Казанский (приволжский) федеральный университет

Институт вычислительной математики и информационных технологий

Кафедра информационных систем и технологий

Отчет семестровых работ по дисциплине:

“Информатика”

Выполнил студент группы 09-913

Фаизова А.Н.

Проверил преподаватель

Ахтямов Р.Б.

Казань

2020г

Содержание:

[**Бинарные файлы**](#_tpk65h4xx3ow) **3**

[Постановка задачи](#_27k41t7jafs3) 3

[Задание 1. Выбор книги](#_7vkz1kq2bekk) 4

[Задание 2. Помощь начинающим инженерам-материаловедам](#_fng7ozmpa95) 10

[**Метод состояний**](#_r6girhqn3u8o) **18**

[Постановка задачи](#_rfnx25h548a5) 18

[Задание 1](#_w93w8kj85kxn) 19

[Задание 2](#_ha98hdr42ak4) 23

[Задание 3](#_vm8mcw3794yb) 27

[**Моделирование машины Тьюринга**](#_72xauw5j35qa) **31**

[Постановка задачи](#_g71ar3hrfk7f) 31

[Задание 1](#_48lgp1409w2w) 32

[Задание 2](#_r709b9hph5w0) 36

[Задание 3](#_q2f2h6x7pste) 42

[**Вывод**](#_ojefjd8fmi0f) **48**

[**Список использованных ресурсов**](#_lpb53kwin34d) **49**

# Бинарные файлы

## Постановка задачи

Задачи в этом разделе подразумевают работу с бинарными файлами и создание структур, соответствующих данным.

### Задание 1. Выбор книги

Необходимо отобрать n книг для подарка букинисту-романтику. Каждая книга из книжного магазина имеет следующие атрибуты:

* название книги;
* формат(0-бумажная, 1-электронная, 2-аудио);
* стоимость;
* массив жанров, куда можно отнести это произведение(0-детская, 1-поэзия, 2-роман/повесть, 3-драма, 4-биографический, 5-детектив, 6-мистика, 7-фантастика, 8-научная фантастика, 9-ужасы, 10-романтический, 11-исторический, 12-специальная/профессиональная литература);
* автор книги;
* количество страниц;
* издательство;
* год публикации;
* язык, на котором написана книга;
* возрастные ограничения;
* рейтинг.

Текстовый файл с данными содержит строки вида:

CameraObeskura 0 142 0 0 1 1 0 0 0 0 0 0 1 0 0 Nabokov 224 Azbuka 2013 rus 16 4

1. название книги CameraObeskura
2. формат 0(бумажная)
3. стоимость 142
4. жанр 0(детская-нет)

0(поэзия-нет)

1(роман/повесть-да)

1(драма-да)

0(биографический-нет)

0(детектив-нет)

0(мистика-нет)

0(фантастика-нет)

0(научная фантастика-нет)

0(ужасы-нет)

1(романтический-да)

0(исторический-нет)

0(специальная/профессиональная литература-нет)

1. автор книги Nabokov
2. количество страниц 224
3. издательство Azbuka
4. год публикации 2013
5. язык, на котором rus

написана книга

1. возрастные ограничения 16
2. рейтинг 4

Создать массив записей. В массив включить перечисленные поля 1) - 11). В массив включаются книги, удовлетворяющие следующему критерию:

* бумажная;
* жанр {научная фантастика} или {поэзия} или {роман/повесть} не {исторический};
* автор {Бродский} или {Набоков} или {Шопенгауэр} или {Киз };
* годов 1880-1990;
* на русском языке.

Массив должен быть упорядочен по невозрастанию рейтинга. Задаётся число n - количество книг на выдаваемой странице сайта магазина. Переносим данные массива в бинарный файл ровно n экземпляров.

Поиск наиболее подходящей книги среди отобранных в бинарном файле:

* не более 500 страниц;
* рейтинга 12+ и выше

Упорядочить по увеличению стоимости(от дешевой к дорогой) и поместить окончательные результаты поиска лучшей книги(название, стоимость, автор, объем, год публикации, рейтинг) во второй бинарный файл, если таковые имеются или разместить сообщение - "There is no appropriate book!".

Код программы:

#include <iostream>

#include <fstream>

#include <cstring>

using namespace std;

struct Book

{

char book\_name[50];

int format;//0-paper, 1-e-book, 2-audiobook

int cost;

int genre[13];//GROUP!!!0-children's,1-poetry,2-novel,3-drama,4-biography,5-detective,6-mystety,7-fantasy,8-science fiction,9-horror,10-romance,11-historical,12-spec/prof literature

char author[30];

int length;

char publisher[30];

int pub\_year;

char language[30];

int age\_restr;

int rating;

};

void Sort\_rating(Book\* A, int k)//сорт по убыванию

{

int i, j;

Book t;

for (i = 0; i < k - 1; i++)

for (j = i + 1; j < k; j++)

if (A[i].rating < A[j].rating)

{

t = A[i];

A[i] = A[j];

A[j] = t;

}

}

void Sort\_cost(Book\* A, int k)//сорт по возрастанию

{

int i, j;

Book t;

for (i = 0; i < k - 1; i++)

for (j = i + 1; j < k; j++)

if (A[i].cost > A[j].cost)

{

t = A[i];

A[i] = A[j];

A[j] = t;

}

}

int main()

{

ifstream in("books.txt");

ofstream out\_bin("appropriate\_books.bin", ios::binary);

Book x;

int k = 0, n = 0;

int i;

Book mas[100];

if (!in)

{

cout << "File 'books.txt' not found" << endl;

system("pause");

exit(-1);

}

while (in >> x.book\_name >> x.format >> x.cost >> x.genre[0] >> x.genre[1] >> x.genre[2] >> x.genre[3] >> x.genre[4] >> x.genre[5] >> x.genre[6] >> x.genre[7] >> x.genre[8] >> x.genre[9] >> x.genre[10] >> x.genre[11] >> x.genre[12] >> x.author >> x.length >> x.publisher >> x.pub\_year >> x.language >> x.age\_restr >> x.rating)

{

// выбираем бумажную книгу жанра (научная фантастика или поэзия или роман) не (исторический) от авторов (Бродский или Набоков или Шопенгауэр или Киз) годов 1880-1980 язык русский

if ((x.format == 0) && ((x.genre[1] == 1) || (x.genre[2] == 1) || (x.genre[7] == 1)) && (x.genre[11] == 0) &&

((!strcmp(x.author, "Brodsky")) || (!strcmp(x.author, "Nabokov")) || (!strcmp(x.author, "Keyes"))) &&

((x.pub\_year >= 1880) && (x.pub\_year <= 1990)) && (!strcmp(x.language, "rus")))

{

mas[k] = x;

k++;

}

}

in.close();

Sort\_rating(mas, k); // сортируем по убыванию рейтинга

cout << "Enter number of books on one page: "; cin >> n;

//out << "List of the n bestsellers" << endl;

for (i = 0; i < n; i++)

{

out\_bin.write((char\*)& mas[i], sizeof(struct Book));

}

out\_bin.close();

ifstream good\_rating("appropriate\_books.bin", ios::binary);

ofstream good\_rating\_cost("best\_app\_books.txt");

if (!good\_rating)

{

cout << "File 'appropriate\_books.bin' not found" << endl;

system("pause");

exit(-1);

}

Book\* book = new Book[n];

k = 0;

for (i = 0; i < n; i++)

{

good\_rating.read((char\*)& x, sizeof(struct Book));

if ((x.length < 500) && (x.age\_restr >= 12))// отбираем не слишком длинные и не слишком детские книги

{

book[k] = x; k++;

}

}

if (k == 0) { good\_rating\_cost << "There is no appropriate book!"; }

else

{

Sort\_cost(book, k);// отсортировали книги с хорошим рейтингом по возрастанию(от дешёвого к дорогому)

for (i = 0; i < k; i++)

{

good\_rating\_cost << book[i].book\_name << " " << book[i].cost << " " << book[i].author << " " << book[i].length << " " << book[i].pub\_year << " " << book[i].rating << endl;

}

}

good\_rating.close();

good\_rating\_cost.close();

delete[] book;

}

Файл book.txt

CameraObeskura 0 142 0 0 1 1 0 0 0 0 0 0 1 0 0 Nabokov 224 Azbuka 2013 rus 16 4

IzobretatelValsa 0 358 0 0 1 1 0 0 0 0 0 0 1 0 0 Nabokov 480 Azbuka 1978 rus 16 5

Volshebnik 0 140 0 0 1 1 0 0 0 0 0 0 1 0 0 Nabokov 144 Azbuka 1986 rus 18 4

Lolita 0 151 0 0 1 1 0 0 0 0 0 0 1 0 0 Nabokov 448 Azbuka 2020 rus 18 5

CvetiDlyaEldgernona 0 216 0 0 1 1 0 0 0 1 1 0 0 0 0 Keyes 320 Eksmo 2014 rus 16 5

CameraObeskura 1 199 0 0 1 1 0 0 0 0 0 0 1 0 0 Nabokov 224 Azbuka 2015 rus 16 5

TainstvennayaIstoriaBM 0 506 0 0 1 1 0 0 0 1 1 0 0 0 0 Keyes 576 Eksmo 2000 rus 18 5

Prikosnovenie 0 358 0 0 1 1 0 0 0 1 1 0 0 0 0 Keyes 352 Eksmo 2000 rus 16 5

ChastRechi 0 272 0 1 0 0 1 0 0 0 0 0 0 0 0 Brodsky 144 LenIzdat 1987 rus 12 5

BlednyOgon 0 140 0 0 1 1 0 0 0 0 0 0 1 0 0 Nabokov 320 Azbuka 1962 rus 16 4

Lolita 1 169 0 0 1 1 0 0 0 0 0 0 1 0 0 Nabokov 448 Azbuka 2012 rus 18 5

TainstvennayaIstoriaBM 0 222 0 0 1 1 0 0 0 1 1 0 0 0 0 Keyes 576 Eksmo 1989 rus 18 5

Nazidanie 0 320 0 1 0 0 0 0 0 0 0 0 0 0 0 Brodsky 260 Smart 1990 rus 12 4

Prikosnovenie 0 358 0 0 1 1 0 0 0 1 1 0 0 0 0 Keyes 352 Eksmo 1977 rus 16 5

NabereznayaNeicelimih 0 272 0 1 0 0 1 0 0 0 0 0 0 0 0 Brodsky 192 LenIzdat 1987 rus 12 5

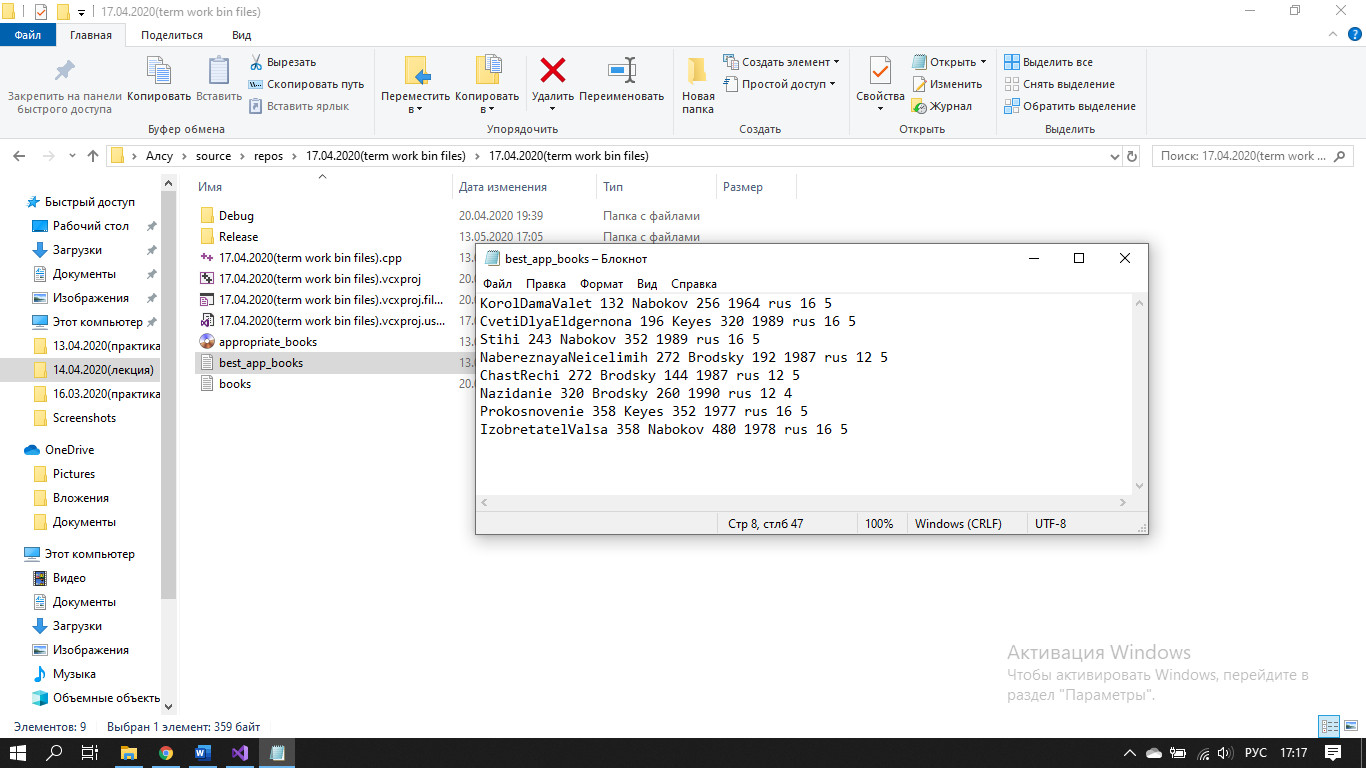
Lolita 0 370 0 0 1 1 0 0 0 0 0 0 1 0 0 Nabokov 448 Azbuka 2020 rus 18 5

Stihi 0 243 0 1 0 1 0 0 0 0 0 0 1 0 0 Nabokov 352 Azbuka 1989 rus 16 5

KorolDamaValet 0 132 0 0 1 1 0 0 0 0 0 0 1 0 0 Nabokov 256 Azbuka 1964 rus 16 5

CvetiDlyaEldgernona 0 196 0 0 1 1 0 0 0 1 1 0 0 0 0 Keyes 320 Eksmo 1989 rus 16 5

Результат работы программы:



### Задание 2. Помощь начинающим инженерам-материаловедам

Задача состоит в том, чтобы отобрать химические элементы со свойствами, требующимися инженерам в их исследовательской работе.

Предполагается, что каждый х.э. обладает следующим набором атрибутов:

* атомное число(порядковый номер);
* символьное обозначение;
* полное латинское название;
* номер группы в ПСХЭ(Периодической системе химических элементов)
* номер периода в ПСХЭ;
* атомную массу(в а.е.м.);
* плотность(г/см^3 или г/л);
* температуру кипения(в Кельвинах);
* удельную теплопроводность(Дж/(г\*К));
* электроотрицательность;
* год открытия(где 0 - доисторический период);
* распространенность в Земной коре(мг/кг).

Текстовый файл с данными содержит строки вида:

17 Cl Chlorine 17 3 35.45 0.003 239.11 0.479 3.16 1774 145

1. порядковый номер 17
2. символьное обозначение Cl
3. полное латинское название Chlorine
4. номер группы 17
5. номер периода 3
6. атомная масса 35.45
7. плотность 0.003
8. температура кипения 239.11
9. удельная теплопр-ть 0.479
10. электроотрицательность 3.16
11. год открытия 1774
12. распространенность 145

Создать два массива записей. В запись включить перечисленные поля 1) - 12).

В первый массив включаются х.э., соответствующие заданным командой инженеров критериям:

* тяжелее 40а.е.м;
* температура кипения менее 1500К;
* электроотрицательность больше 1.5.

Массив должен быть упорядочен по уменьшению распространенности.

Во второй массив записываются х.э., удовлетворяющие следующим условиям отбора:

* щелочные и щелочноземельные(то есть элементы первой и второй групп).

Массив следует отсортировать по возрастанию автономного числа.

В конце показать, что получилось в файлах, выводя значения на консоль.

Замечание: в программе для удобства демонстрации результатов перегружен оператор вывода в поток.

Код программы:

#include <iostream>

#include <fstream>

#include <cstring>

#include <iomanip>

using namespace std;

struct Chem\_element

{

int atom\_num;

char symbol[4];

char latin\_name[20];

int group;

int period;

float atom\_weight; // (Da)

float density; // (g/cm^3) or (g/l)

float boil\_point; // (K)

float specif\_heat\_capacity; // (J/(g\*K))

float elect\_negat;

int open\_year;

float abudance; // (mg/kg)

};

void Sort\_abudance(Chem\_element\* A, int k)//сортируем по убыванию распространенности

{

int i, j;

Chem\_element t;

for (i = 0; i < k - 1; i++)

for (j = i + 1; j < k; j++)

if (A[i].abudance < A[j].abudance)

{

t = A[i];

A[i] = A[j];

A[j] = t;

}

}

void Sort\_atom\_num(Chem\_element\* A, int k)//сортируем по возрастанию атомного числа

{

int i, j;

Chem\_element t;

for (i = 0; i < k - 1; i++)

for (j = i + 1; j < k; j++)

if (A[i].atom\_num > A[j].atom\_num)

{

t = A[i];

A[i] = A[j];

A[j] = t;

}

}

void Print\_head()

{

cout << "------------------------------------------------------------------------------------------------------" << endl;

cout << setw(4) << "Num" << "|" << setw(5) << "Symb" << "|" << setw(15) << "Latin" << "|" << setw(8) << "Group" << "|" << setw(10)

<< "Period" << "|" << setw(10) << "Weight" << "|" << setw(9) << "Density" << "|" << setw(7) << "Boil" << "|" << setw(7)

<< "Heat" << "|" << setw(7) << "Elect" << "|" << setw(10) << "Spread" << endl;

cout << "------------------------------------------------------------------------------------------------------" << endl;

}

ostream& operator<< (ostream& out, const Chem\_element& element)

{

out << setw(4) << element.atom\_num << "|"<< setw(5) << element.symbol << "|" << setw(15) << element.latin\_name << "|" << setw(8) << element.group << "|" << setw(10) << element.period << "|" << setw(10) <<

element.atom\_weight << "|" << setw(9) << element.density << "|" << setw(7) << element.boil\_point << "|" << setw(7) << element.specif\_heat\_capacity << "|" << setw(7) << element.elect\_negat << "|" << setw(10) << element.abudance << endl;

return out;

}

// разделю данные в два бинарника: в одном тяжелые с темп кипения меньше.., c электроотрицательностью больше.. , отсортированные на уменьшению распространенности, в другом - щелочные и щелочноземельные, то есть первой и второй групп, остортированные по атомному числу по возрастанию

int main()

{

ifstream in("elements.txt");// исходный файл с данными

ofstream out1\_bin("rife\_heavy\_metals.bin", ios::binary);// бинарный файл, куда положим распространенные элементы, соответствующие критериям по атомной массе, температуре кипения и электроотрицательности

ofstream out2\_bin("alkali\_and\_alkaline\_earth\_metals.bin", ios::binary);// бинарный файл, где будут щелочные и щелочно-земельные металлы, отсортированные по атомному числу

Chem\_element x;

int k1 = 0, k2 = 0;

int i;

Chem\_element mas1[100];

Chem\_element mas2[100];

if (!in)

{

cout << "File 'elements.txt' not found" << endl;

system("pause");

exit(-1);

}

while (in >> x.atom\_num >> x.symbol >> x.latin\_name >> x.group >> x.period >> x.atom\_weight >> x.density >> x.boil\_point >> x.specif\_heat\_capacity >> x.elect\_negat >> x.open\_year >> x.abudance)

{

// отбираем химические элементы тяжелее 40а.е.м., которые кипят при темп-ре меньше 1500К и электроотрицательности больше 1.5 по практической шкале

if ((x.atom\_weight > 40) && (x.boil\_point < 1500) & (x.elect\_negat > 1.5))

{

mas1[k1] = x;

k1++;

}

// отбираем щелочные и щелочно-земельные металлы, то есть те химические элементы номер группы которых либо 1, либо 2

if (((x.group == 1) || (x.group == 2)) && (x.period > 1))

{

mas2[k2] = x;

k2++;

}

}

in.close();

Sort\_abudance(mas1, k1); //сортируем по убыванию распространенности первый массив

Sort\_atom\_num(mas2, k2);//сортируем по возрастанию атомного числа второй массив

for (i = 0; i < k1; i++)

{

out1\_bin.write((char\*)& mas1[i], sizeof(struct Chem\_element));

}

for (i = 0; i < k2; i++)

{

out2\_bin.write((char\*)& mas2[i], sizeof(struct Chem\_element));

}

//out.close();

out1\_bin.close();

out2\_bin.close();

//посмотрим, что записалось

ifstream input1("rife\_heavy\_metals.bin", ios::binary);

ifstream input2("alkali\_and\_alkaline\_earth\_metals.bin", ios::binary);

//Chem\_element\* mas1\_elem = new Chem\_element[k1];

if (!input1)

{

cout << "File 'rife\_heavy\_metals.bin' not found" << endl;

system("pause");

exit(-1);

}

else

{

cout << "\n\t\tRife heavy metals\n";

Print\_head();

for (i = 0; i < k1; i++)

{

input1.read((char\*)& x, sizeof(struct Chem\_element));

cout << x;

}

}

if (!input2)

{

cout << "File 'alkali\_and\_alkaline\_earth\_metals.bin' not found" << endl;

system("pause");

exit(-1);

}

else

{

cout << "\n\t\tAlkali and alkaline earth metals\n";

Print\_head();

for (i = 0; i < k2; i++)

{

input2.read((char\*)& x, sizeof(struct Chem\_element));

cout << x;

}

}

input1.close();

input2.close();

}

Файл elements.txt

14 Si Silicon 14 2 28.085 2.329 3538 0.705 1.9 1824 282000

2 He Helium 18 1 4.003 0.000 4.22 5.193 0 1895 0.008

6 C Carbon 14 2 12.011 2.267 4300 0.709 2.55 0 200

8 O Oxygen 16 2 15.999 0.001 90.20 0.918 3.44 1774 461000

19 K Potassium 1 4 39.098 0.862 1032 0.757 0.82 1807 20900

20 Calcium 2 4 40.078 1.54 1757 0.647 1 1808 41500

29 Cu Copper 11 4 63.546 8.96 2835 0.385 1.9 0 60

31 Ga Gallium 13 4 69.723 5.907 2673 0.371 1.81 1875 19

37 Rb Rubidium 1 5 85.468 1.532 961 0.363 0.82 1861 90

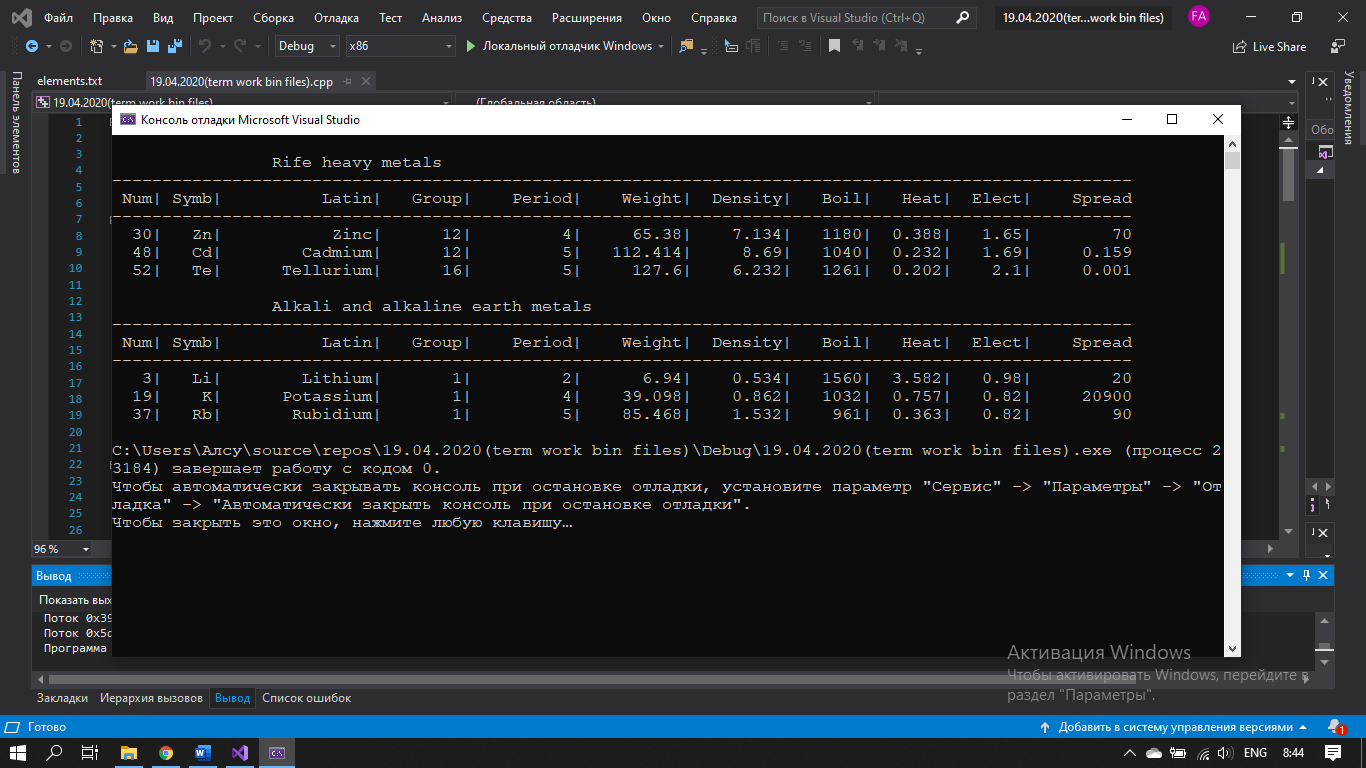
44 Ru Ruthenium 8 5 101.07 12.37 4423 0.238 2.2 1844 0.001

48 Cd Cadmium 12 5 112.414 8.69 1040 0.232 1.69 1817 0.159

49 In Indium 13 5 114.818 7.31 2345 0.233 1.78 1863 0.25

52 Te Tellurium 16 5 127.60 6.232 1261 0.202 2.1 1782 0.001

Результат работы программы:



# Метод состояний

## Постановка задачи

Проверка синтаксиса языка.Задан алфавит языка ALF= {‘A’,’B’,’C’}. Язык L – множество конечных последовательностей символов языка P=a1,…,an, a[i] in Alf, удовлетворяющих синтаксической диаграмме D.

С помощью метода состояний проверить, принадлежит ли последовательность во входном файле “word.txt” языку L. Нарисовать диаграмму метода состояний, решающую данную задачу.

### Задание 1

Диаграмма D , описывающая язык L:

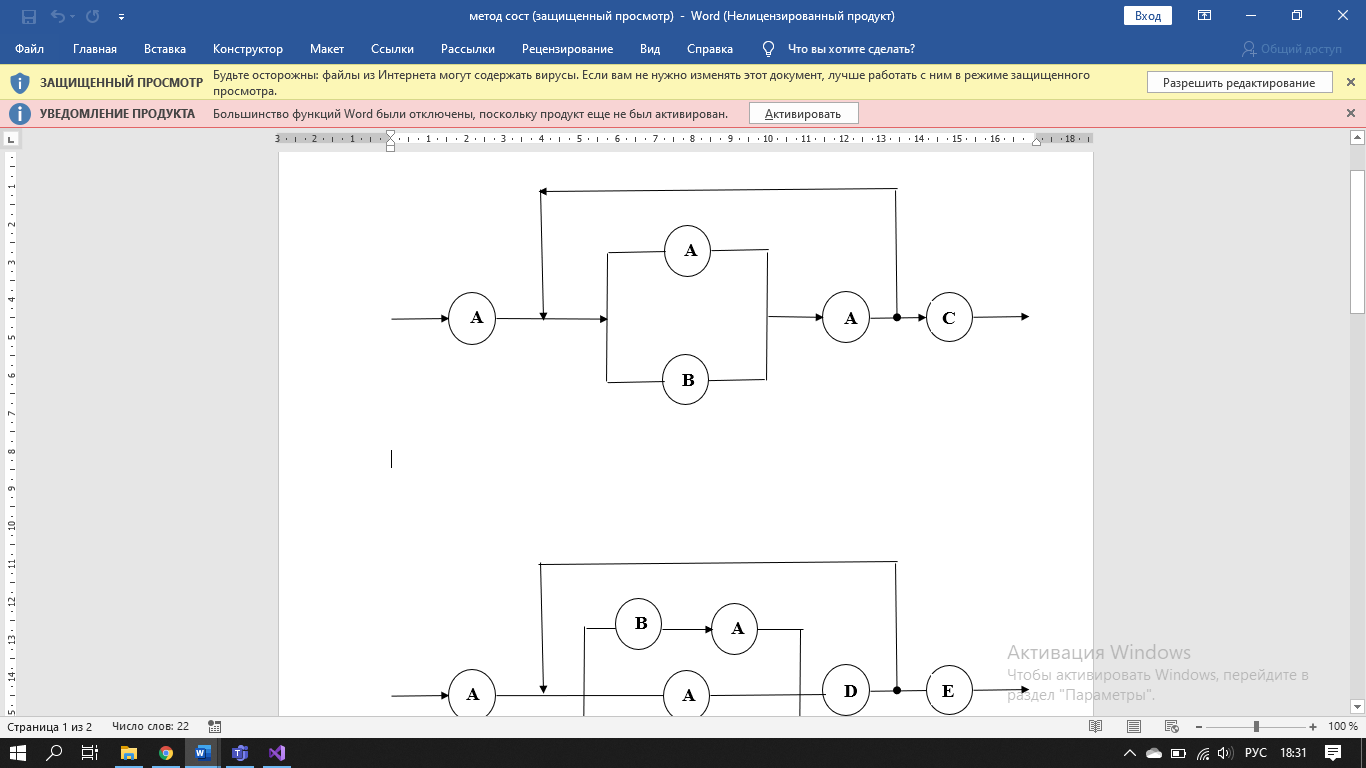
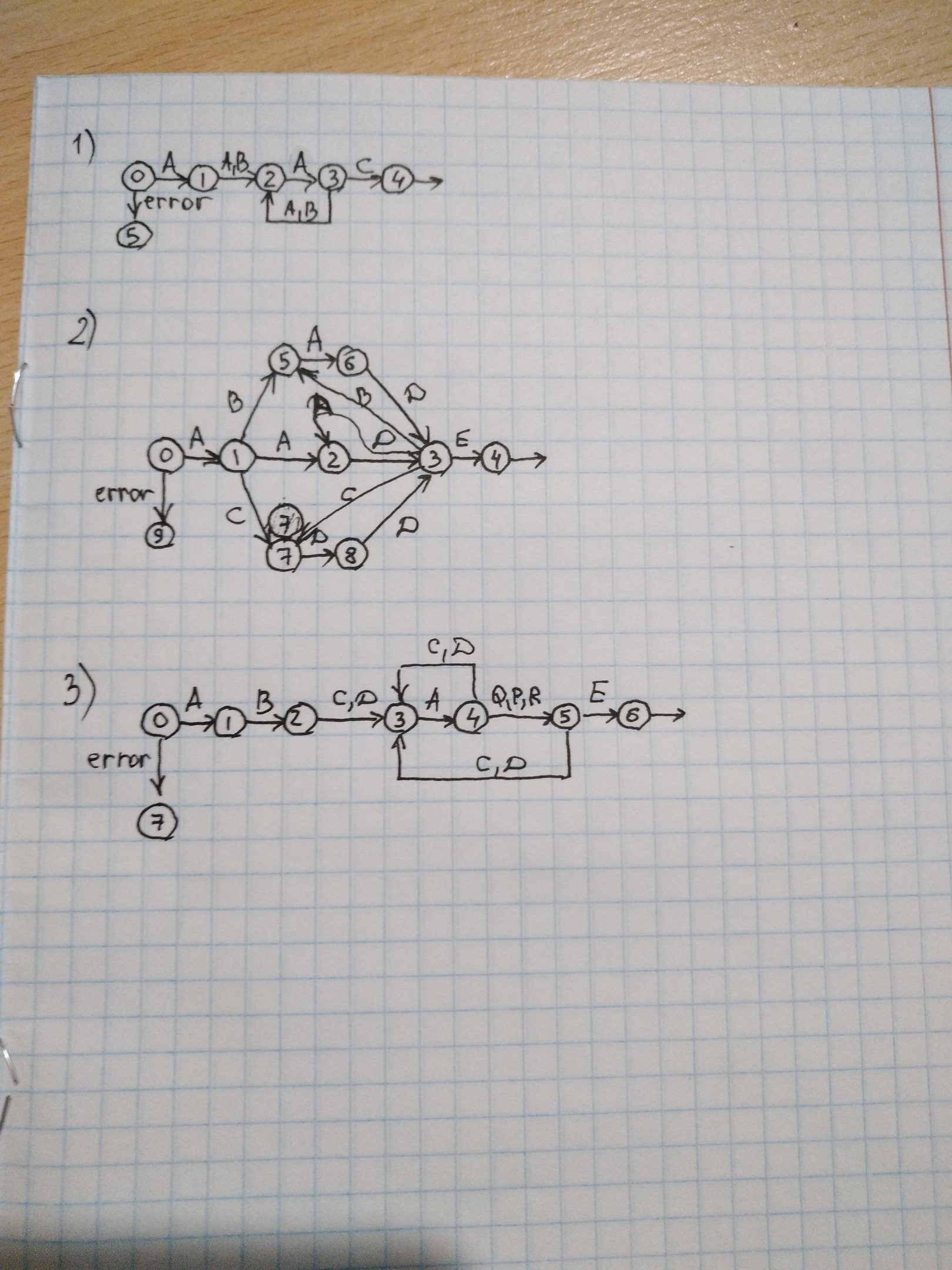


Диаграмма метода состояния(диаграмма Мура):



Код программы:

#include <iostream>

#include <fstream>

#include<iomanip>

using namespace std;

//////////////////////////////

void main ()

{

ifstream in("word.txt");

ofstream out("result.txt");

char x;

int q=0;

if (!in)

{

cout<<"Can't open file"<<endl;

exit(-1);

}

while ( in>>x && q!=5)

{

out<<x;

switch (q)

{

case 0: if (x=='A') q=1; else q=5; break;

case 1: if (x=='A' || x=='B') q=2; else q=5; break;

case 2: if (x=='A') q=3; else q=5; break;

case 3: if (x=='A' || x=='B') q=2;

else if (x=='C') q=4;

else q=5; break;

default: q=5; break;

}

out<<setw(2)<<q<<endl;

}

out<<endl;

if (q==4) out<<" sequence is element of L"<<endl;

else out<<"sequence is not element of L"<<endl;

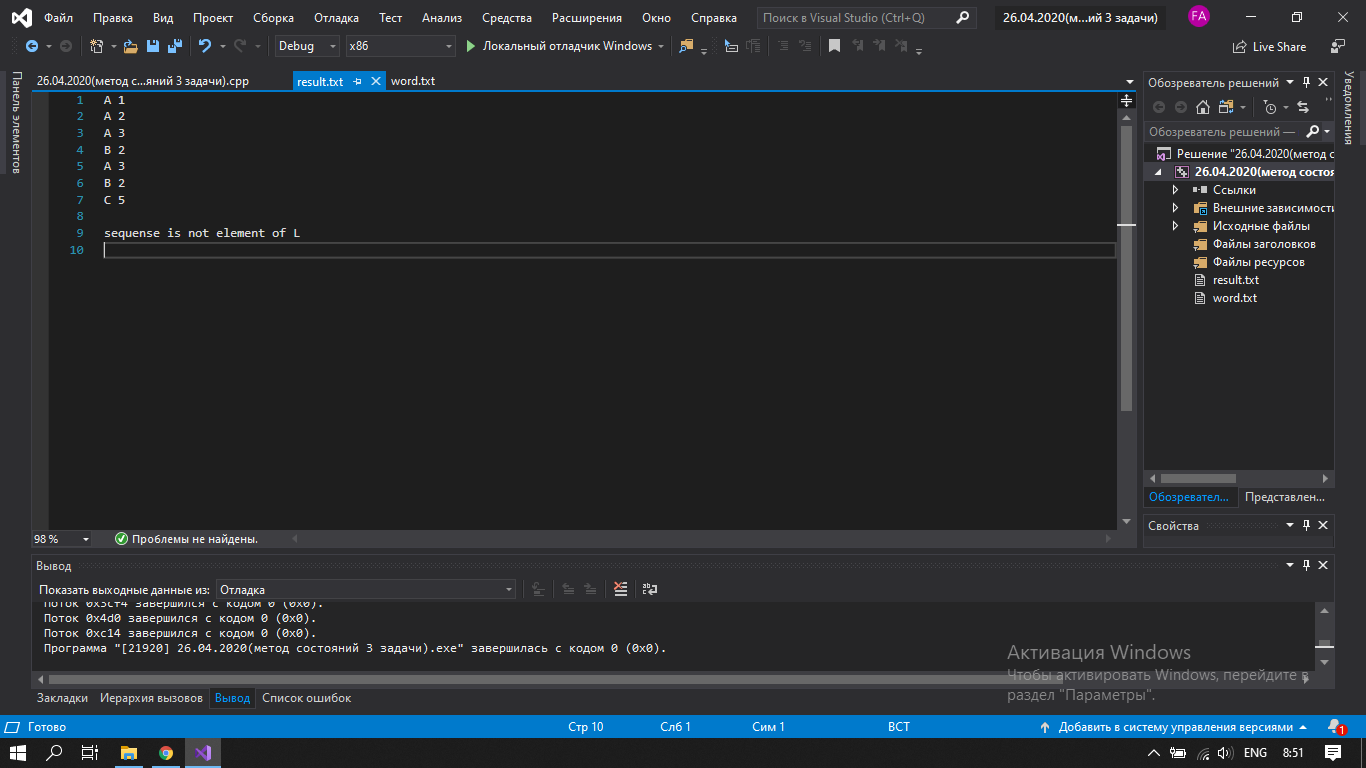
in.close();

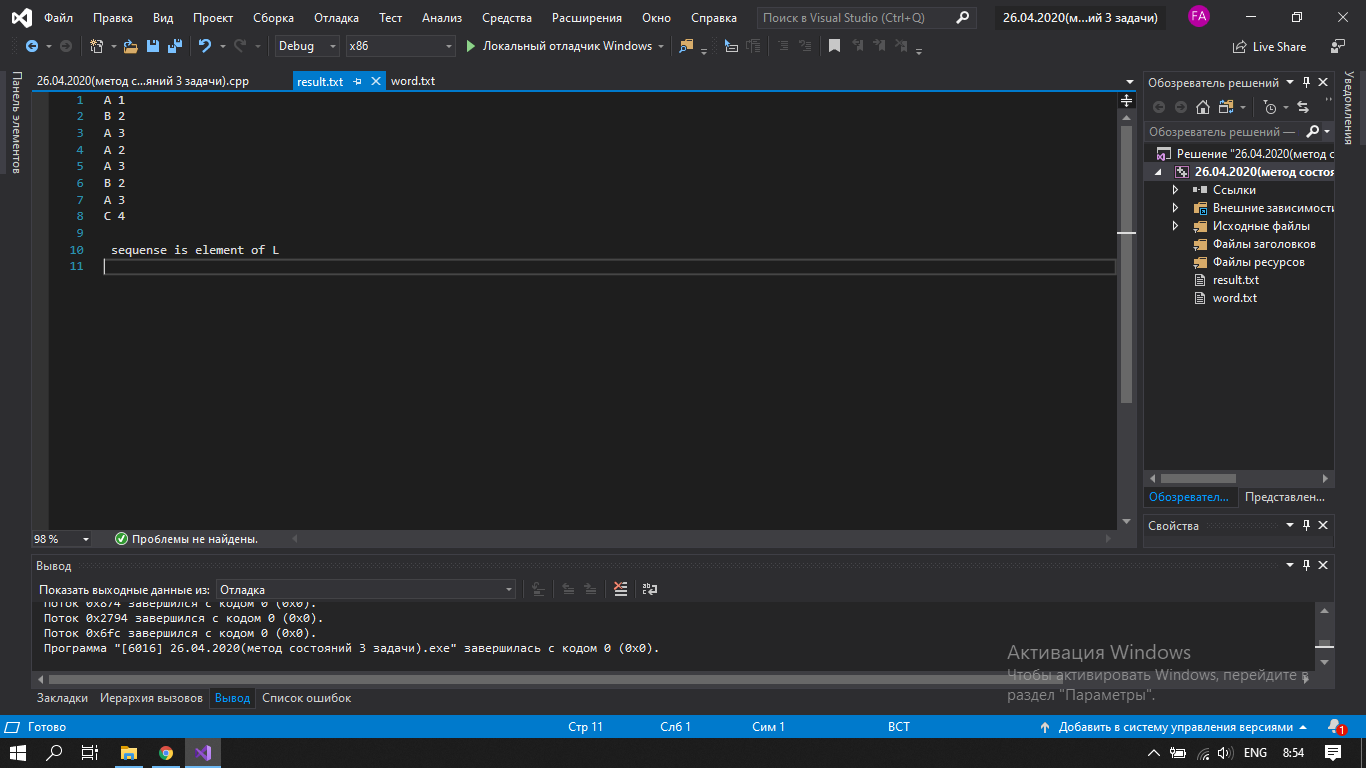
out.close();

}

Пример работы программы:

В файле “word.txt”: AAABABC



В файле “word.txt”: ABAAABAC

### Задание 2

Диаграмма D , описывающая язык L:

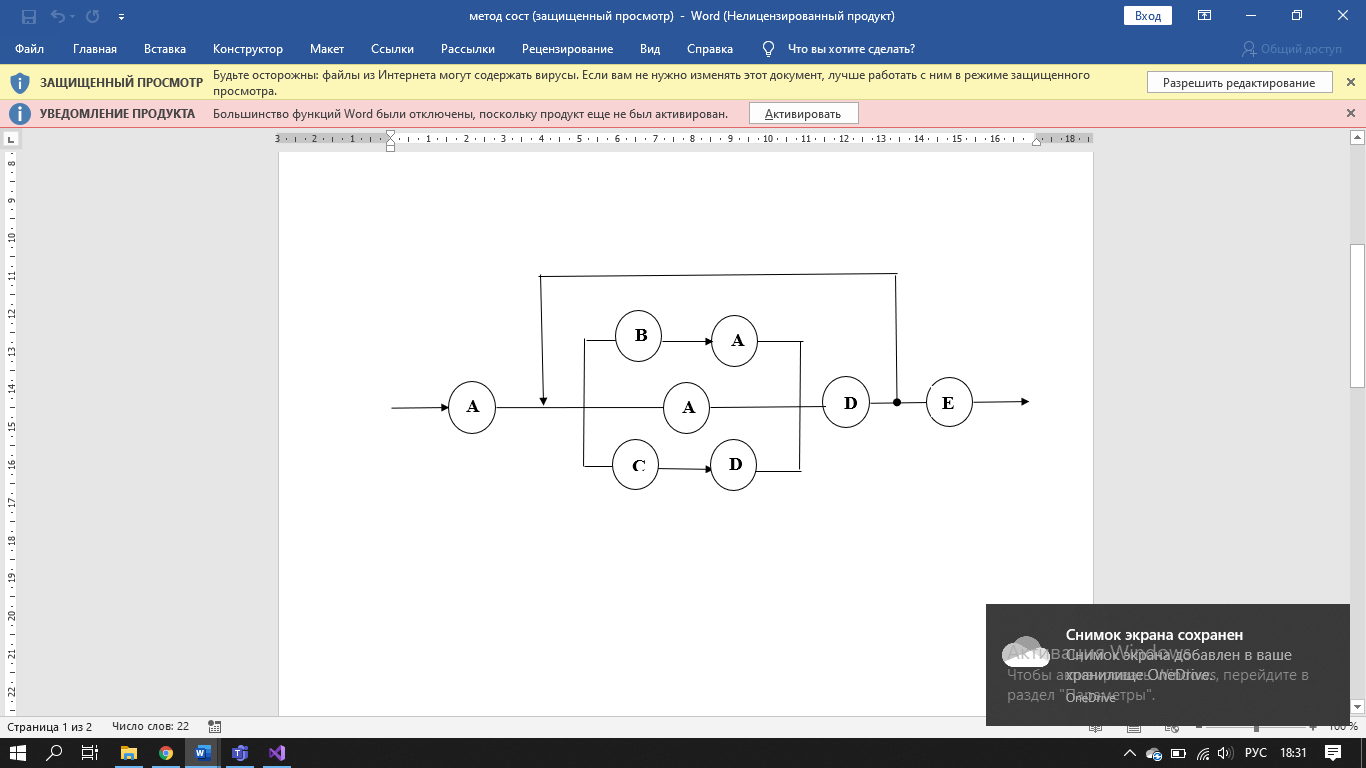


Диаграмма метода состояния(диаграмма Мура):



Код программы:

#include <iostream>

#include <fstream>

#include<iomanip>

using namespace std;

//////////////////////////////

int main()

{

ifstream in("word.txt");

ofstream out("result.txt");

char x;

int q = 0;

if (!in)

{

cout << "Can't open file" << endl;

exit(-1);

}

while (in >> x && q != 7)

{

out << x;

switch (q)

{

case 0: if (x == 'A') q = 1;

else q = 7; break;

case 1: if (x == 'A') q = 2;

else if (x == 'B') q = 5;

else if (x == 'C') q = 6;

else q = 7; break;

case 2: if (x == 'D') q = 3;

else q = 7; break;

case 3: if (x == 'E') q = 4;

else if (x == 'A') q = 2;

else if (x == 'B') q = 5;

else if (x == 'C') q = 6;

else q = 7; break;

case 5: if (x == 'A') q = 2;

else q = 7; break;

case 6: if (x == 'D') q = 2;

else q = 7; break;

default: q = 7; break;

}

out << setw(2) << q << endl;

}

out << endl;

if (q == 4) out << " sequence is element of L" << endl;

else out << "sequence is not element of L" << endl;

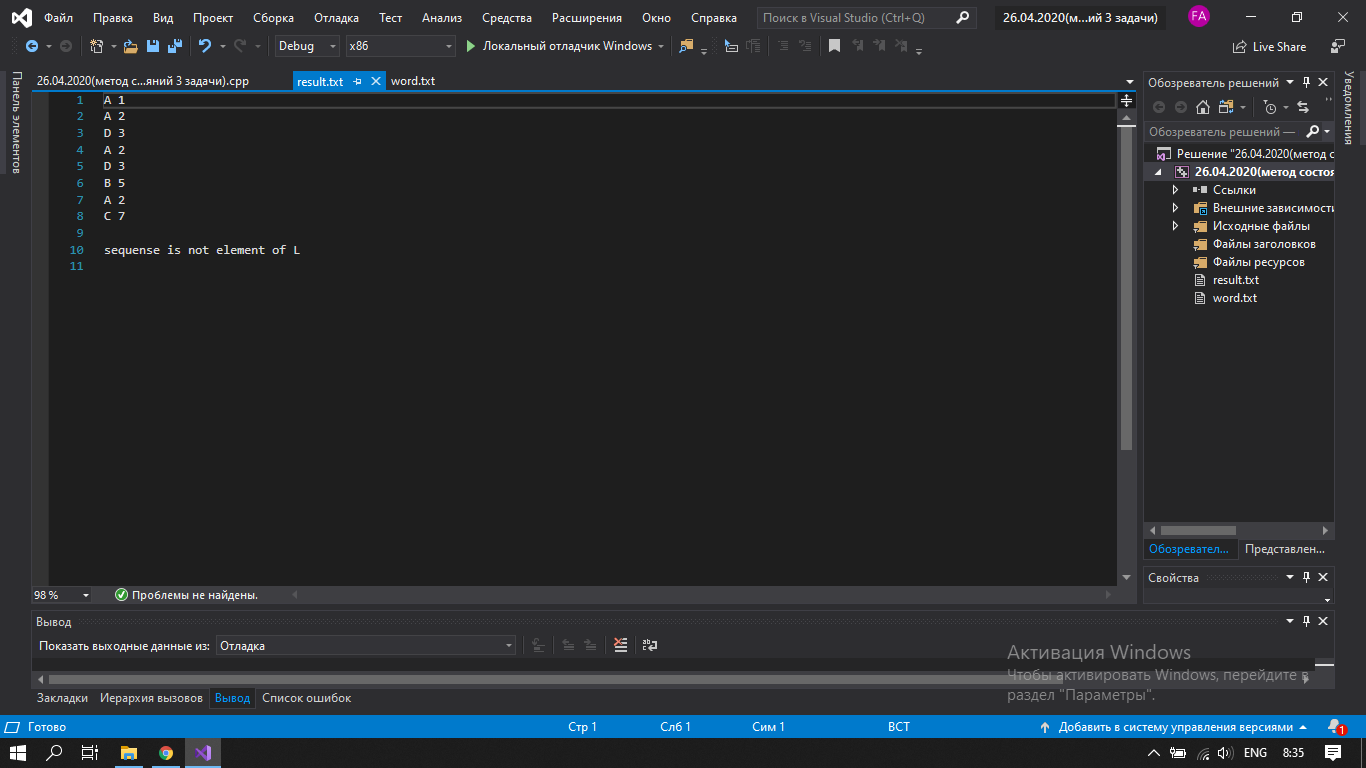
in.close();

out.close();

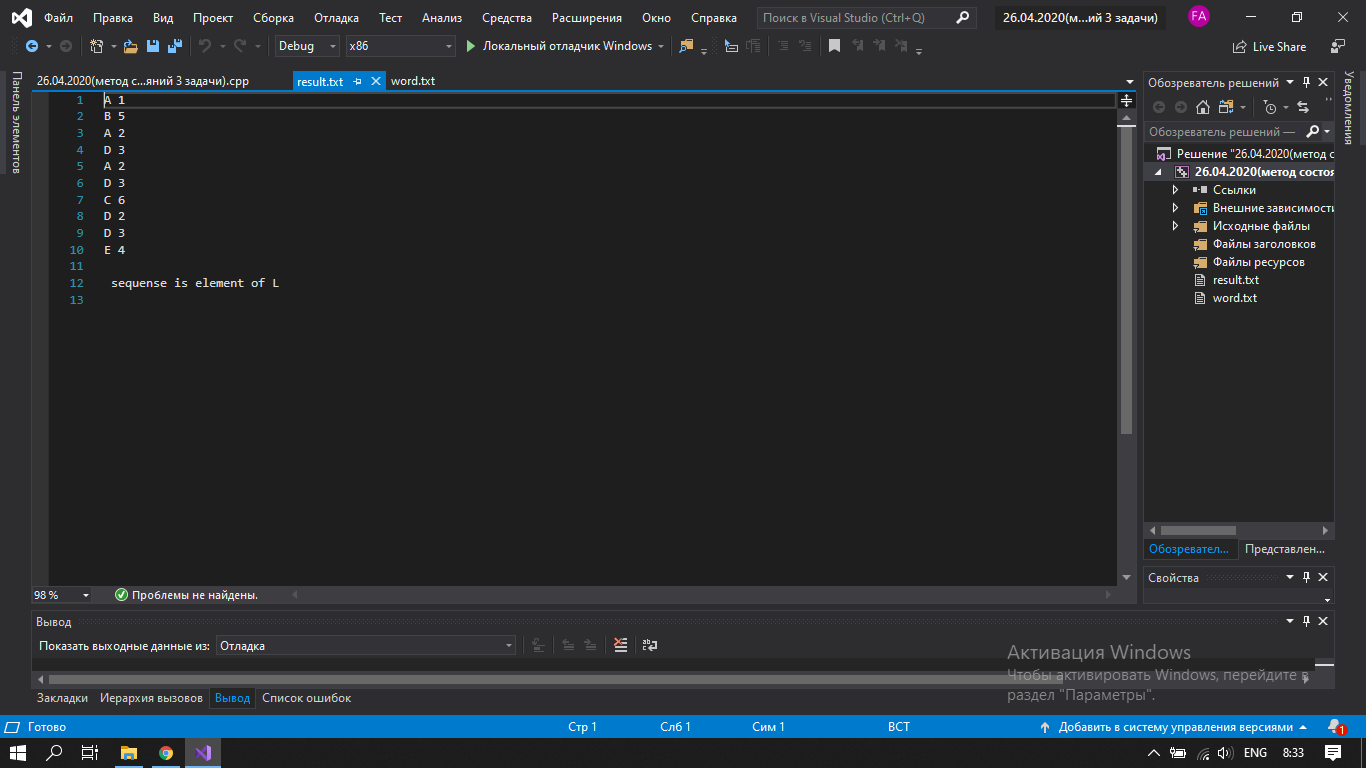
}

Пример работы программы:

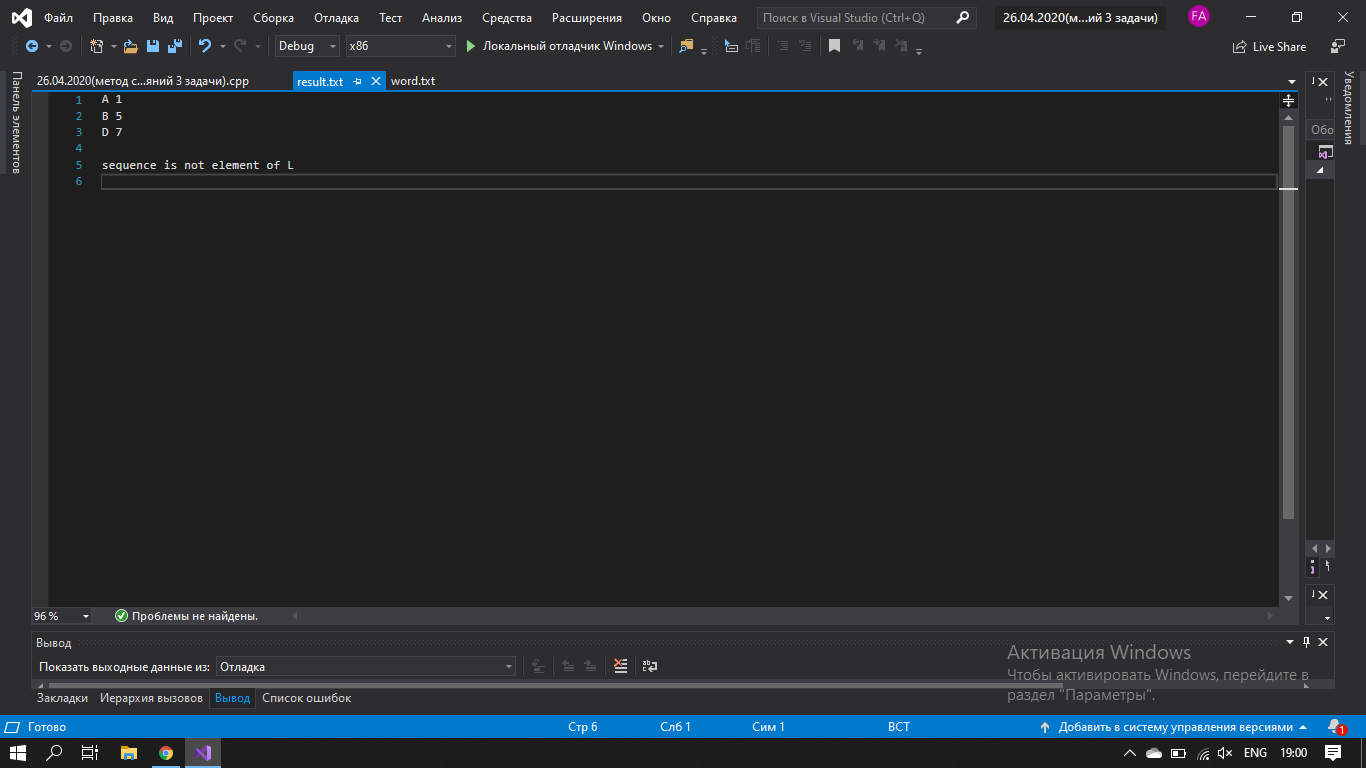
В файле “word.txt”: AADADBACE



В файле “word.txt”: ABADADCDDE



В файле “word.txt”:ABDARCDAQE



### Задание 3

Диаграмма D , описывающая язык L:

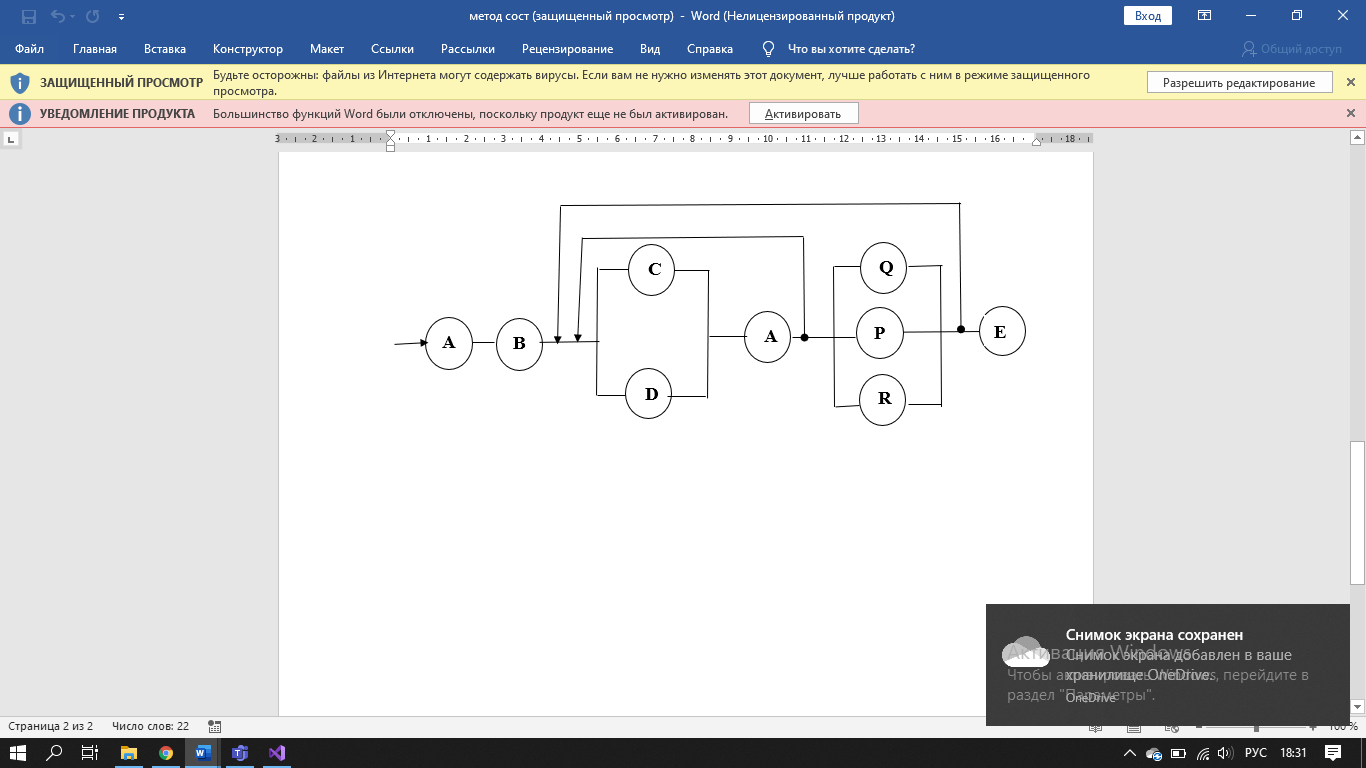
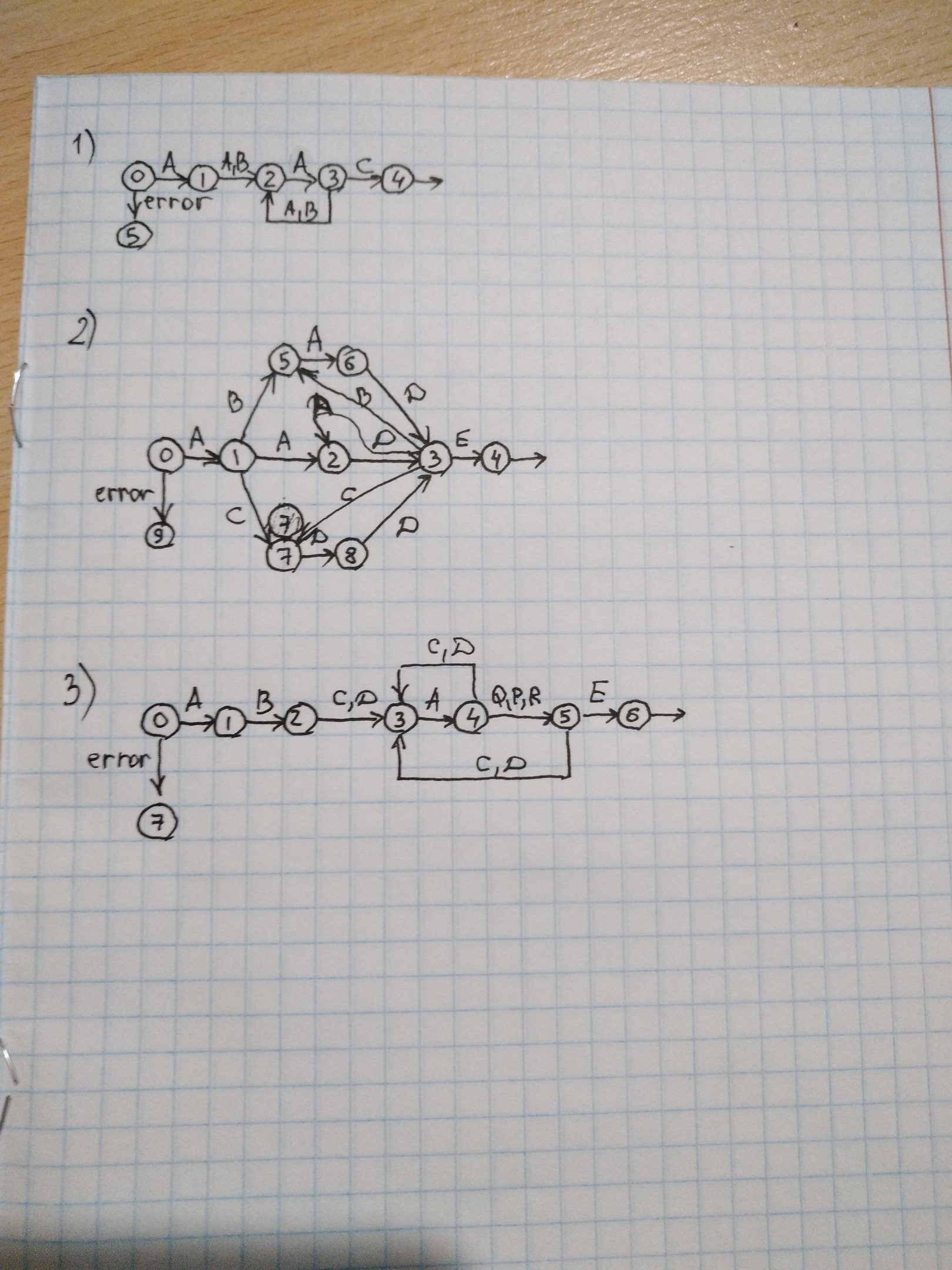


Диаграмма метода состояния(диаграмма Мура):



Код программы:

#include <iostream>

#include <fstream>

#include<iomanip>

using namespace std;

//////////////////////////////

int main()

{

ifstream in("word.txt");

ofstream out("result.txt");

char x;

int q = 0;

if (!in)

{

cout << "Can't open file" << endl;

exit(-1);

}

while (in >> x && q != 7)

{

out << x;

switch (q)

{

case 0: if (x == 'A') q = 1;

else q = 7; break;

case 1: if (x == 'B') q = 2;

else q = 7; break;

case 2: if ((x == 'C') || ( x == 'D')) q = 3;

else q = 7; break;

case 3: if (x == 'A') q = 4;

else q = 7; break;

case 4: if ((x == 'Q') || (x == 'P') || (x == 'R')) q = 5;

else if ((x == 'C') || (x == 'D')) q = 3;

else q = 7; break;

case 5: if (x == 'E') q = 6;

else if ((x == 'C') || (x == 'D')) q = 3;

else q = 7; break;

default: q = 7; break;

}

out << setw(2) << q << endl;

}

out << endl;

if (q == 6) out << " sequence is element of L" << endl;

else out << "sequence is not element of L" << endl;

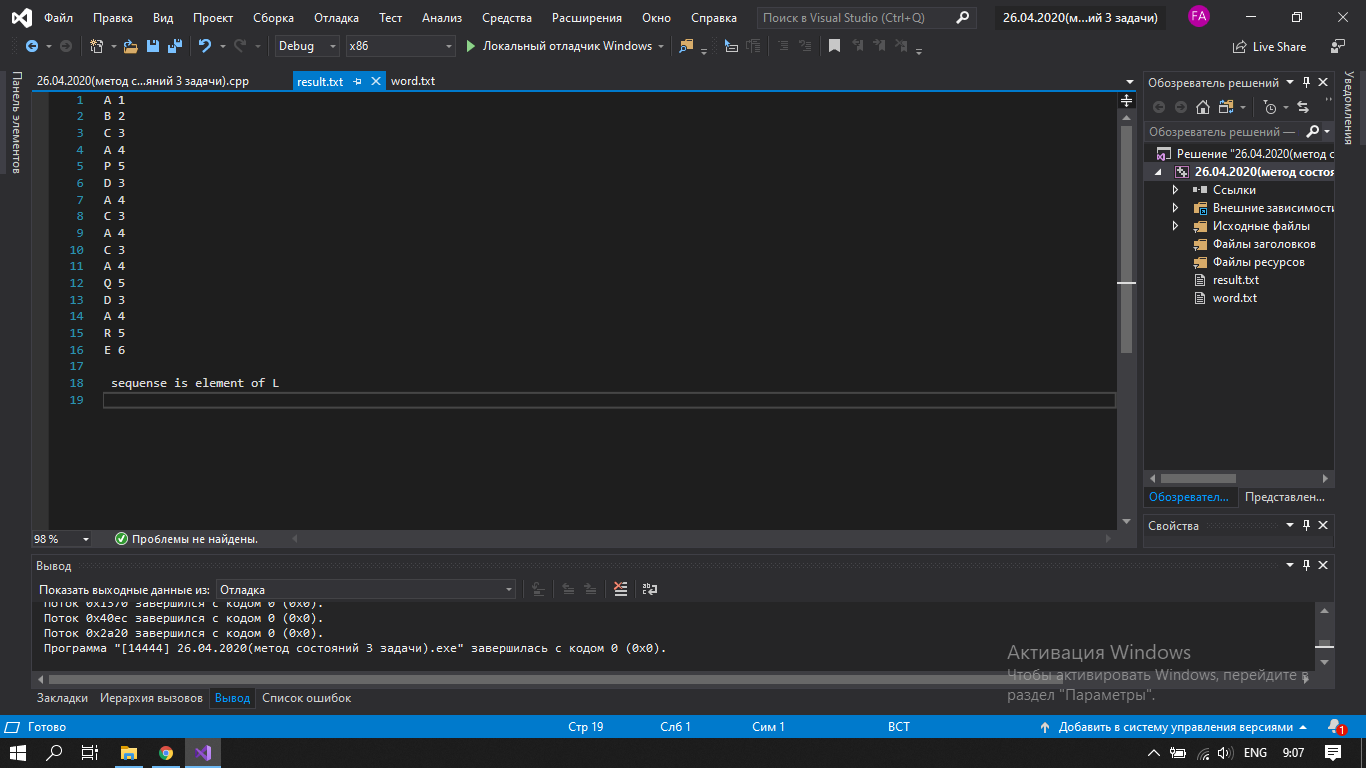
in.close();

out.close();

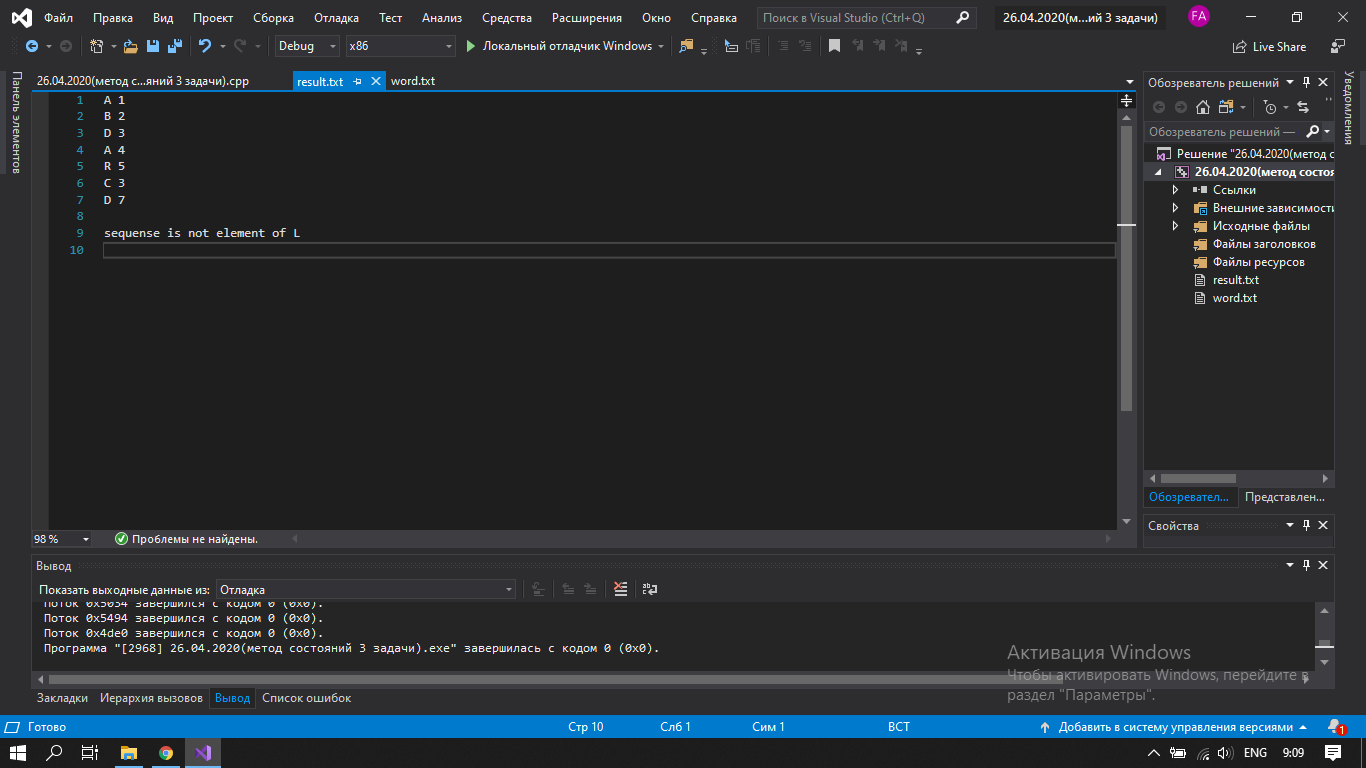
}

Пример работы программы:

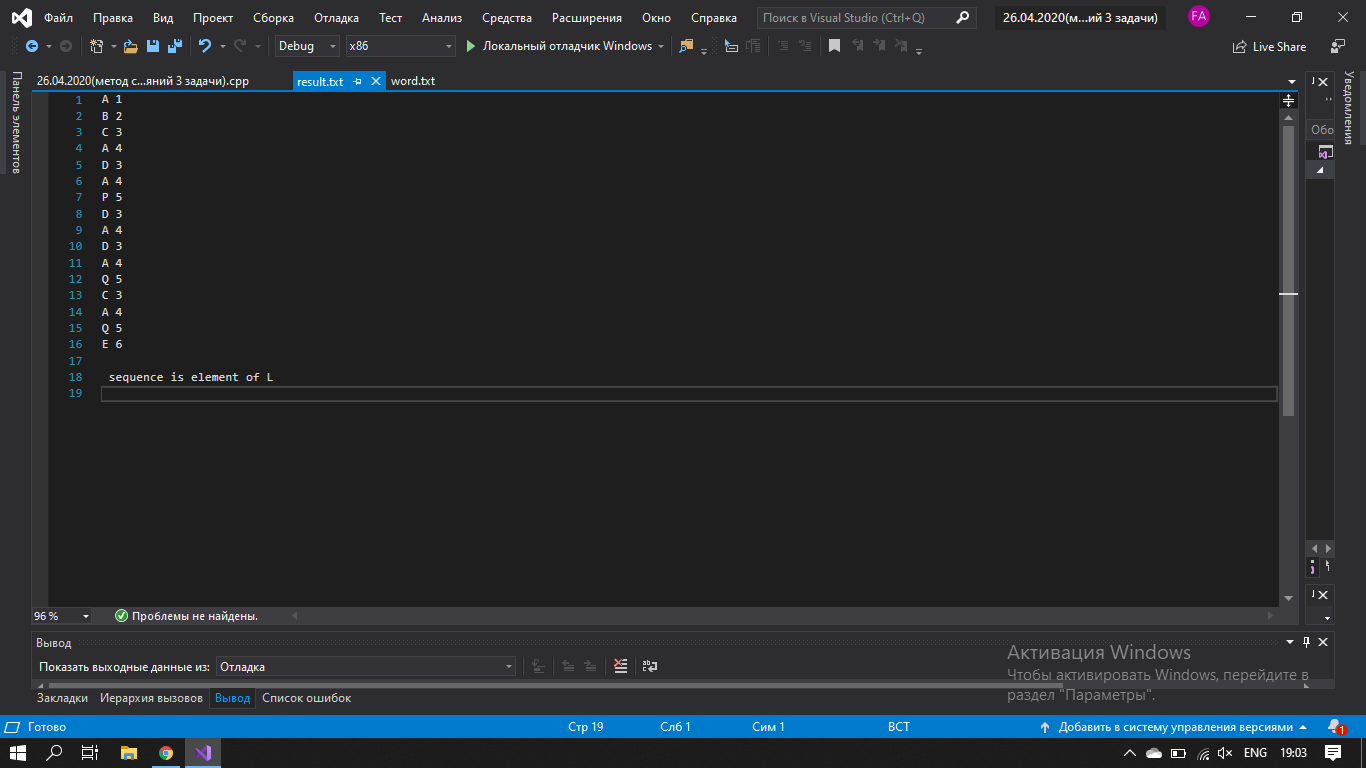
В файле “word.txt”: ABCAPDACACAQDARE



В файле “word.txt”: ABDARCDAQE



В файле “word.txt”: ABCADAPDADAQCAQE



# Моделирование машины Тьюринга

## Постановка задачи

Всякая машина Тьюринга M состоит из:

* - ячейки с бесконечным в обе стороны числом разрядов …, , , , , , … В каждом разряде может быть записан один из n символов множества A = , , … , , причем символ называется пустым (n2);
* - устройство, которое в каждый момент времени обозревает один из разрядов ячейки, а само может находиться в состояниях { , , … , } = Q; состояние называется «стоп-состояние»;
* - программы работы – это таблица из n строк и m-1 столбцов. Строки соответствуют символам из множества A , столбцы – символам из Q; в каждой клетке помещен один из символов A, один из символов Q и одна из букв Л, П, Н.

Машина предназначена для переработки информации в ячейке. Переработка совершается тактами, номер такта обозначается буквой t (t = 1,2,3, …). Перед первым тактом устройство устанавливается в состояние q1 на обозревание разряда x0, а начальная информация записывается в ячейку так, что в разрядах , , … , – не пустые символы, в остальных разрядах – пустой символ .

В общем случае – если перед тактом t устройство оказалось в состоянии q Q и обозревает разряд xl , то работа одного такта сводится к следующему:

* - если q = , то машина останавливается и информация в ячейке считается результатом работы машины;
* - если q q0, то берется символ a из разряда и находится клетка в программе, соответствующая символу a и состоянию q, - в этой клетке три символа (a , q , R);
* - в разряд помещается символ a (вместо a); устройство переходит в состояние q и начинает обозревать разряд , если R = П, или , если R = Л, или , если R = Н.

По завершении такта t машина автоматически переходит к выполнению действий такта t+1.

Задание состоит в том, чтобы построить программу, моделирующую работу конкретной машины Тьюринга и решающую для этой машины задачу вычисления унарной функции(Задание 1-2) и шифрования символьной последовательности(Задание 3).

### Задание 1

Реализовать функцию:

Код программы:

#include<iostream>

#include<fstream>

#include<cmath>

#include<iomanip>

using namespace std;

int main()

{

const int n = 100, kol\_sost = 6;

const int number\_comand = 12;

char a[n];

struct zap

{

char sym;

int sost;

char move;// R-right L-left H-halt

};

zap tabl[kol\_sost][100];

int i, p;

int sost = 1;

for (i = 0; i < n; i++)

a[i] = '\_';

ifstream in("turing.txt");

ofstream out("out.txt");

char sym\_in, sym\_out;

int sost\_in, sost\_out;

char mv;

if (!in)

{

cout << "Can't open file " << endl;

system("pause");

exit(-1);

}

for (i = 0; i < number\_comand; i++)

{

in >> sym\_in >> sym\_out >> mv >> sost\_in >> sost\_out;

tabl[sost\_in][sym\_in].sost = sost\_out;

tabl[sost\_in][sym\_in].sym = sym\_out;

tabl[sost\_in][sym\_in].move = mv;

}

//cout << endl << tabl[10]['1'].sost << tabl[10]['1'].sym << tabl[10]['1'].move << endl;

in.close();

out << " TURING MACHINES (ALAN TURING 1937) " << endl;

out << setw(7) << 1 << setw(5) << 2 << setw(5) << 3 << setw(8) << 4 << setw(7) << 5 << setw(6) << 6 << endl;

out << '\_';

for (sost\_in = 1; sost\_in <= kol\_sost; sost\_in++)

out << setw(4) << tabl[sost\_in]['\_'].sym << tabl[sost\_in]['\_'].sost

<< tabl[sost\_in]['\_'].move;

out << endl;

out << '1';

for (sost\_in = 1; sost\_in <= kol\_sost; sost\_in++)

out << setw(4) << tabl[sost\_in]['1'].sym << tabl[sost\_in]['1'].sost

<< tabl[sost\_in]['1'].move;

out << endl;

in.close();

cout << endl << "enter input word; number of symbols <=20; symbol '\_' is a last symbol " << endl;

cout << "alfabet ['\_','1'] " << endl;

cout << endl << "input word " << endl;

for (i = 50; i < n; i++)

{

cin >> a[i];

if (a[i] == '\_') break;

}

out << "input word" << endl;

for (i = 0; i < n; i++) out << a[i];

out << endl;

i = 50; int t = 0;

while (sost && (t < 1000))

{

char r = tabl[sost][a[i]].sym;

int s = tabl[sost][a[i]].sost;

mv = tabl[sost][a[i]].move;

sost = s;

a[i] = r;

if (mv == 'R') i++; else

if (mv == 'L') i--;

t++;

};

out << endl << "output word" << endl;

for (i = 0; i < n; i++) out << a[i];

out << endl;

out.close();

system("pause");

return 0;

}

Файл turing.txt

1\_R 1 2

11R 2 2

11R 3 3

11L 4 5

11L 5 5

11L 6 6

\_\_H 1 0

\_\_R 2 3

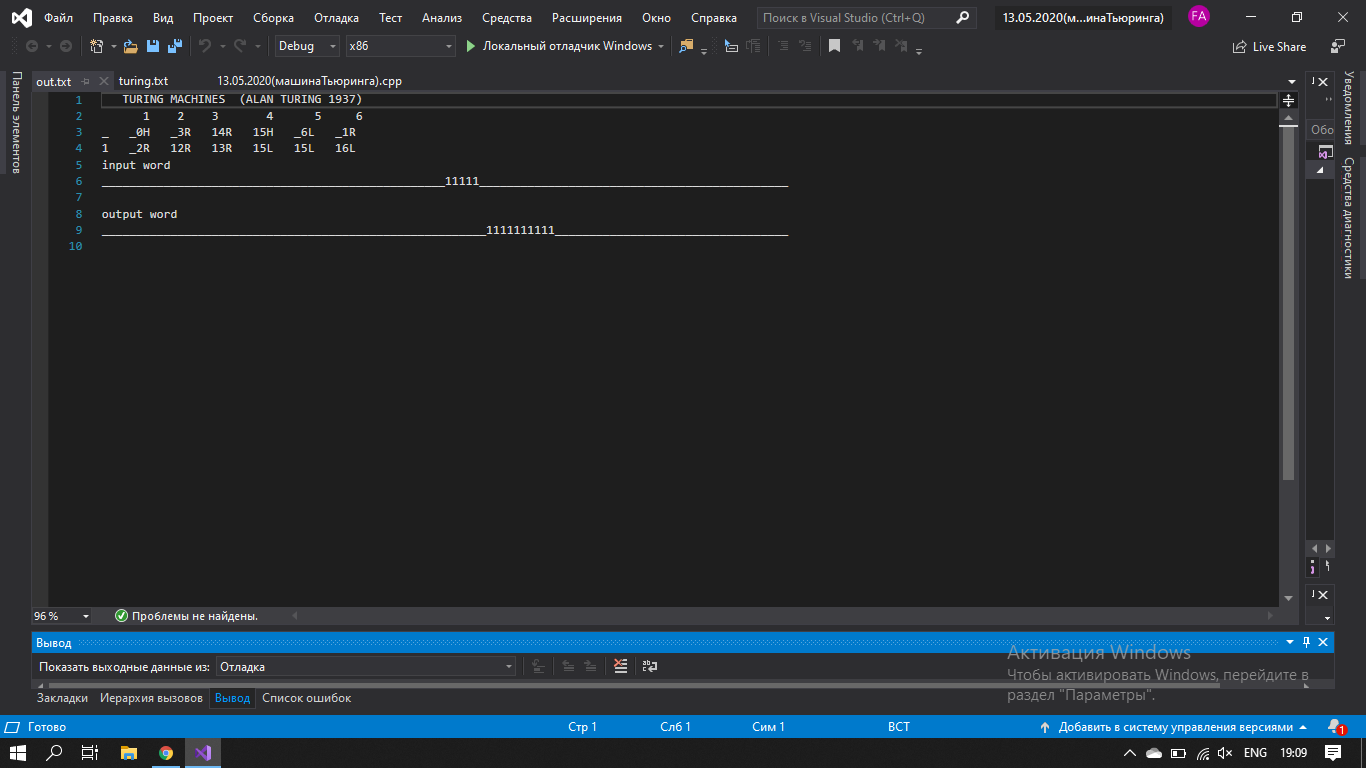
\_1R 3 4

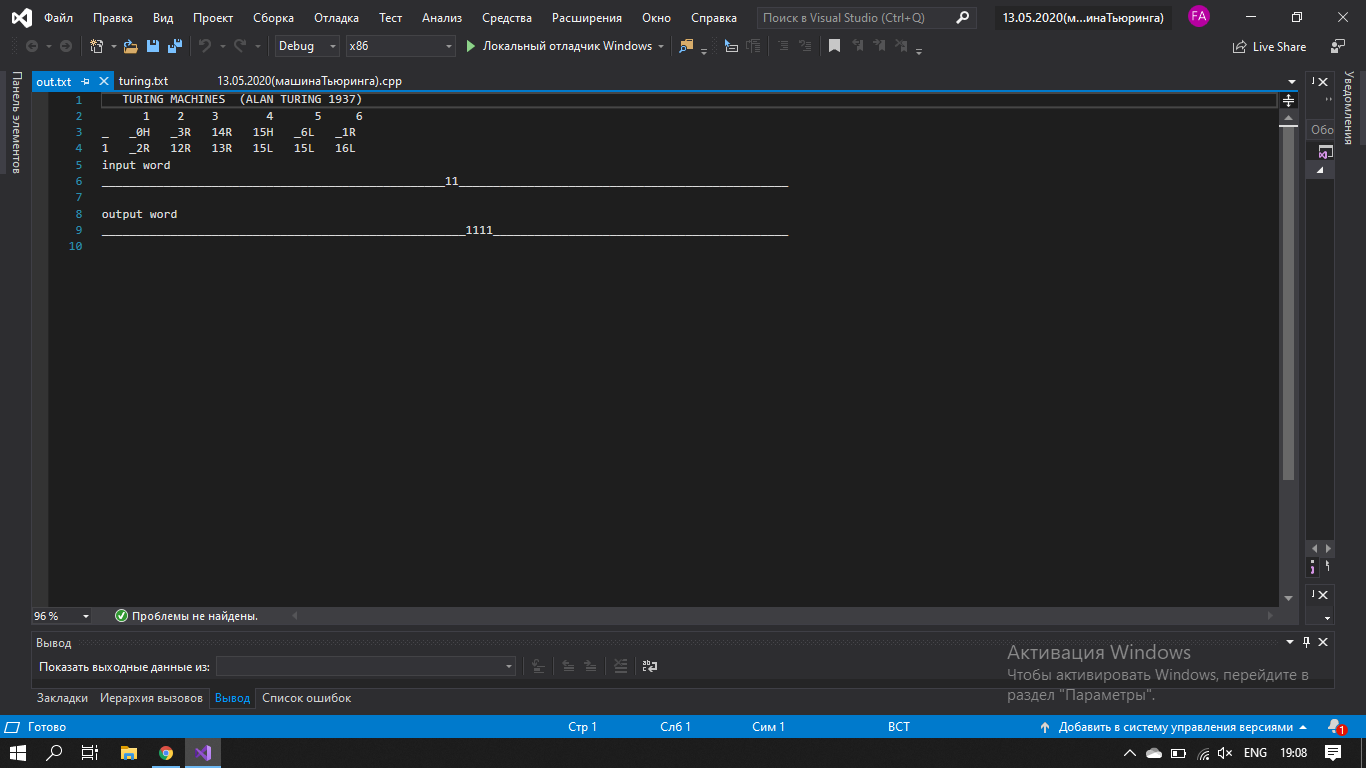
\_1H 4 5

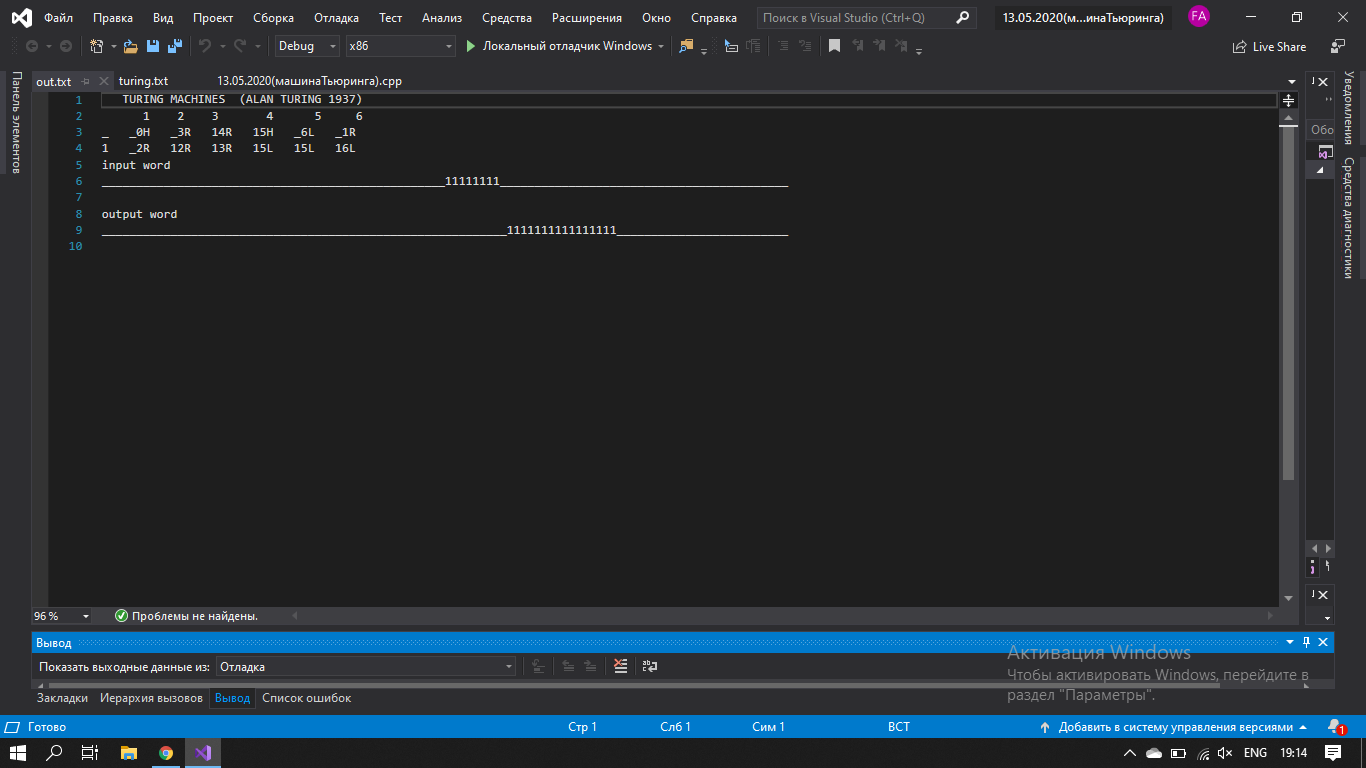
\_\_L 5 6

\_\_R 6 1

Пример работы программы:







### Задание 2

Реализовать функцию:

⎨

Состояния, которые функция не отрабатывает, в таблице вывести как “OOO”.

Код программы:

#include<iostream>

#include<fstream>

#include<cmath>

#include<iomanip>

using namespace std;

int main()

{

const int n = 100, kol\_sost = 22;

const int number\_comand = 44;

char a[n];

struct zap

{

char sym;

int sost;

char move;// R-right L-left H-halt

};

zap tabl[kol\_sost][100];

int i, p;

int sost = 1;

for (i = 0; i < n; i++)

a[i] = '\_';

ifstream in("turing.txt");

ofstream out("out.txt");

char sym\_in, sym\_out;

int sost\_in, sost\_out;

char mv;

if (!in)

{

cout << "Can't open file " << endl;

system("pause");

exit(-1);

}

for (i = 0; i < number\_comand; i++)

{

in >> sym\_in >> sym\_out >> mv >> sost\_in >> sost\_out;

tabl[sost\_in][sym\_in].sost = sost\_out;

tabl[sost\_in][sym\_in].sym = sym\_out;

tabl[sost\_in][sym\_in].move = mv;

}

in.close();

out << " TURING MACHINES (ALAN TURING 1937) " << endl;

out << setw(10) << 1 << setw(9) << 2 << setw(9) << 3 << setw(9) << 4 << setw(9) << 5 << setw(9) << 6 << setw(9) << 7 << setw(9) << 8 << setw(9) << 9 << setw(9) << 10 << setw(9) << 11 << setw(9) << 12 << setw(9) << 13 << setw(9) << 14 << setw(9) << 15 << setw(9) << 16 << setw(9) << 17 << setw(9) << 18 << setw(9) << 19 << setw(9) << 20 << setw(9) << 21 << setw(9) << 22 << endl;

out << '\_';

for (sost\_in = 1; sost\_in <= kol\_sost; sost\_in++)

{

if (tabl[sost\_in]['\_'].sost == (-858993460)) { out << setw(7) << 'O' << 'O' << 'O'; }

else

{

if (tabl[sost\_in]['\_'].sost>9) out << setw(6) << tabl[sost\_in]['\_'].sym << tabl[sost\_in]['\_'].sost

<< tabl[sost\_in]['\_'].move;

else out << setw(7) << tabl[sost\_in]['\_'].sym << tabl[sost\_in]['\_'].sost

<< tabl[sost\_in]['\_'].move;

}

}

out << endl;

out << '1';

for (sost\_in = 1; sost\_in <= kol\_sost; sost\_in++)

{

if (tabl[sost\_in]['1'].sost == (-858993460)) { out << setw(7) << 'O' << 'O' << 'O'; }

else

{

if (tabl[sost\_in]['1'].sost>9) out << setw(6) << tabl[sost\_in]['1'].sym << tabl[sost\_in]['1'].sost

<< tabl[sost\_in]['1'].move;

else out << setw(7) << tabl[sost\_in]['1'].sym << tabl[sost\_in]['1'].sost

<< tabl[sost\_in]['1'].move;

}

}

out << endl;

in.close();

cout << endl << "enter input word; number of symbols <=20; symbol '\_' is a last symbol " << endl;

cout << "alfabet ['\_','1'] " << endl;

cout << endl << "input word " << endl;

for (i = 50; i < n; i++)

{

cin >> a[i];

if (a[i] == '\_') break;

}

out << "input word" << endl;

for (i = 0; i < n; i++) out << a[i];

out << endl;

i = 50; int t = 0;

while (sost && (t < 10000))

{

char r = tabl[sost][a[i]].sym;

int s = tabl[sost][a[i]].sost;

mv = tabl[sost][a[i]].move;

sost = s;

a[i] = r;

if (mv == 'R') i++; else

if (mv == 'L') i--;

t++;

};

out << endl << "output word" << endl;

for (i = 0; i < n; i++) out << a[i];

out << endl;

out.close();

system("pause");

return 0;

}

Файл turing.txt

\_\_H 1 0

\_\_R 2 3

\_1R 3 4

\_1R 4 5

\_1H 5 6

\_\_L 6 7

\_\_R 1 1

\_\_R 8 9

\_1R 9 10

\_1R 10 11

\_1R 11 12

\_1R 12 13

\_1R 13 14

\_1H 14 15

\_\_L 15 16

\_\_H 16 0

\_\_R 17 18

\_1R 18 19

\_1H 19 20

\_\_L 20 21

\_\_R 21 22

\_\_H 22 0

11R 1 2

11R 2 8

\_\_R 1 1

\_\_R 1 1

\_\_R 1 1

11L 6 6

1\_H 7 0

11R 8 17

\_\_R 1 1

\_\_R 1 1

\_\_R 1 1

\_\_R 1 1

\_\_R 1 1

\_\_R 1 1

11L 15 15

1\_L 16 16

11R 17 17

11R 18 18

\_\_R 1 1

11L 20 20

11L 21 21

1\_R 22 17

Пример работы программы:

TURING MACHINES (ALAN TURING 1937)

TURING MACHINES (ALAN TURING 1937)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

\_ \_1R \_3R 14R 15R 16H \_7L OOO \_9R 110R 111R 112R 113R 114R 115H \_16L \_0H \_18R 119R 120H \_21L \_22R \_0H

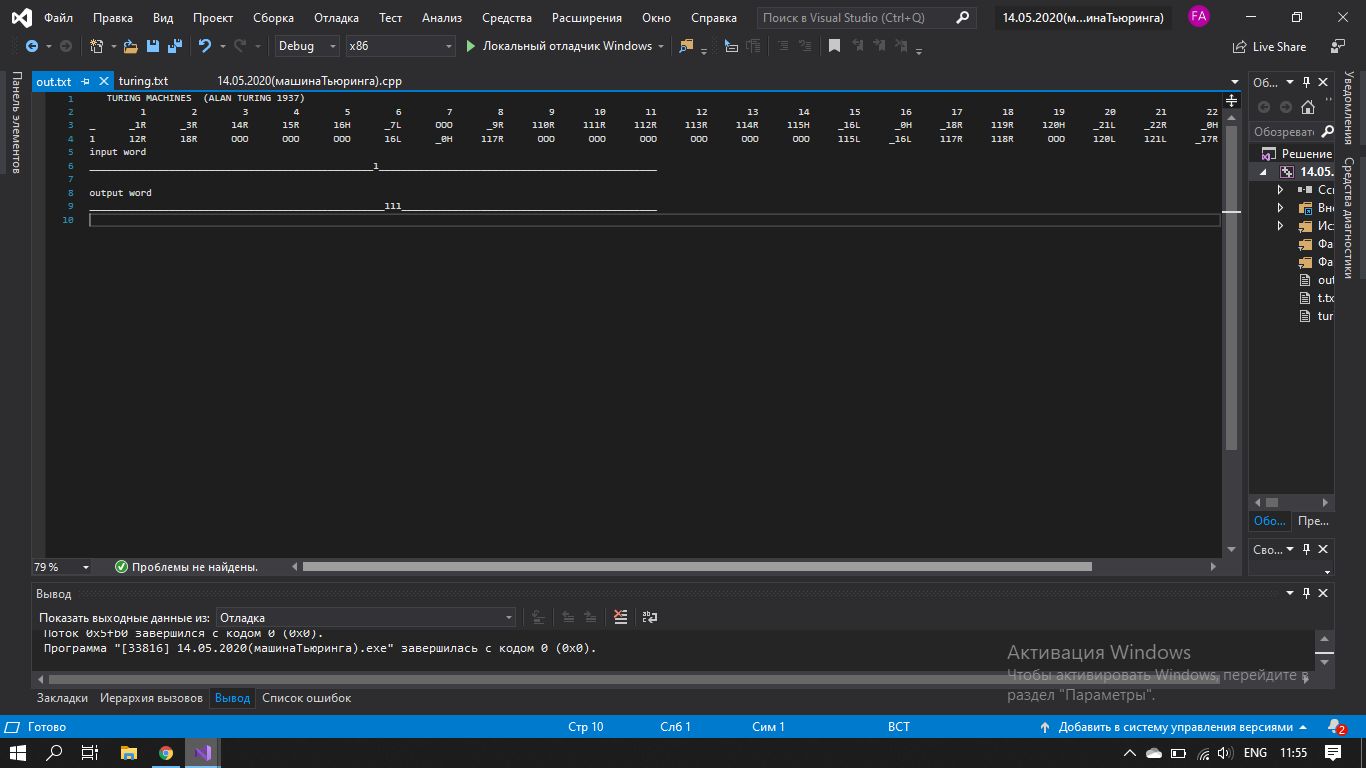
1 12R 18R OOO OOO OOO 16L \_0H 117R OOO OOO OOO OOO OOO OOO 115L \_16L 117R 118R OOO 120L 121L \_17R

input word

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

output word

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_111\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



TURING MACHINES (ALAN TURING 1937)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

\_ \_1R \_3R 14R 15R 16H \_7L OOO \_9R 110R 111R 112R 113R 114R 115H \_16L \_0H \_18R 119R 120H \_21L \_22R \_0H

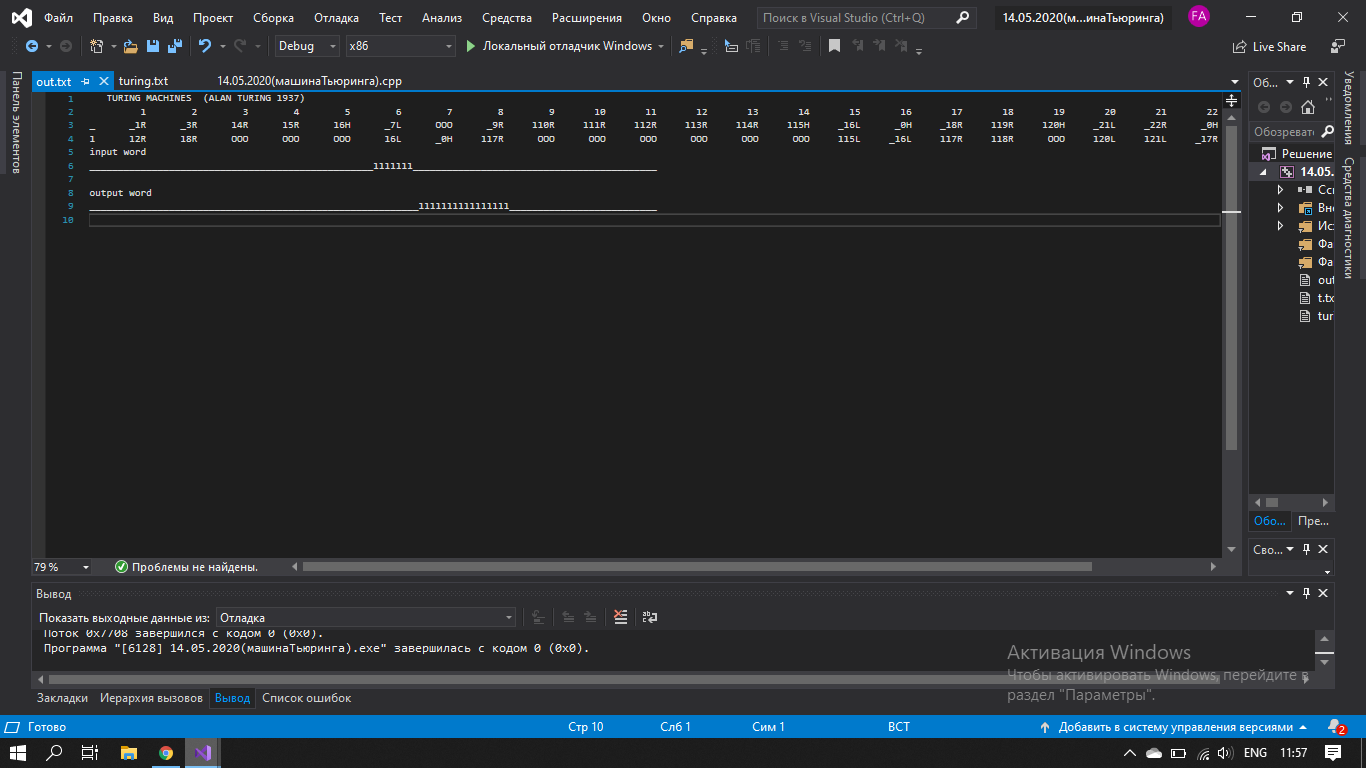
1 12R 18R OOO OOO OOO 16L \_0H 117R OOO OOO OOO OOO OOO OOO 115L \_16L 117R 118R OOO 120L 121L \_17R

input word

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11111111\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

output word

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_111111111111111111\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



### Задание 3

Зашифровать последовательность, состоящую из символов ‘a’, ‘b’, замещая ‘a’ на ‘x’, ‘b’ на ‘y’. Выходная последовательность должна содержать исходную и зашифрованную последовательности разделенные знаком ‘=’.

Алгоритм:

* вначале записываем знак ‘=’ за входным словом
* возвращаемся под первый символ входного слова
* заменяем видимый символ ‘a’(‘b’) на двойник ‘A’(‘B’)
* «бежим» вправо до первой свободной клетки и записываем в неё символ ‘a’(‘b’)
* возвращаемся влево к клетке с двойником ‘A’(‘B’)
* восстанавливаем прежний символ ‘a’(‘b’)
* сдвигаемся вправо к следующему символу
* аналогичным образом копируем второй символ (заменяем его на двойник, в конец дописываем копируемый символ и т.д.) и все последующие символы входного слова.
* когда мы скопируем последний символ входного слова и вернемся к его двойнику, то затем после сдвига на одну позицию вправо мы попадем на знак ‘=’. Это сигнал о том, что входное слово полностью скопировано, поэтому работу МТ надо завершать.

Состояния, которые функция не отрабатывает, в таблице вывести как “OOO”.

Код программы:

#include<iostream>

#include<fstream>

#include<cmath>

#include<iomanip>

using namespace std;

const int n = 255, kol\_sost = 6, ch\_num = 8;

const int number\_comand = 28;

char alf[ch\_num] = { 'a', 'b', 'A', 'B', 'x', 'y', '=', '\_' }; // алфавит

struct zap

{

char sym;

int sost;

char move;// R-right L-left H-halt

};

void Print\_sost(ofstream & out, zap tabl[kol\_sost][n]) // функция вывода в файл таблицы состояний

{

out << " TURING MACHINES (ALAN TURING 1937) " << endl;

out << setw(10) << 1 << setw(9) << 2 << setw(9) << 3 << setw(9) << 4 << setw(9) << 5 << setw(9) << 6 << endl;

for (int i = 0; i < ch\_num; i++)

{

out << alf[i]; // входной символ

for (int sost\_in = 1; sost\_in <= kol\_sost; sost\_in++)

{

if (tabl[sost\_in][alf[i]].sost == (-858993460)) { out << setw(7) << 'O' << 'O' << 'O'; } // если такого состояния нет(пустое), то в ячейку таблицы запишем OOO

else

{

// условие предназначено для ровного вывода таблицы

if (tabl[sost\_in][alf[i]].sost > 9) out << setw(6) << tabl[sost\_in][alf[i]].sym << tabl[sost\_in][alf[i]].sost

<< tabl[sost\_in][alf[i]].move; // печатаем выходной символ, следующее состояние и действие

else out << setw(7) << tabl[sost\_in][alf[i]].sym << tabl[sost\_in][alf[i]].sost

<< tabl[sost\_in][alf[i]].move;

}

}

out << endl;

}

}

int main()

{

char a[n];

zap tabl[kol\_sost][n];

int i, p;

int sost = 1;

for (i = 0; i < n; i++)

a[i] = '\_';

ifstream in("turing.txt");

ofstream out("out.txt");

char sym\_in, sym\_out;

int sost\_in, sost\_out;

char mv;

if (!in)

{

cout << "Can't open file " << endl;

system("pause");

exit(-1);

}

for (i = 0; i < number\_comand; i++)

{

in >> sym\_in >> sym\_out >> mv >> sost\_in >> sost\_out;

tabl[sost\_in][sym\_in].sost = sost\_out;

tabl[sost\_in][sym\_in].sym = sym\_out;

tabl[sost\_in][sym\_in].move = mv;

}

in.close();

Print\_sost(out, tabl);

cout << endl << "enter input word; number of symbols <=20; symbol '\_' is a last symbol " << endl;

cout << "alfabet ['\_','a', 'b'] " << endl;

cout << endl << "input word " << endl;

for (i = 50; i < n; i++)

{

cin >> a[i];

if (a[i] == '\_') break;

}

out << "input word" << endl;

for (i = 0; i < n; i++) out << a[i];

out << endl;

i = 50; int t = 0;

while (sost && (t < 1000)) // количество тактов увеличено, иначе последовательности не дообрабатываются алгоритмом

{

char r = tabl[sost][a[i]].sym;

int s = tabl[sost][a[i]].sost;

mv = tabl[sost][a[i]].move;

sost = s;

a[i] = r;

if (mv == 'R') i++; else

if (mv == 'L') i--;

t++;

};

out << endl << "output word" << endl;

for (i = 0; i < n; i++) out << a[i];

out << endl;

out.close();

system("pause");

return 0;

}

Файл turing.txt

aaR 1 1

aaL 2 2

aAR 3 4

aaR 4 4

aaR 5 5

aaL 6 6

bbR 1 1

bbL 2 2

bBR 3 5

bbR 4 4

bbR 5 5

bbL 6 6

==R 3 0

==R 4 4

==R 5 5

==L 6 6

AaR 6 3

BbR 6 3

xxR 4 4

xxR 5 5

xxL 6 6

yyR 4 4

yyR 5 5

yyL 6 6

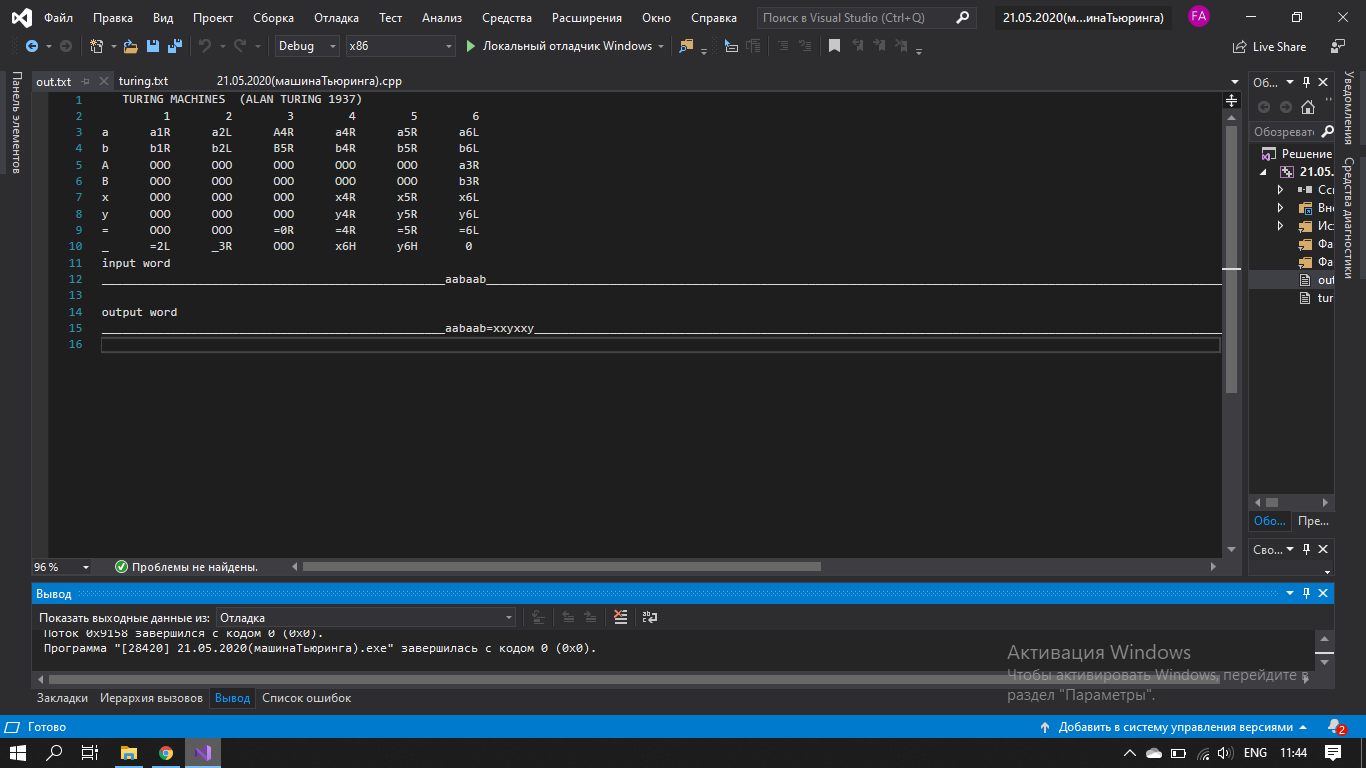
\_=L 1 2

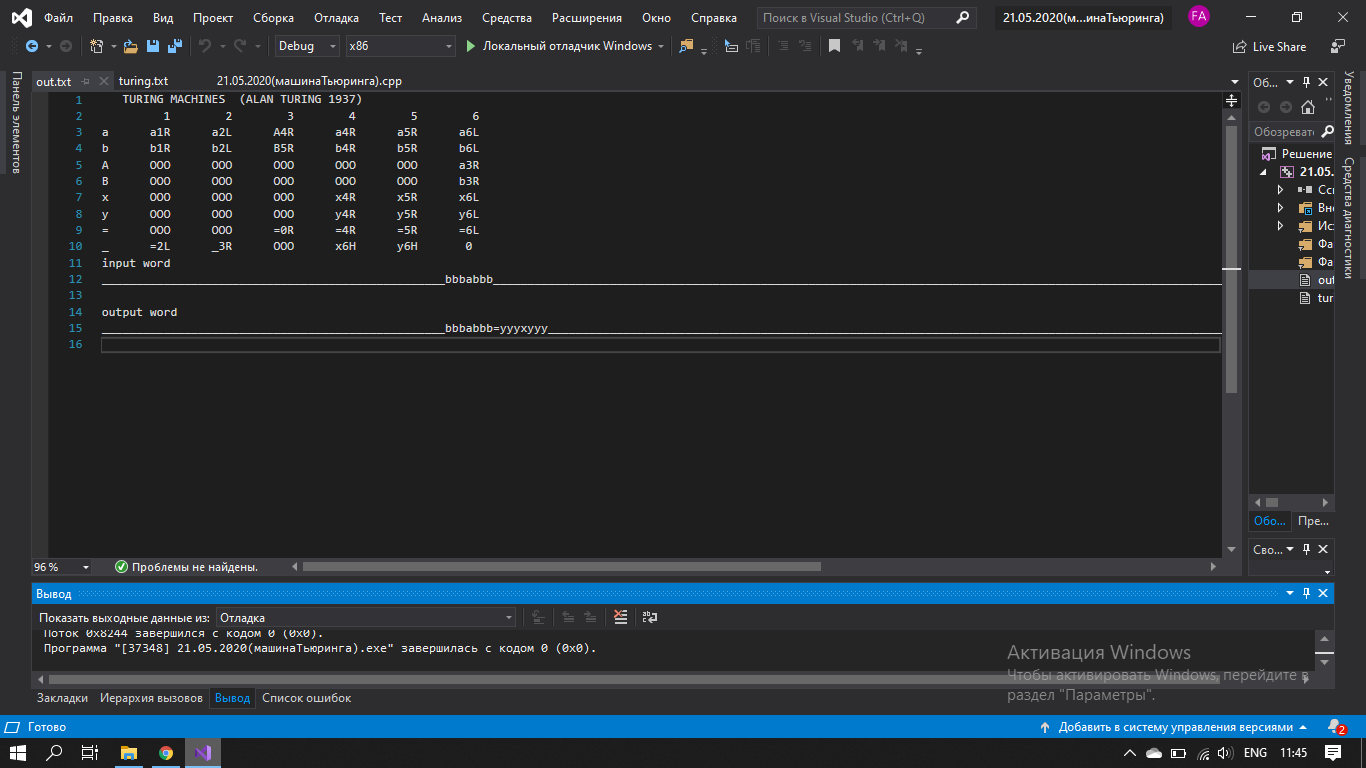
\_\_R 2 3

\_xH 4 6

\_yH 5 6

Пример работы программы:





# Вывод

В течение семестра мы изучили работу с бинарными файлами(считывание, запись), создавали пользовательские структуры, проверяли синтаксис языка с помощью метода состояний и моделировали работу конкретной машины Тьюринга, решающую задачи вычисления унарной функции и шифрования последовательности. Также были изучены функции работы с односвязным и двусвязным списками, алгебра множеств, алгебра полиномов(мономов).

# Список использованных ресурсов

1. Машина Тьюринга и алгоритмы Маркова. Решение задач: учебно-методическое пособие/В.Н. Пильщиков, В.Г. Абрамов, А.А. Вылиток, И.В. Горячая. - Москва, Издательский отдел факультета вычислительной математики и кибернетики МГУ им. М.В. Ломоносова, 2006.(Дата обращения 14.05.2020)
2. Машина Тьюринга [Электронный ресурс] <https://goto7.ru/tyring/tyring.html> (Дата обращения 21.05.2020).