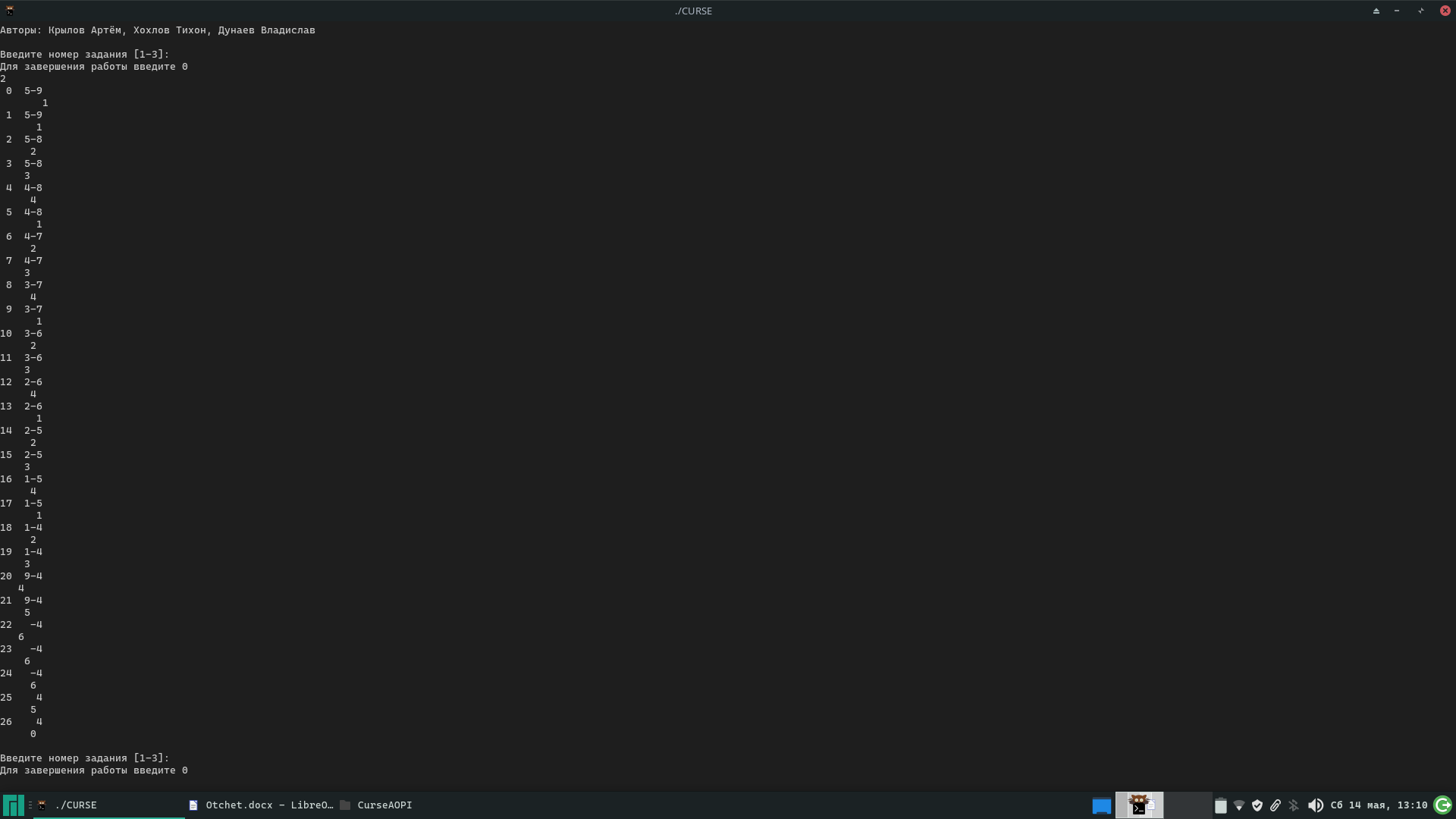
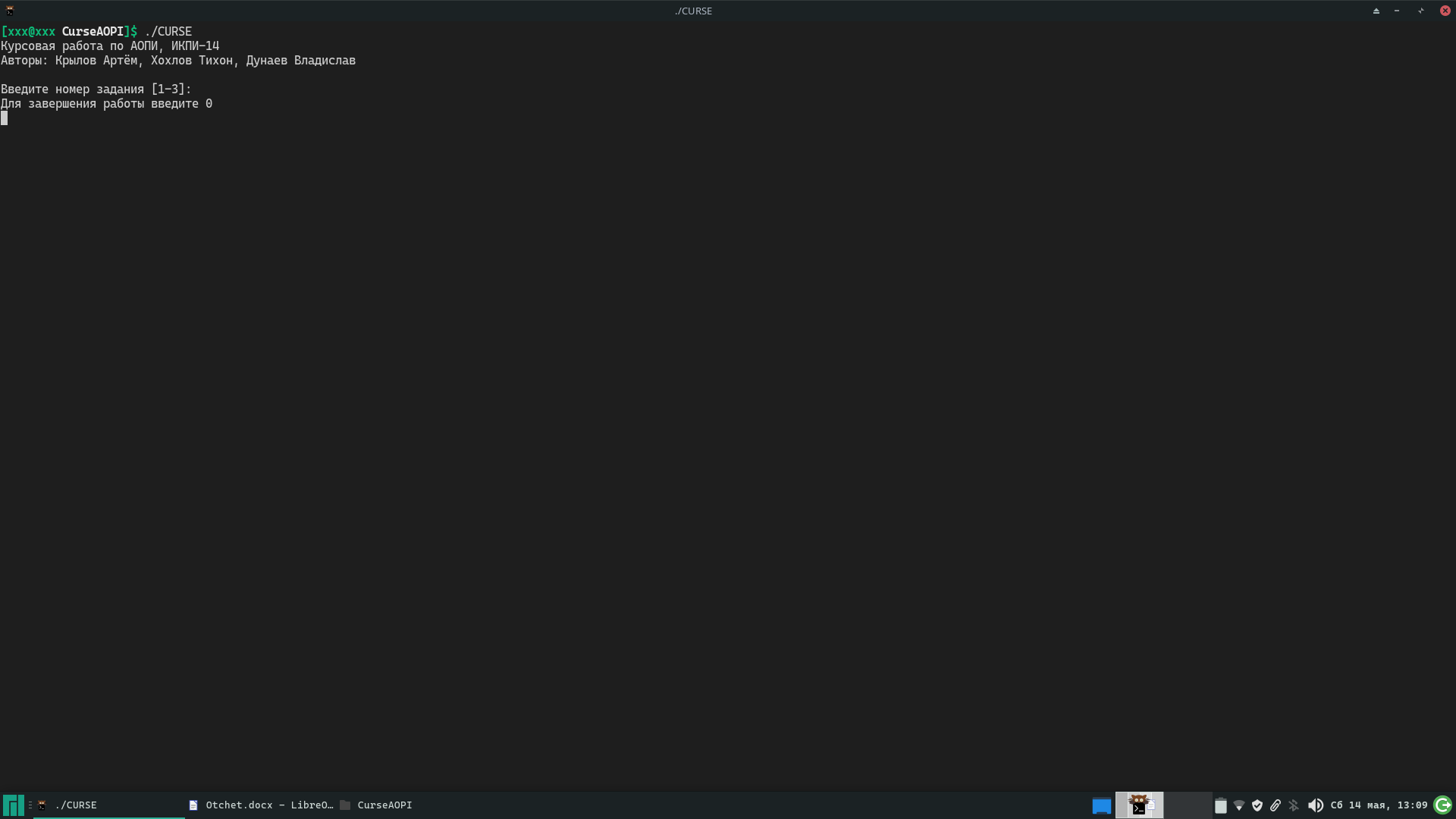
// каретка всегда на первой единице

while (s[car\_pos] != '1') ++car\_pos;

emul("^1\* ", s, car\_pos, 14, t);

}

1. Cкриншоты с контрольными примерами



1. Выводы

В ходе лабораторной работы мы составляли документацию, поняли принцип работы Машины Тьюринга, научились читать и составлять последоваетльность команд для МТ, улучшили навыки в создании блок-схем.

Перечень принятых сокращений

|  |  |
| --- | --- |
| МТ | Машина Тьюринга |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |