МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт информационных технологий

*наименование института(факультета)*

Кафедра математического и программного обеспечения ЭВМ

*наименование кафедры*

Теория автоматов и формальных языков.

*наименование дисциплины в соответствии с учебным планом*

ЛАБОРАТОРНАЯ РАБОТА №3-4

Построение таблиц идентификаторов

|  |  |
| --- | --- |
| Исполнитель |  |
| Студент | 1ПИб-02-1оп-22 |
|  | *группа* |
|  | Харламов Д.А |
|  | *ФИО* |
| Руководитель | Ганичева О. Г. |
|  | *ФИО преподавателя* |
| Оценка |  |
| Подпись |  |

2024 год

ЗАДАНИЕ

Написать часть/функцию  лексического анализатора, обрабатывающую  входную строку  по конечному автомату  на основе оператора выбора (эта часть из предыдущей работы )  и **дополнить ее формированием таблиц с классами лексем.**

**Результат работы программы:**

**1. считываемая входная строка (код в соответствие со своим вариантом)**

**2. таблицы  с выделенными классами лексем (в которых содержится информация для формирования дискрипторного и псевдокодов).**

**3. результаты тестирования этой функции (проверка на повторное вхождение лексемы в заданный класс, отнесение к правильному классу лексем, выдача ошибки в случае неверной лексемы и др. )**

Ход работы:

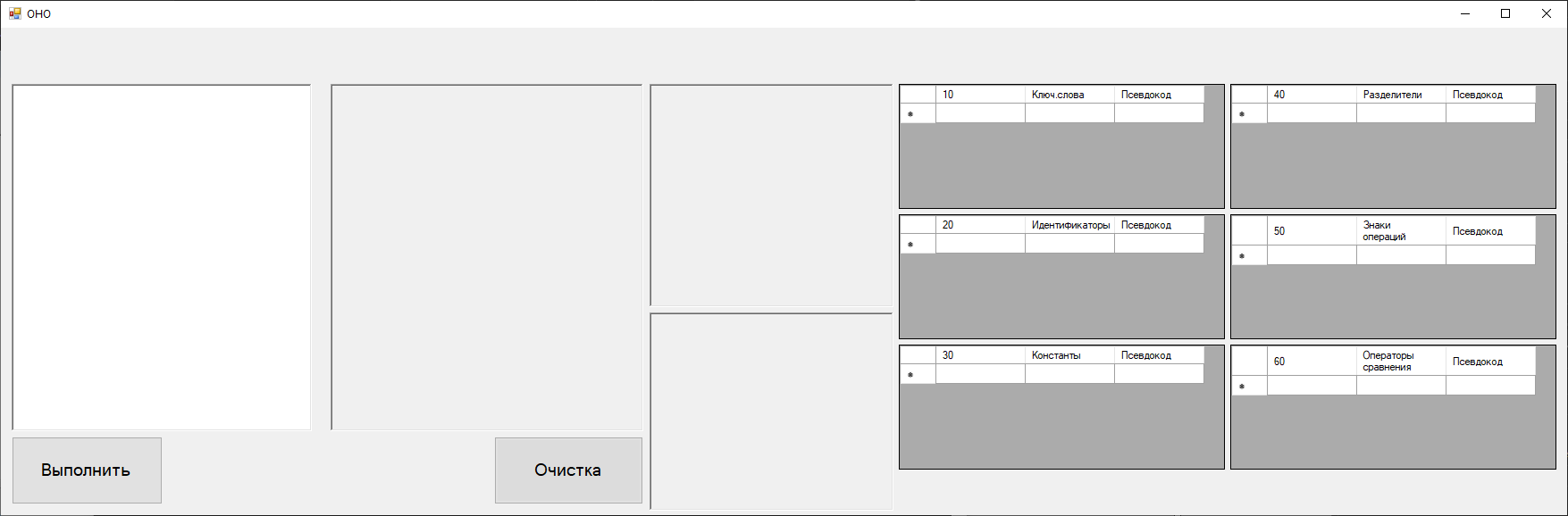


Рис. 11. Интерфейс программы

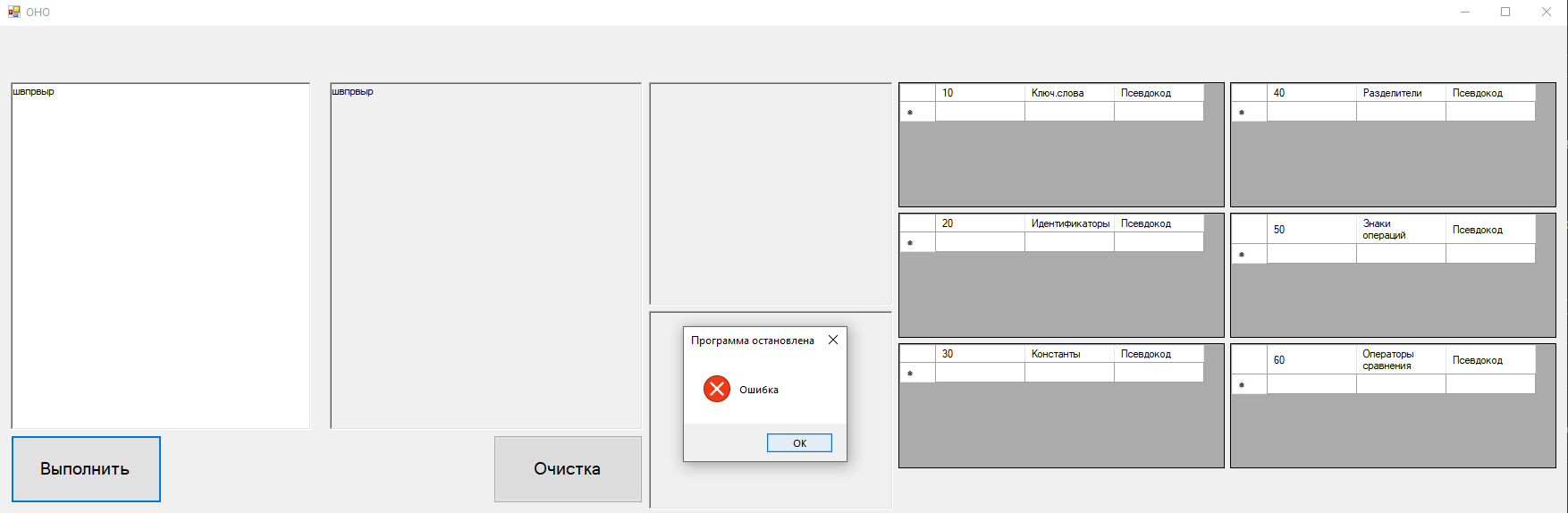


Рис. 11. Ошибка работы программы

Тесты работы программы:

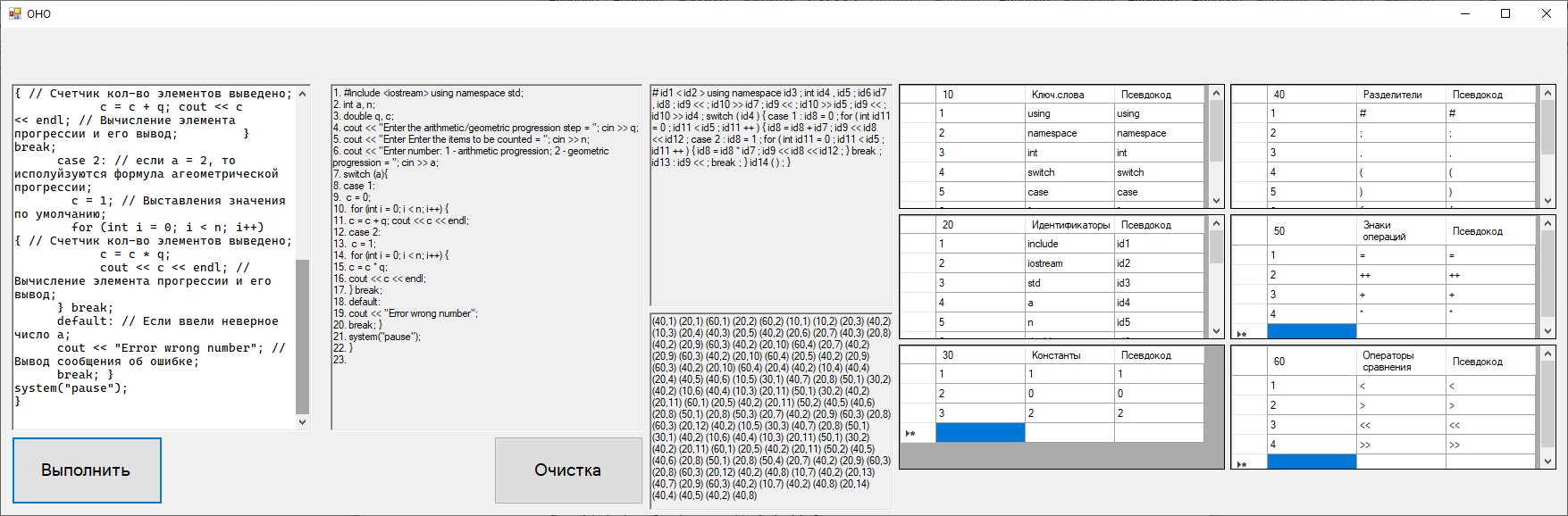


Рис. 12.

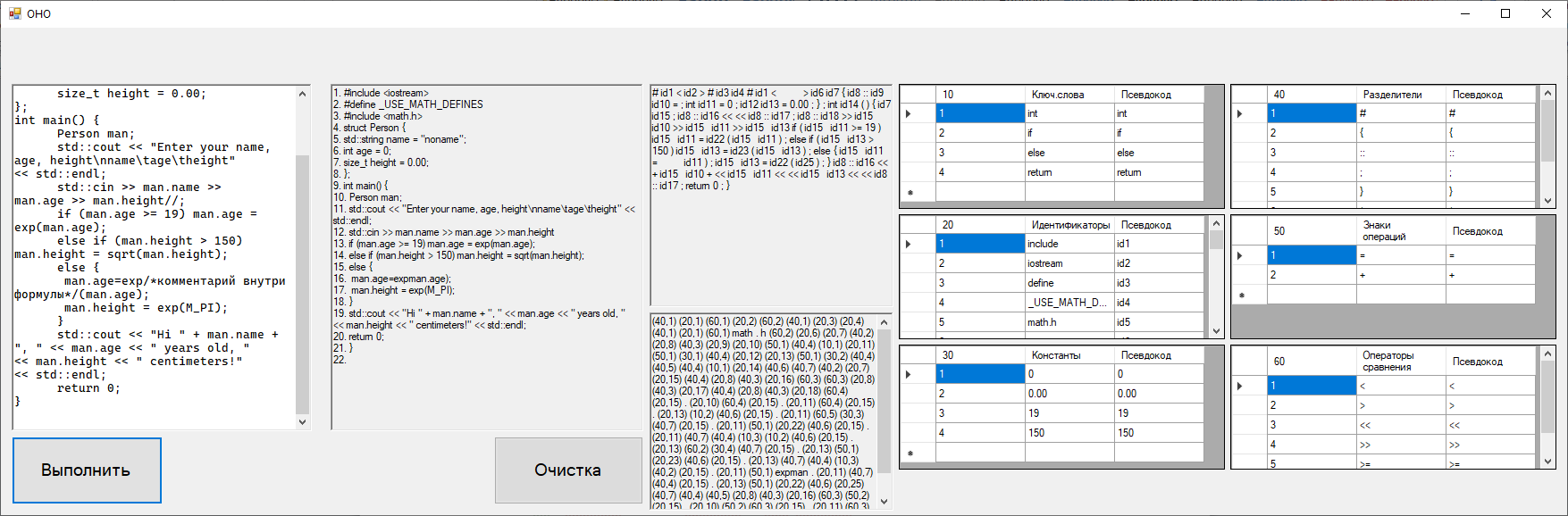


Рис. 13.

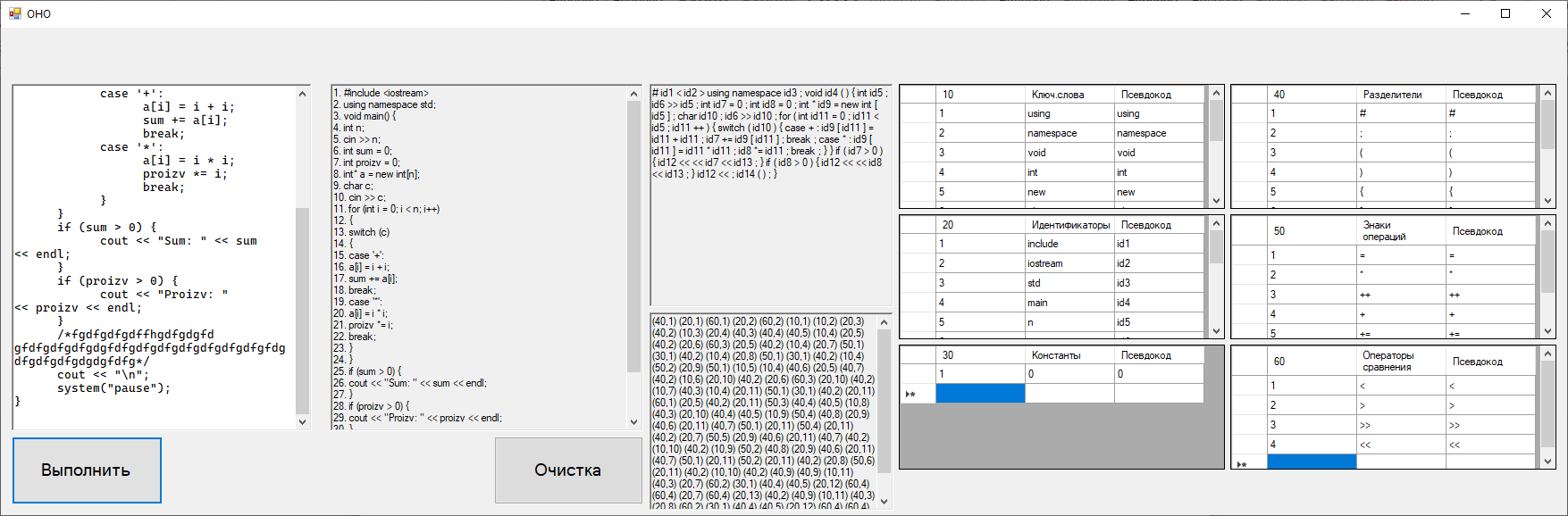


Рис. 14.

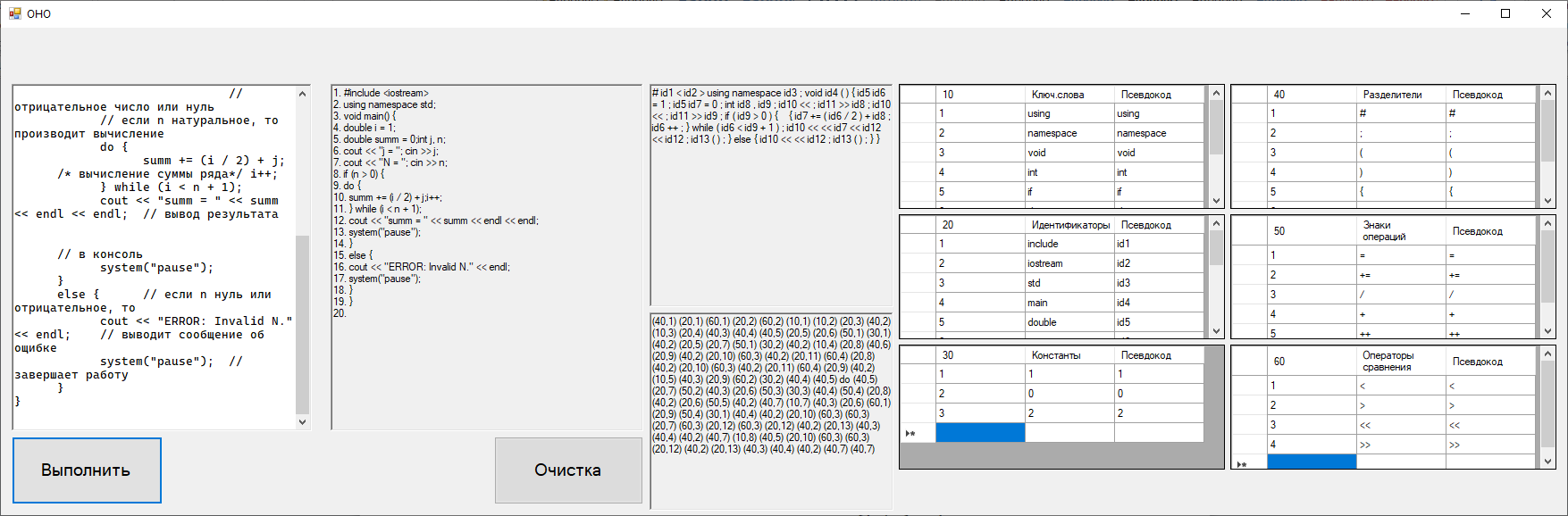


Рис. 15.

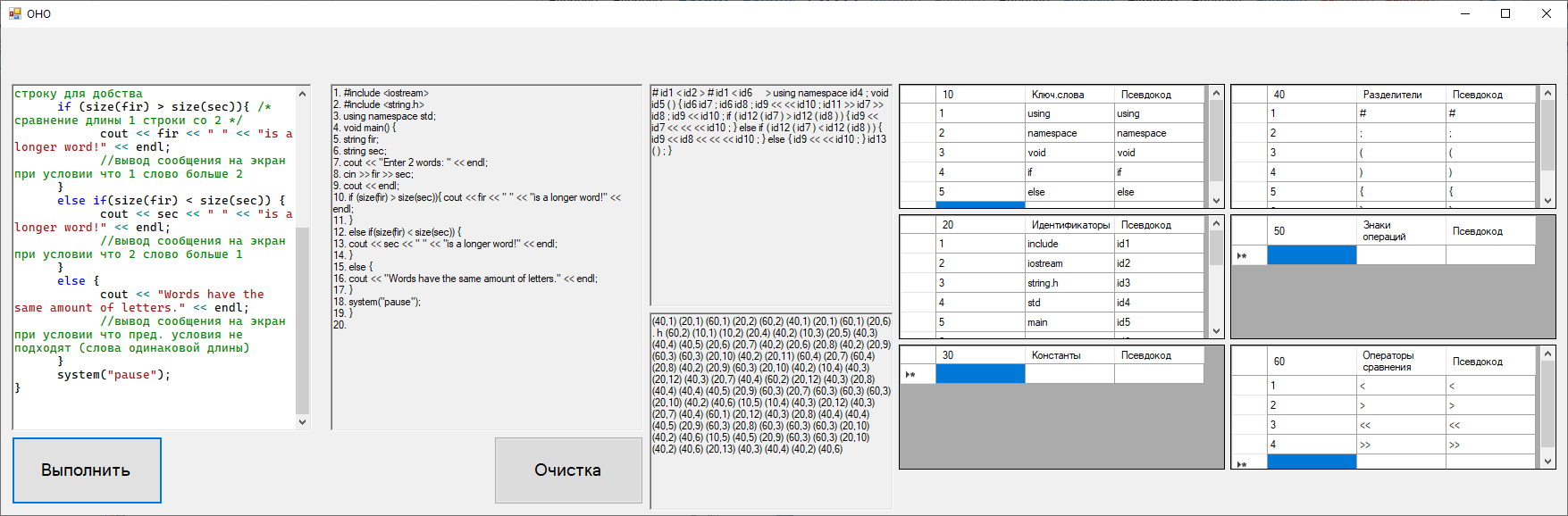


Рис. 15.

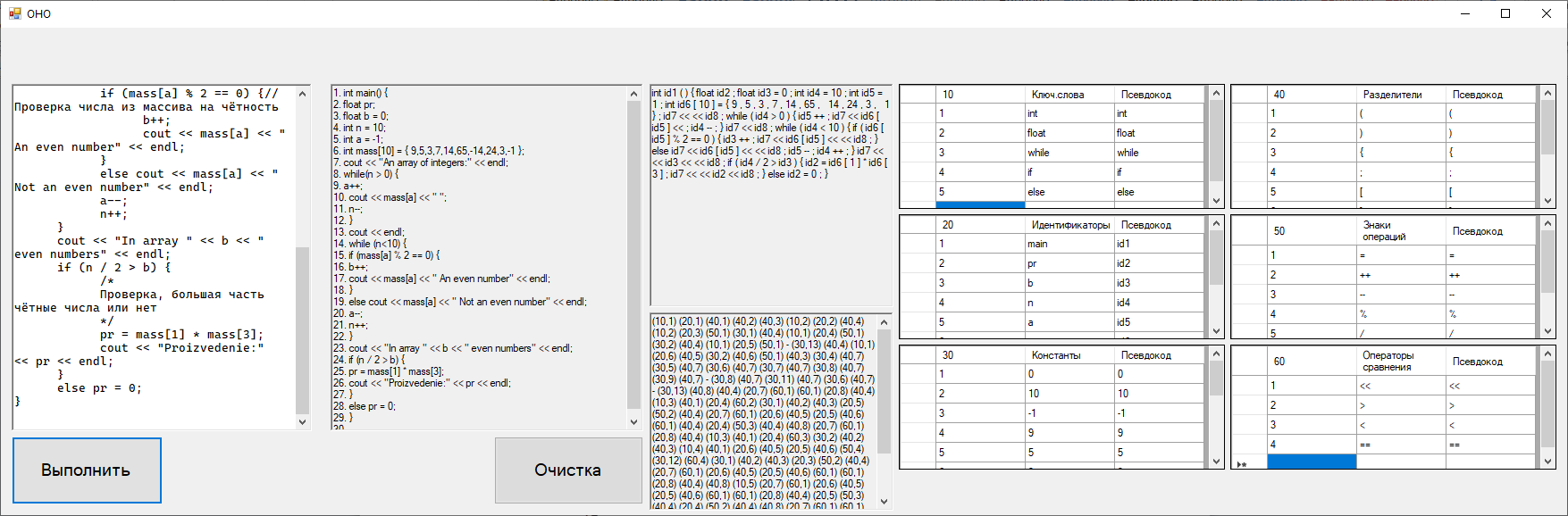


Рис. 15.

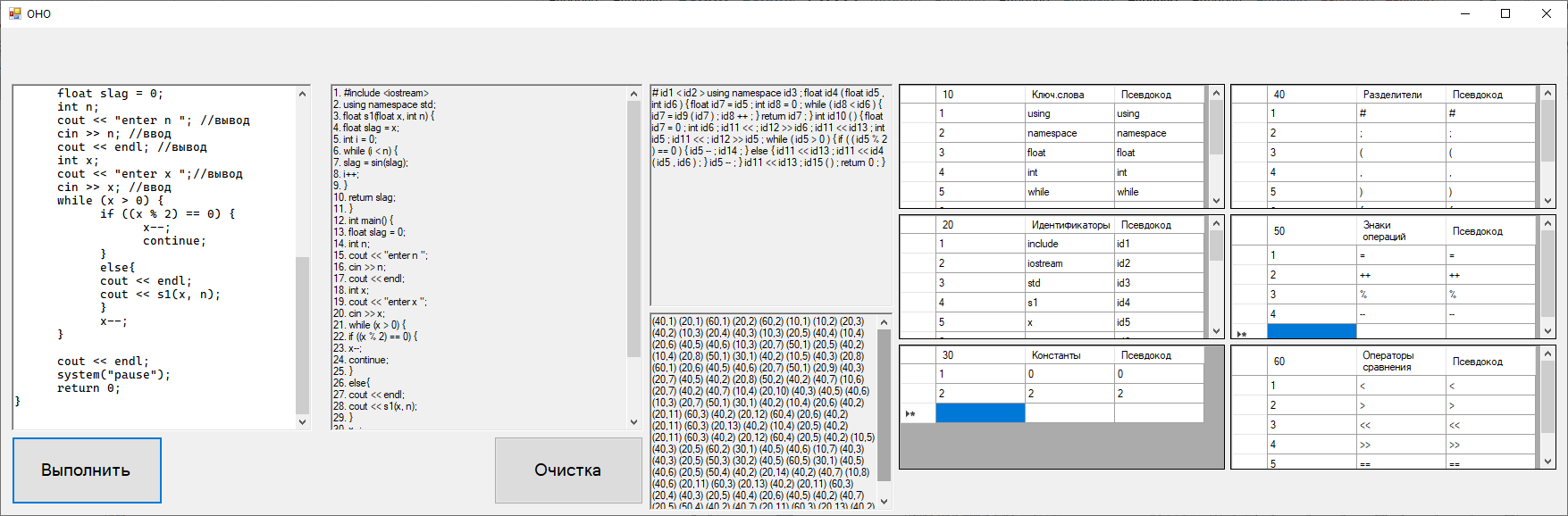


Рис. 15.

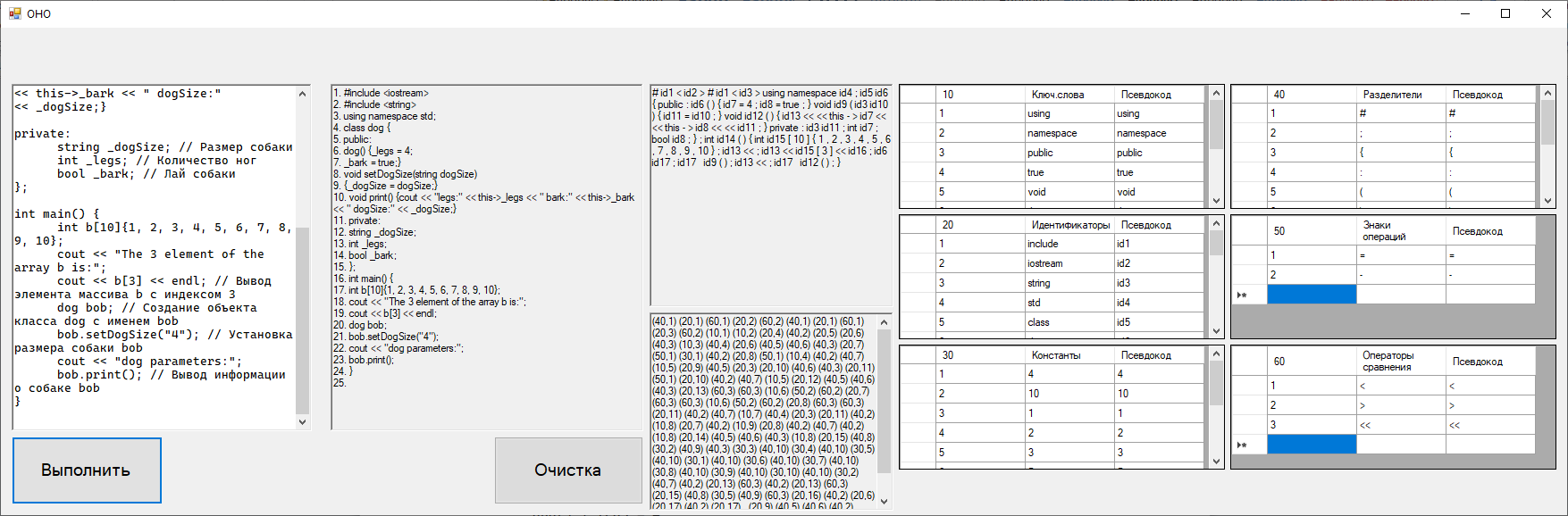


Рис. 15.

Код программы:

#pragma once

#include <string>

namespace A1 {

using namespace std;

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Сводка для MyForm

/// </summary>

public ref class MyForm : public System::Windows::Forms::Form

{

public:

MyForm(void)

{

InitializeComponent();

}

private: System::Windows::Forms::Label^ label1;

private: System::Windows::Forms::RichTextBox^ richTextBox1;

private: System::Windows::Forms::RichTextBox^ richTextBox2;

private: System::Windows::Forms::Button^ button1;

private: System::Windows::Forms::Button^ button2;

private:System::ComponentModel::Container^ components;

private: System::Windows::Forms::DataGridView^ t\_lexme;

private: System::Windows::Forms::DataGridView^ t\_cnst;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Clex1;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Clex2;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Ccnst1;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Ccnst2;

private: System::Windows::Forms::DataGridView^ t\_id;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Cidet1;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Cidet2;

private: System::Windows::Forms::DataGridView^ t\_op\_sings;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn1;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn2;

private: System::Windows::Forms::DataGridView^ t\_op\_comp;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn3;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn4;

private: System::Windows::Forms::DataGridView^ t\_sep;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn5;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ dataGridViewTextBoxColumn6;

private: System::Windows::Forms::RichTextBox^ pseTextBox3;

private: System::Windows::Forms::RichTextBox^ desTextBox4;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Clex3;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Ccnst;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Cidet;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column1;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column2;

private: System::Windows::Forms::DataGridViewTextBoxColumn^ Column3;

private: System::Windows::Forms::Label^ label2;

protected:

~MyForm()

{

if (components)

{

delete components;

}

}

private:

bool isnum(wchar\_t tmp)

{

return (tmp == '1' || tmp == '2' || tmp == '3' || tmp == '4' || tmp == '5' || tmp == '6' || tmp == '7' || tmp == '8' || tmp == '9') ? true : false;

}

bool skip\_sings(wchar\_t tmp)

{

return (tmp == ' ' || tmp == '\n' || tmp == '\r') ? true : false;

}

bool op\_comp(wchar\_t tmp)

{

return (tmp == '<' || tmp == '>' || tmp == '=' || tmp == '!') ? true : false;

}

bool op\_sings(wchar\_t tmp)

{

return (tmp == '=' || tmp == '-' || tmp == '+' || tmp == '\*' || tmp == '/' || tmp == '%' || tmp == '|' || tmp == '&') ? true : false;

}

bool spec\_chr(wchar\_t tmp)

{

return (tmp == ';' || tmp == '(' || tmp == ')' || tmp == '{' || tmp == '}' || tmp == ':' || tmp == '#' || tmp == ',' || tmp == '[' || tmp == ']' || tmp == '.') ? true : false;

}

bool stop\_sing(wchar\_t tmp)

{

return (spec\_chr(tmp) || op\_comp(tmp) || op\_sings(tmp) || skip\_sings(tmp)) ? true : false;

}

bool allchar(wchar\_t tmp)

{

return (tmp == '\_' || tmp == 'a' || tmp == 'b' || tmp == 'c' || tmp == 'd' || tmp == 'e' || tmp == 'f' || tmp == 'g' || tmp == 'h' || tmp == 'i' || tmp == 'j' || tmp == 'k' || tmp == 'l' || tmp == 'm'

|| tmp == 'n' || tmp == 'o' || tmp == 'p' || tmp == 'q' || tmp == 'r' || tmp == 's' || tmp == 't' || tmp == 'u' || tmp == 'v' || tmp == 'w' || tmp == 'x' || tmp == 'y' || tmp == 'z' || tmp == 'A'

|| tmp == 'B' || tmp == 'C' || tmp == 'D' || tmp == 'E' || tmp == 'F' || tmp == 'G' || tmp == 'H' || tmp == 'I' || tmp == 'J' || tmp == 'K' || tmp == 'L' || tmp == 'M' || tmp == 'N' || tmp == 'O'

|| tmp == 'P' || tmp == 'Q' || tmp == 'R' || tmp == 'S' || tmp == 'T' || tmp == 'U' || tmp == 'V' || tmp == 'W' || tmp == 'X' || tmp == 'Y' || tmp == 'Z' || isnum(tmp) || tmp == '0') ? true : false;

}

bool search\_c(String^ wrd, int rows) {

for (int j = 0; j <= rows - 1; j++) {

if (Convert::ToString(t\_cnst->Rows[j]->Cells[1]->Value) == wrd) return true;

}

return false;

}

bool search\_l(String^ wrd, int rows) {

for (int j = 0; j <= rows - 1; j++) {

if (Convert::ToString(t\_lexme->Rows[j]->Cells[1]->Value) == wrd) return true;

}

return false;

}

bool search\_id(String^ wrd, int rows) {

for (int j = 0; j <= rows - 1; j++) {

if (Convert::ToString(t\_id->Rows[j]->Cells[1]->Value) == wrd) return true;

}

return false;

}

bool search\_op\_s(String^ wrd, int rows) {

for (int j = 0; j <= rows - 1; j++) {

if (Convert::ToString(t\_op\_sings->Rows[j]->Cells[1]->Value) == wrd) return true;

}

return false;

}

bool search\_comp(String^ wrd, int rows) {

for (int j = 0; j <= rows - 1; j++) {

if (Convert::ToString(t\_op\_comp->Rows[j]->Cells[1]->Value) == wrd) return true;

}

return false;

}

bool search\_separ(String^ wrd, int rows) {

for (int j = 0; j <= rows - 1; j++) {

if (Convert::ToString(t\_sep->Rows[j]->Cells[1]->Value) == wrd) return true;

}

return false;

}

void add\_cnst(String^ wrd, int &rows)

{

if (!search\_c(wrd, rows)) {

t\_cnst->RowCount = rows + 1;

t\_cnst->Rows[rows - 1]->Cells[0]->Value = Convert::ToString(rows);

t\_cnst->Rows[rows - 1]->Cells[1]->Value = wrd;

t\_cnst->Rows[rows - 1]->Cells[2]->Value = wrd;

rows++;

}

}

void add\_lex(String^ wrd, int &rows)

{

if (!search\_l(wrd, rows)) {

t\_lexme->RowCount = rows + 1;

t\_lexme->Rows[rows - 1]->Cells[0]->Value = Convert::ToString(rows);

t\_lexme->Rows[rows - 1]->Cells[1]->Value = wrd;

t\_lexme->Rows[rows - 1]->Cells[2]->Value = wrd;

rows++;

}

}

void add\_idet(String^ wrd, int& rows)

{

if (!search\_id(wrd, rows)) {

t\_id->RowCount = rows + 1;

t\_id->Rows[rows - 1]->Cells[0]->Value = Convert::ToString(rows);

t\_id->Rows[rows - 1]->Cells[1]->Value = wrd;

t\_id->Rows[rows - 1]->Cells[2]->Value = "id" + Convert::ToString(rows);

rows++;

}

}

void add\_op\_s(String^ wrd, int& rows)

{

if (!search\_op\_s(wrd, rows)) {

t\_op\_sings->RowCount = rows + 1;

t\_op\_sings->Rows[rows - 1]->Cells[0]->Value = Convert::ToString(rows);

t\_op\_sings->Rows[rows - 1]->Cells[1]->Value = wrd;

t\_op\_sings->Rows[rows - 1]->Cells[2]->Value = wrd;

rows++;

}

}

void add\_comp(String^ wrd, int& rows)

{

if (!search\_comp(wrd, rows)) {

t\_op\_comp->RowCount = rows + 1;

t\_op\_comp->Rows[rows - 1]->Cells[0]->Value = Convert::ToString(rows);

t\_op\_comp->Rows[rows - 1]->Cells[1]->Value = wrd;

t\_op\_comp->Rows[rows - 1]->Cells[2]->Value = wrd;

rows++;

}

}

void add\_separ(String^ wrd, int& rows)

{

if (!search\_separ(wrd, rows)) {

t\_sep->RowCount = rows + 1;

t\_sep->Rows[rows - 1]->Cells[0]->Value = Convert::ToString(rows);

t\_sep->Rows[rows - 1]->Cells[1]->Value = wrd;

t\_sep->Rows[rows - 1]->Cells[2]->Value = wrd;

rows++;

}

}

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->label2 = (gcnew System::Windows::Forms::Label());

this->label1 = (gcnew System::Windows::Forms::Label());

this->richTextBox1 = (gcnew System::Windows::Forms::RichTextBox());

this->richTextBox2 = (gcnew System::Windows::Forms::RichTextBox());

this->button1 = (gcnew System::Windows::Forms::Button());

this->button2 = (gcnew System::Windows::Forms::Button());

this->t\_lexme = (gcnew System::Windows::Forms::DataGridView());

this->Clex1 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Clex2 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->t\_cnst = (gcnew System::Windows::Forms::DataGridView());

this->Ccnst1 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Ccnst2 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->t\_id = (gcnew System::Windows::Forms::DataGridView());

this->Cidet1 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Cidet2 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->t\_op\_sings = (gcnew System::Windows::Forms::DataGridView());

this->dataGridViewTextBoxColumn1 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn2 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->t\_op\_comp = (gcnew System::Windows::Forms::DataGridView());

this->dataGridViewTextBoxColumn3 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn4 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->t\_sep = (gcnew System::Windows::Forms::DataGridView());

this->dataGridViewTextBoxColumn5 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->dataGridViewTextBoxColumn6 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->pseTextBox3 = (gcnew System::Windows::Forms::RichTextBox());

this->desTextBox4 = (gcnew System::Windows::Forms::RichTextBox());

this->Clex3 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Ccnst = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Cidet = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Column1 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Column2 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

this->Column3 = (gcnew System::Windows::Forms::DataGridViewTextBoxColumn());

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->t\_lexme))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->t\_cnst))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->t\_id))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->t\_op\_sings))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->t\_op\_comp))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->t\_sep))->BeginInit();

this->SuspendLayout();

//

// label2

//

this->label2->AutoSize = true;

this->label2->Font = (gcnew System::Drawing::Font(L"Segoe UI Symbol", 14.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->label2->Location = System::Drawing::Point(471, 27);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(145, 25);

this->label2->TabIndex = 13;

this->label2->Text = L"Готовый текст";

//

// label1

//

this->label1->AutoSize = true;

this->label1->Font = (gcnew System::Drawing::Font(L"Segoe UI Symbol", 14.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->label1->Location = System::Drawing::Point(92, 27);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(158, 25);

this->label1->TabIndex = 14;

this->label1->Text = L"Исходный текст";

//

// richTextBox1

//

this->richTextBox1->Location = System::Drawing::Point(12, 63);

this->richTextBox1->Name = L"richTextBox1";

this->richTextBox1->Size = System::Drawing::Size(336, 389);

this->richTextBox1->TabIndex = 15;

this->richTextBox1->Text = L"";

//

// richTextBox2

//

this->richTextBox2->Location = System::Drawing::Point(369, 63);

this->richTextBox2->Name = L"richTextBox2";

this->richTextBox2->ReadOnly = true;

this->richTextBox2->Size = System::Drawing::Size(350, 389);

this->richTextBox2->TabIndex = 16;

this->richTextBox2->Text = L"";

//

// button1

//

this->button1->Font = (gcnew System::Drawing::Font(L"Segoe UI Symbol", 14.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->button1->Location = System::Drawing::Point(12, 458);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(169, 76);

this->button1->TabIndex = 17;

this->button1->Text = L"Выполнить";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &MyForm::button1\_Click);

//

// button2

//

this->button2->BackColor = System::Drawing::Color::Gainsboro;

this->button2->Font = (gcnew System::Drawing::Font(L"Segoe UI Symbol", 14.25F, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->button2->Location = System::Drawing::Point(552, 458);

this->button2->Name = L"button2";

this->button2->Size = System::Drawing::Size(167, 76);

this->button2->TabIndex = 20;

this->button2->Text = L"Очистка";

this->button2->UseVisualStyleBackColor = false;

this->button2->Click += gcnew System::EventHandler(this, &MyForm::button2\_Click);

//

// t\_lexme

//

this->t\_lexme->ColumnHeadersHeightSizeMode = System::Windows::Forms::DataGridViewColumnHeadersHeightSizeMode::AutoSize;

this->t\_lexme->Columns->AddRange(gcnew cli::array< System::Windows::Forms::DataGridViewColumn^ >(3) {

this->Clex1, this->Clex2,

this->Clex3

});

this->t\_lexme->Location = System::Drawing::Point(1005, 63);

this->t\_lexme->Name = L"t\_lexme";

this->t\_lexme->ReadOnly = true;

this->t\_lexme->Size = System::Drawing::Size(365, 140);

this->t\_lexme->TabIndex = 21;

//

// Clex1

//

this->Clex1->HeaderText = L"10";

this->Clex1->Name = L"Clex1";

this->Clex1->ReadOnly = true;

//

// Clex2

//

this->Clex2->HeaderText = L"Ключ.слова";

this->Clex2->Name = L"Clex2";

this->Clex2->ReadOnly = true;

//

// t\_cnst

//

this->t\_cnst->ColumnHeadersHeightSizeMode = System::Windows::Forms::DataGridViewColumnHeadersHeightSizeMode::AutoSize;

this->t\_cnst->Columns->AddRange(gcnew cli::array< System::Windows::Forms::DataGridViewColumn^ >(3) {

this->Ccnst1, this->Ccnst2,

this->Ccnst

});

this->t\_cnst->Location = System::Drawing::Point(1005, 209);

this->t\_cnst->Name = L"t\_cnst";

this->t\_cnst->ReadOnly = true;

this->t\_cnst->Size = System::Drawing::Size(365, 140);

this->t\_cnst->TabIndex = 22;

//

// Ccnst1

//

this->Ccnst1->HeaderText = L"20";

this->Ccnst1->Name = L"Ccnst1";

this->Ccnst1->ReadOnly = true;

//

// Ccnst2

//

this->Ccnst2->HeaderText = L"Константы";

this->Ccnst2->Name = L"Ccnst2";

this->Ccnst2->ReadOnly = true;

//

// t\_id

//

this->t\_id->ColumnHeadersHeightSizeMode = System::Windows::Forms::DataGridViewColumnHeadersHeightSizeMode::AutoSize;

this->t\_id->Columns->AddRange(gcnew cli::array< System::Windows::Forms::DataGridViewColumn^ >(3) {

this->Cidet1, this->Cidet2,

this->Cidet

});

this->t\_id->Location = System::Drawing::Point(1005, 355);

this->t\_id->Name = L"t\_id";

this->t\_id->Size = System::Drawing::Size(365, 140);

this->t\_id->TabIndex = 23;

//

// Cidet1

//

this->Cidet1->HeaderText = L"30";

this->Cidet1->Name = L"Cidet1";

//

// Cidet2

//

this->Cidet2->HeaderText = L"Идентификаторы";

this->Cidet2->Name = L"Cidet2";

//

// t\_op\_sings

//

this->t\_op\_sings->ColumnHeadersHeightSizeMode = System::Windows::Forms::DataGridViewColumnHeadersHeightSizeMode::AutoSize;

this->t\_op\_sings->Columns->AddRange(gcnew cli::array< System::Windows::Forms::DataGridViewColumn^ >(3) {

this->dataGridViewTextBoxColumn1,

this->dataGridViewTextBoxColumn2, this->Column1

});

this->t\_op\_sings->Location = System::Drawing::Point(1376, 63);

this->t\_op\_sings->Name = L"t\_op\_sings";

this->t\_op\_sings->ReadOnly = true;

this->t\_op\_sings->Size = System::Drawing::Size(365, 140);

this->t\_op\_sings->TabIndex = 24;

//

// dataGridViewTextBoxColumn1

//

this->dataGridViewTextBoxColumn1->HeaderText = L"40";

this->dataGridViewTextBoxColumn1->Name = L"dataGridViewTextBoxColumn1";

this->dataGridViewTextBoxColumn1->ReadOnly = true;

//

// dataGridViewTextBoxColumn2

//

this->dataGridViewTextBoxColumn2->HeaderText = L"Знаки операций";

this->dataGridViewTextBoxColumn2->Name = L"dataGridViewTextBoxColumn2";

this->dataGridViewTextBoxColumn2->ReadOnly = true;

//

// t\_op\_comp

//

this->t\_op\_comp->ColumnHeadersHeightSizeMode = System::Windows::Forms::DataGridViewColumnHeadersHeightSizeMode::AutoSize;

this->t\_op\_comp->Columns->AddRange(gcnew cli::array< System::Windows::Forms::DataGridViewColumn^ >(3) {

this->dataGridViewTextBoxColumn3,

this->dataGridViewTextBoxColumn4, this->Column2

});

this->t\_op\_comp->Location = System::Drawing::Point(1376, 209);

this->t\_op\_comp->Name = L"t\_op\_comp";

this->t\_op\_comp->ReadOnly = true;

this->t\_op\_comp->Size = System::Drawing::Size(365, 140);

this->t\_op\_comp->TabIndex = 25;

//

// dataGridViewTextBoxColumn3

//

this->dataGridViewTextBoxColumn3->HeaderText = L"50";

this->dataGridViewTextBoxColumn3->Name = L"dataGridViewTextBoxColumn3";

this->dataGridViewTextBoxColumn3->ReadOnly = true;

//

// dataGridViewTextBoxColumn4

//

this->dataGridViewTextBoxColumn4->HeaderText = L"Операторы сравнения";

this->dataGridViewTextBoxColumn4->Name = L"dataGridViewTextBoxColumn4";

this->dataGridViewTextBoxColumn4->ReadOnly = true;

//

// t\_sep

//

this->t\_sep->ColumnHeadersHeightSizeMode = System::Windows::Forms::DataGridViewColumnHeadersHeightSizeMode::AutoSize;

this->t\_sep->Columns->AddRange(gcnew cli::array< System::Windows::Forms::DataGridViewColumn^ >(3) {

this->dataGridViewTextBoxColumn5,

this->dataGridViewTextBoxColumn6, this->Column3

});

this->t\_sep->Location = System::Drawing::Point(1376, 355);

this->t\_sep->Name = L"t\_sep";

this->t\_sep->ReadOnly = true;

this->t\_sep->Size = System::Drawing::Size(365, 140);

this->t\_sep->TabIndex = 26;

//

// dataGridViewTextBoxColumn5

//

this->dataGridViewTextBoxColumn5->HeaderText = L"60";

this->dataGridViewTextBoxColumn5->Name = L"dataGridViewTextBoxColumn5";

this->dataGridViewTextBoxColumn5->ReadOnly = true;

//

// dataGridViewTextBoxColumn6

//

this->dataGridViewTextBoxColumn6->HeaderText = L"Разделители";

this->dataGridViewTextBoxColumn6->Name = L"dataGridViewTextBoxColumn6";

this->dataGridViewTextBoxColumn6->ReadOnly = true;

//

// pseTextBox3

//

this->pseTextBox3->Location = System::Drawing::Point(726, 63);

this->pseTextBox3->Name = L"pseTextBox3";

this->pseTextBox3->ReadOnly = true;

this->pseTextBox3->Size = System::Drawing::Size(273, 250);

this->pseTextBox3->TabIndex = 27;

this->pseTextBox3->Text = L"";

//

// desTextBox4

//

this->desTextBox4->ImeMode = System::Windows::Forms::ImeMode::NoControl;

this->desTextBox4->Location = System::Drawing::Point(726, 319);

this->desTextBox4->Name = L"desTextBox4";

this->desTextBox4->ReadOnly = true;

this->desTextBox4->Size = System::Drawing::Size(273, 222);

this->desTextBox4->TabIndex = 28;

this->desTextBox4->Text = L"";

//

// Clex3

//

this->Clex3->HeaderText = L"Псевдокод";

this->Clex3->Name = L"Clex3";

this->Clex3->ReadOnly = true;

//

// Ccnst

//

this->Ccnst->HeaderText = L"Псевдокод";

this->Ccnst->Name = L"Ccnst";

this->Ccnst->ReadOnly = true;

//

// Cidet

//

this->Cidet->HeaderText = L"Псевдокод";

this->Cidet->Name = L"Cidet";

//

// Column1

//

this->Column1->HeaderText = L"Псевдокод";

this->Column1->Name = L"Column1";

this->Column1->ReadOnly = true;

//

// Column2

//

this->Column2->HeaderText = L"Псевдокод";

this->Column2->Name = L"Column2";

this->Column2->ReadOnly = true;

//

// Column3

//

this->Column3->HeaderText = L"Псевдокод";

this->Column3->Name = L"Column3";

this->Column3->ReadOnly = true;

//

// MyForm

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(1753, 546);

this->Controls->Add(this->desTextBox4);

this->Controls->Add(this->pseTextBox3);

this->Controls->Add(this->t\_sep);

this->Controls->Add(this->t\_op\_comp);

this->Controls->Add(this->t\_op\_sings);

this->Controls->Add(this->t\_id);

this->Controls->Add(this->t\_cnst);

this->Controls->Add(this->t\_lexme);

this->Controls->Add(this->button2);

this->Controls->Add(this->button1);

this->Controls->Add(this->richTextBox2);

this->Controls->Add(this->richTextBox1);

this->Controls->Add(this->label1);

this->Controls->Add(this->label2);

this->Name = L"MyForm";

this->Text = L"ОНО";

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->t\_lexme))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->t\_cnst))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->t\_id))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->t\_op\_sings))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->t\_op\_comp))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->t\_sep))->EndInit();

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

public:

int S = 0, a = 0; //S - состояние автомата, a - количество удаленных элементов

int tmpid; //временная переменная для обозначения начального элемента удаления

void prework()

{

String^ tmpcd = richTextBox1->Text; //обрабатываемый программой текстовый массив

for (int i = 0; i < richTextBox1->Text->Length; i++) {

switch (S) {

case 0:

switch (richTextBox1->Text[i]) {

case '/':

S = 2;

break;

case ' ': //пробел

S = 1;

break;

case '\t':

tmpcd = tmpcd->Remove(i - a, 1); //удаление табуляции

a += 1;

break;

case '\n': //перенос

S = 6;

break;

}

break;

case 1:

switch (richTextBox1->Text[i]) {

case ' ': //пробел повторяется вновь

tmpcd = tmpcd->Remove(i - a, 1);

a += 1;

break;

case '/':

S = 2;

break;

default:

S = 0;

break;

}

break;

case 2:

switch (richTextBox1->Text[i]) {

case '/': //второй слэш - однострочный

tmpid = i - 1; //начальный элемент, с которого осуществляется удаление

S = 3;

break;

case '\*': //звездочка полсе слэша - начало многострочного

tmpid = i - 1;

S = 4;

break;

default:

S = 0;

break;

}

break;

case 3:

switch (richTextBox1->Text[i]) {

case '\n': //переход на новую строку после однострочного комментария

tmpcd = tmpcd->Remove(tmpid - a, i - tmpid);

a += i - tmpid;

if (tmpid == 0 || tmpcd[i - a - 1] == '\n') {

tmpcd = tmpcd->Remove(i - a, 1);

a += 1;

}

S = 0;

break;

}

break;

case 4:

switch (richTextBox1->Text[i]) {

case '\*': //вторая звездочка - многострочный комментарий заканчивается

S = 5;

break;

}

break;

case 5:

switch (richTextBox1->Text[i]) {

case '/': //слэш после звезды - конец многострочного комментария

tmpcd = tmpcd->Remove(tmpid - a, i - tmpid + 2);

a += i - tmpid + 2;

S = 0;

break;

default: //возврат

S = 4;

break;

}

break;

case 6:

switch (richTextBox1->Text[i]) {

case '\n':

tmpcd = tmpcd->Remove(i - a, 1);

a += 1;

break;

case '/':

S = 2;

break;

case '\t': // табуляция

tmpcd = tmpcd->Remove(i - a, 1);

a += 1;

break;

default: // сброс

S = 0;

break;

}

break;

}

}

if (S == 4) {

tmpcd = tmpcd->Remove(tmpid - a, richTextBox1->Text->Length - tmpid);

}

a = 1;

/\* for (int i = 0; i < tmpcd->Length; i++) {

if (i == 0) {

tmpcd = tmpcd->Insert(i, a.ToString() + ". ");

a++;

}

if (tmpcd[i] == '\n') {

tmpcd = tmpcd->Insert(i + 1, a.ToString() + ". ");

a++;

}

}\*/

richTextBox2->Text = tmpcd;

}

void work(int &lex\_rows, int &cnst\_rows, int &idet\_rows, int &op\_rows, int &comp\_rows, int &sep\_rows)

{

int S = 0, line = 0;

String^ tmpcd = richTextBox2->Text;

String^ tmpwd = "";

for (int i = 0; i < richTextBox2->Text->Length; i++) {

switch (S) {

case 0:

if (skip\_sings(tmpcd[i])) S = 0;

else if (isnum(tmpcd[i])) S = 1;

else if (tmpcd[i] == '"') S = 351;

else if (tmpcd[i] == Convert::ToChar("'")) S = 399;

else if (tmpcd[i] == '0') S = 2;

else if (tmpcd[i] == 'a') S = 7;

else if (tmpcd[i] == 'b') S = 12;

else if (tmpcd[i] == 'c') S = 23;

else if (tmpcd[i] == 'd') S = 218;

else if (tmpcd[i] == 'e') S = 50;

else if (tmpcd[i] == 'f') S = 59;

else if (tmpcd[i] == 'i') S = 67;

else if (tmpcd[i] == 'n') S = 73;

else if (tmpcd[i] == 'o') S = 91;

else if (tmpcd[i] == 'p') S = 112;

else if (tmpcd[i] == 'r') S = 128;

else if (tmpcd[i] == 'g') S = 133;

else if (tmpcd[i] == 's') S = 136;

else if (tmpcd[i] == 't') S = 155;

else if (tmpcd[i] == 'u') S = 177;

else if (tmpcd[i] == 'v') S = 181;

else if (tmpcd[i] == 'w') S = 194;

else if (tmpcd[i] == 'm') S = 198;

else if (tmpcd[i] == 'x') S = 204;

else if (tmpcd[i] == '.' || tmpcd[i] == ',' || tmpcd[i] == '{' || tmpcd[i] == '}' || tmpcd[i] == '[' || tmpcd[i] == ']' || tmpcd[i] == '(' || tmpcd[i] == ')' || tmpcd[i] == ';' || tmpcd[i] == '#') S = 352;

else if (tmpcd[i] == ':') S = 206;

else if (tmpcd[i] == '<') S = 207;

else if (tmpcd[i] == '>') S = 208;

else if (tmpcd[i] == '|') S = 209;

else if (tmpcd[i] == '&') S = 210;

else if (tmpcd[i] == '=') S = 211;

else if (tmpcd[i] == '!') S = 212;

else if (tmpcd[i] == '+') S = 213;

else if (tmpcd[i] == '-') S = 214;

else if (tmpcd[i] == '?') S = 357;

else if (tmpcd[i] == '%') S = 215;

else if (tmpcd[i] == '\*') S = 216;

else if (tmpcd[i] == '/') S = 217;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 1:

if (isnum(tmpcd[i]) || tmpcd[i] == '0') S = 1;

else if (tmpcd[i] == '.') S = 3;

else if (stop\_sing(tmpcd[i])) S = 301;

else S = 400;

break;

case 2:

if (tmpcd[i] == '.') S = 3;

else if (stop\_sing(tmpcd[i])) S = 301;

else S = 400;

break;

case 3:

if (isnum(tmpcd[i]) || tmpcd[i] == '0') S = 4;

else S = 400;

break;

case 4:

if (isnum(tmpcd[i]) || tmpcd[i] == '0') S = 4;

else if (tmpcd[i] == 'E' || tmpcd[i] == 'e') S = 5;

else if (stop\_sing(tmpcd[i])) S = 301;

else S = 400;

break;

case 5:

if (isnum(tmpcd[i]) || tmpcd[i] == '0') S = 5;

if (tmpcd[i] == '+') S = 6;

else if (tmpcd[i] == '-') S = 6;

else S = 400;

break;

case 6:

if (isnum(tmpcd[i]) || tmpcd[i] == '0') S = 6;

else if (stop\_sing(tmpcd[i])) S = 301;

else S = 400;

break;

case 7:

if (tmpcd[i] == 's') S = 8;

if (tmpcd[i] == 'n') S = 9;

if (tmpcd[i] == 'u') S = 10;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 8:

if (tmpcd[i] == 'm') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 9:

if (tmpcd[i] == 'd') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 10:

if (tmpcd[i] == 't') S = 11;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 11:

if (tmpcd[i] == 'o') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 12:

if (tmpcd[i] == 'i') S = 13;

else if (tmpcd[i] == 'o') S = 18;

else if (tmpcd[i] == 'r') S = 20;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 13:

if (tmpcd[i] == 't') S = 14;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 14:

if (tmpcd[i] == 'a') S = 15;

else if (tmpcd[i] == 'o') S = 17;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 15:

if (tmpcd[i] == 'n') S = 16;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 16:

if (tmpcd[i] == 'd') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 17:

if (tmpcd[i] == 'r') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 18:

if (tmpcd[i] == 'o') S = 19;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 19:

if (tmpcd[i] == 'l') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 20:

if (tmpcd[i] == 'e') S = 21;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 21:

if (tmpcd[i] == 'a') S = 22;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 22:

if (tmpcd[i] == 'k') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 23:

if (tmpcd[i] == 'a') S = 24;

else if (tmpcd[i] == 'h') S = 28;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 24:

if (tmpcd[i] == 's') S = 25;

else if (tmpcd[i] == 't') S = 26;

else if (tmpcd[i] == 'l') S = 30;

else if (tmpcd[i] == 'o') S = 33;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 25:

if (tmpcd[i] == 'e') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 26:

if (tmpcd[i] == 'c') S = 27;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 27:

if (tmpcd[i] == 'h') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 28:

if (tmpcd[i] == 'a') S = 29;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 29:

if (tmpcd[i] == 'r') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 30:

if (tmpcd[i] == 'a') S = 31;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 31:

if (tmpcd[i] == 's') S = 32;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 32:

if (tmpcd[i] == 's') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 33:

if (tmpcd[i] == 'n') S = 34;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 34:

if (tmpcd[i] == 's') S = 35;

else if (tmpcd[i] == 't') S = 36;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 35:

if (tmpcd[i] == 't') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 36:

if (tmpcd[i] == 'i') S = 38;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 38:

if (tmpcd[i] == 'n') S = 39;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 39:

if (tmpcd[i] == 'u') S = 40;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 40:

if (tmpcd[i] == 'e') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 41:

if (tmpcd[i] == 'e') S = 42;

else if (tmpcd[i] == 'o') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 42:

if (tmpcd[i] == 'f') S = 43;

else if (tmpcd[i] == 'l') S = 47;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 43:

if (tmpcd[i] == 'a') S = 44;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 44:

if (tmpcd[i] == 'u') S = 45;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 45:

if (tmpcd[i] == 'l') S = 46;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 46:

if (tmpcd[i] == 't') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 47:

if (tmpcd[i] == 'e') S = 48;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 48:

if (tmpcd[i] == 't') S = 49;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 49:

if (tmpcd[i] == 'e') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 50:

if (tmpcd[i] == 'l') S = 51;

else if (tmpcd[i] == 'n') S = 53;

else if (tmpcd[i] == 'x') S = 55;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 51:

if (tmpcd[i] == 's') S = 52;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 52:

if (tmpcd[i] == 'e') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 53:

if (tmpcd[i] == 'u') S = 54;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 54:

if (tmpcd[i] == 'm') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 55:

if (tmpcd[i] == 'p') S = 56;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 56:

if (tmpcd[i] == 'o') S = 57;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 57:

if (tmpcd[i] == 'r') S = 58;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 58:

if (tmpcd[i] == 't') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 59:

if (tmpcd[i] == 'a') S = 60;

else if (tmpcd[i] == 'l') S = 63;

else if (tmpcd[i] == 'o') S = 66;

else if (tmpcd[i] == 'r') S = 102;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 60:

if (tmpcd[i] == 'l') S = 61;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 61:

if (tmpcd[i] == 's') S = 62;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 62:

if (tmpcd[i] == 'e') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 63:

if (tmpcd[i] == 'o') S = 64;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 64:

if (tmpcd[i] == 'a') S = 65;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 65:

if (tmpcd[i] == 't') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 66:

if (tmpcd[i] == 'r') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 67:

if (tmpcd[i] == 'f') S = 300;

else if (tmpcd[i] == 'n') S = 68;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 68:

if (tmpcd[i] == 't') S = 300;

else if (tmpcd[i] == 'l') S = 99;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 69:

if (tmpcd[i] == 'o') S = 70;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 70:

if (tmpcd[i] == 'o') S = 71;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 71:

if (tmpcd[i] == 'n') S = 72;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 72:

if (tmpcd[i] == 'g') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 73:

if (tmpcd[i] == 'a') S = 74;

else if (tmpcd[i] == 'e') S = 81;

else if (tmpcd[i] == 'o') S = 82;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 74:

if (tmpcd[i] == 'm') S = 75;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 75:

if (tmpcd[i] == 'e') S = 76;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 76:

if (tmpcd[i] == 's') S = 77;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 77:

if (tmpcd[i] == 'p') S = 78;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 78:

if (tmpcd[i] == 'a') S = 79;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 79:

if (tmpcd[i] == 'c') S = 80;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 80:

if (tmpcd[i] == 'e') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 81:

if (tmpcd[i] == 'w') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 82:

if (allchar(tmpcd[i + 1]) && tmpcd[i] == 't') S = 83;

else if (!allchar(tmpcd[i + 1]) && tmpcd[i] == 't') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 83:

if (tmpcd[i] == 'e') S = 84;

else if (tmpcd[i] == '\_') S = 89;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

case 84:

if (tmpcd[i] == 'x') S = 85;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 85:

if (tmpcd[i] == 'p') S = 86;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 86:

if (tmpcd[i] == 'e') S = 87;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 87:

if (tmpcd[i] == 'c') S = 88;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 88:

if (tmpcd[i] == 't') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 89:

if (tmpcd[i] == 'e') S = 90;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 90:

if (tmpcd[i] == 'q') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 91:

if (tmpcd[i] == 'p') S = 92;

else if (tmpcd[i] == 'r') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 92:

if (tmpcd[i] == 'e') S = 93;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 93:

if (tmpcd[i] == 'r') S = 94;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 94:

if (tmpcd[i] == 'a') S = 95;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 95:

if (tmpcd[i] == 't') S = 96;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 96:

if (tmpcd[i] == 'o') S = 97;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 98:

if (tmpcd[i] == 'r') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 99:

if (tmpcd[i] == 'i') S = 100;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 100:

if (tmpcd[i] == 'n') S = 101;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 101:

if (tmpcd[i] == 'e') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 102:

if (tmpcd[i] == 'i') S = 103;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 103:

if (tmpcd[i] == 'e') S = 104;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 104:

if (tmpcd[i] == 'n') S = 105;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 105:

if (tmpcd[i] == 'd') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 106:

if (tmpcd[i] == 'u') S = 107;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 107:

if (tmpcd[i] == 'l') S = 108;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 108:

if (tmpcd[i] == 'l') S = 109;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 109:

if (tmpcd[i] == 'p') S = 110;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 110:

if (tmpcd[i] == 't') S = 111;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 111:

if (tmpcd[i] == 'r') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 112:

if (tmpcd[i] == 'r') S = 113;

else if (tmpcd[i] == 'u') S = 124;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 113:

if (tmpcd[i] == 'i') S = 114;

else if (tmpcd[i] == 'o') S = 118;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 114:

if (tmpcd[i] == 'v') S = 115;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 115:

if (tmpcd[i] == 'a') S = 116;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 116:

if (tmpcd[i] == 't') S = 117;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 117:

if (tmpcd[i] == 'e') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 118:

if (tmpcd[i] == 't') S = 119;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 119:

if (tmpcd[i] == 'e') S = 120;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 120:

if (tmpcd[i] == 'c') S = 121;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 121:

if (tmpcd[i] == 't') S = 122;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 122:

if (tmpcd[i] == 'e') S = 123;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 123:

if (tmpcd[i] == 'd') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 124:

if (tmpcd[i] == 'b') S = 125;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 125:

if (tmpcd[i] == 'l') S = 126;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 126:

if (tmpcd[i] == 'i') S = 127;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 127:

if (tmpcd[i] == 'c') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 128:

if (tmpcd[i] == 'e') S = 129;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 129:

if (tmpcd[i] == 't') S = 130;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 130:

if (tmpcd[i] == 'u') S = 131;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 131:

if (tmpcd[i] == 'r') S = 132;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 132:

if (tmpcd[i] == 'n') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 133:

if (tmpcd[i] == 'o') S = 134;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 134:

if (tmpcd[i] == 't') S = 135;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 135:

if (tmpcd[i] == 'o') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 136:

if (tmpcd[i] == 'h') S = 137;

else if (tmpcd[i] == 'i') S = 140;

else if (tmpcd[i] == 't') S = 140;

else if (tmpcd[i] == 'w') S = 151;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 137:

if (tmpcd[i] == 'o') S = 138;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 138:

if (tmpcd[i] == 'r') S = 139;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 139:

if (tmpcd[i] == 't') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 140:

if (tmpcd[i] == 'z') S = 141;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 141:

if (tmpcd[i] == 'e') S = 142;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 142:

if (tmpcd[i] == 'o') S = 143;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 143:

if (tmpcd[i] == 'f') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 144:

if (tmpcd[i] == 'a') S = 145;

else if (tmpcd[i] == 'r') S = 148;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 145:

if (tmpcd[i] == 't') S = 146;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 146:

if (tmpcd[i] == 'i') S = 147;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 147:

if (tmpcd[i] == 'c') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 148:

if (tmpcd[i] == 'u') S = 149;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 149:

if (tmpcd[i] == 'c') S = 150;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 150:

if (tmpcd[i] == 't') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 151:

if (tmpcd[i] == 'i') S = 152;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 152:

if (tmpcd[i] == 't') S = 153;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 153:

if (tmpcd[i] == 'c') S = 154;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 154:

if (tmpcd[i] == 'h') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 155:

if (tmpcd[i] == 'e') S = 156;

else if (tmpcd[i] == 'h') S = 162;

else if (tmpcd[i] == 'r') S = 166;

else if (tmpcd[i] == 'y') S = 168;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 156:

if (tmpcd[i] == 'm') S = 157;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 157:

if (tmpcd[i] == 'p') S = 158;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 158:

if (tmpcd[i] == 'l') S = 159;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 159:

if (tmpcd[i] == 'a') S = 160;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 160:

if (tmpcd[i] == 't') S = 161;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 161:

if (tmpcd[i] == 'e') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 162:

if (tmpcd[i] == 'i') S = 163;

else if (tmpcd[i] == 'r') S = 164;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 163:

if (tmpcd[i] == 's') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 164:

if (tmpcd[i] == 'o') S = 165;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 165:

if (tmpcd[i] == 'w') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 166:

if (tmpcd[i] == 'u') S = 167;

else if (tmpcd[i] == 'y') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 167:

if (tmpcd[i] == 'e') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 168:

if (tmpcd[i] == 'p') S = 169;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 169:

if (tmpcd[i] == 'e') S = 170;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 170:

if (tmpcd[i] == 'n') S = 171;

else if (tmpcd[i] == 'i') S = 174;

else if (tmpcd[i] == 'd') S = 175;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 171:

if (tmpcd[i] == 'a') S = 172;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 172:

if (tmpcd[i] == 'm') S = 173;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 173:

if (tmpcd[i] == 'e') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 174:

if (tmpcd[i] == 'd') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 175:

if (tmpcd[i] == 'e') S = 175;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 176:

if (tmpcd[i] == 'f') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 177:

if (tmpcd[i] == 's') S = 178;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 178:

if (tmpcd[i] == 'i') S = 179;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 179:

if (tmpcd[i] == 'n') S = 180;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 180:

if (tmpcd[i] == 'g') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 181:

if (tmpcd[i] == 'i') S = 182;

else if (tmpcd[i] == 'o') S = 187;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 182:

if (tmpcd[i] == 'r') S = 183;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 183:

if (tmpcd[i] == 't') S = 184;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 184:

if (tmpcd[i] == 'u') S = 185;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 185:

if (tmpcd[i] == 'a') S = 186;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 186:

if (tmpcd[i] == 'l') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 187:

if (tmpcd[i] == 'i') S = 188;

else if (tmpcd[i] == 'l') S = 189;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 188:

if (tmpcd[i] == 'd') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 189:

if (tmpcd[i] == 'a') S = 190;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 190:

if (tmpcd[i] == 't') S = 191;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 191:

if (tmpcd[i] == 'i') S = 192;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 192:

if (tmpcd[i] == 'l') S = 193;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 193:

if (tmpcd[i] == 'e') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 194:

if (tmpcd[i] == 'h') S = 195;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 195:

if (tmpcd[i] == 'i') S = 196;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 196:

if (tmpcd[i] == 'l') S = 197;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 197:

if (tmpcd[i] == 'e') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 198:

if (tmpcd[i] == 'u') S = 199;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 199:

if (tmpcd[i] == 't') S = 200;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 200:

if (tmpcd[i] == 'a') S = 201;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 201:

if (tmpcd[i] == 'b') S = 202;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 202:

if (tmpcd[i] == 'l') S = 203;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 203:

if (tmpcd[i] == 'e') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 204:

if (tmpcd[i] == 'o') S = 205;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 205:

if (tmpcd[i] == 'r') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 206:

if (tmpcd[i] == ':') S = 352;

else S = 353;

break;

case 207:

if (tmpcd[i] == '=') S = 354;

else if (tmpcd[i] == '<') S = 354;

else S = 355;

break;

case 208:

if (tmpcd[i] == '=') S = 354;

else if (tmpcd[i] == '>') S = 354;

else S = 355;

break;

case 209:

if (tmpcd[i] == '=') S = 356;

else if (tmpcd[i] == '|') S = 354;

else S = 357;

break;

case 210:

if (tmpcd[i] == '=') S = 356;

else if (tmpcd[i] == '&') S = 354;

else S = 357;

break;

case 211:

if (tmpcd[i] == '=') S = 354;

else S = 357;

break;

case 212:

if (tmpcd[i] == '=') S = 354;

else S = 357;

break;

case 213:

if (tmpcd[i] == '+' || tmpcd[i] == '=') S = 356;

else S = 357;

break;

case 214:

if (isnum(tmpcd[i])) S = 1;

else if (tmpcd[i] == '-' || tmpcd[i] == '=') S = 356;

else S = 357;

break;

case 215:

if (tmpcd[i] == '=') S = 356;

else S = 357;

break;

case 216:

if (tmpcd[i] == '=') S = 356;

else S = 357;

break;

case 217:

if (tmpcd[i] == '=') S = 356;

else S = 357;

break;

case 218:

if (tmpcd[i] == 'e') S = 219;

else if (tmpcd[i] == 'o') S = 224;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 219:

if (tmpcd[i] == 'l') S = 220;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 221:

if (tmpcd[i] == 'e') S = 222;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 222:

if (tmpcd[i] == 't') S = 223;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;\

case 223:

if (tmpcd[i] == 'e') S = 300;

else if (stop\_sing(tmpcd[i])) S = 0;

else if (allchar(tmpcd[i])) S = 350;

else S = 400;

break;

case 224:

if (allchar(tmpcd[i])) S = 350;

else if (stop\_sing(tmpcd[i])) S = 300;

else S = 400;

break;

case 350:

if (isnum(tmpcd[i]) || tmpcd[i] == '0' || tmpcd[i] == '.') S = 350;

else if (allchar(tmpcd[i])) S = 350;

else if (stop\_sing(tmpcd[i])) S = 0;

else S = 400;

break;

case 351:

if (tmpcd[i] == '"' || tmpcd[i] == Convert::ToChar("'") ) { tmpwd = ""; S = 0; }

break;

case 399:

if (tmpcd[i] == Convert::ToChar("'")) { tmpwd = ""; S = 0; }

break;

case 400:

//error

MessageBox::Show(L"Ошибка", L"Программа остановлена", MessageBoxButtons::OK, MessageBoxIcon::Error);

S = 0;

break;

}

if (tmpcd[i] == '\n') line++;

if (S != 0 && S != 400 && S != 301 && S != 353 && S != 355 && S != 357) tmpwd += tmpcd[i];

if (S == 353 || S == 355 || S == 357 || S == 301) i--;

if (S == 0 && tmpwd != "") {

S = 0;

add\_idet(tmpwd, idet\_rows);

tmpwd = "";

i--;

}

else if (S == 300) {

S = 0;

add\_lex(tmpwd, lex\_rows);

tmpwd = "";

}

else if (S == 301) {

S = 0;

add\_cnst(tmpwd, cnst\_rows);

tmpwd = "";

}

else if (S == 356 || S == 357) {

S = 0;

add\_op\_s(tmpwd, op\_rows);

tmpwd = "";

}

else if (S == 354 || S == 355) {

S = 0;

add\_comp(tmpwd, comp\_rows);

tmpwd = "";

}

else if (S == 352 || S == 353) {

S = 0;

add\_separ(tmpwd, sep\_rows);

tmpwd = "";

}

}

}

void descriptor(int lex\_rows, int cnst\_rows, int idet\_rows, int op\_rows, int comp\_rows, int sep\_rows)

{

String^ tmpcd = richTextBox2->Text;

String^ tmpwd = "";

for (int i = 0; i < tmpcd->Length; i++) {

if (!skip\_sings(tmpcd[i]) && stop\_sing(tmpcd[i])) {

if (op\_sings(tmpcd[i])) {

tmpwd += tmpcd[i++];

if (i < tmpcd->Length && op\_sings(tmpcd[i])) tmpwd += tmpcd[i];

else i--;

}

else if (op\_comp(tmpcd[i])) {

tmpwd += tmpcd[i++];

if (i < tmpcd->Length && op\_comp(tmpcd[i])) tmpwd += tmpcd[i];

else i--;

}

else if (spec\_chr(tmpcd[i])) {

tmpwd += tmpcd[i++];

if (i < tmpcd->Length && tmpcd[i] == ':') {

tmpwd += tmpcd[i];

}

else i--;

}

}

else if (tmpcd[i] == '"') {

i++;

while (i < tmpcd->Length && tmpcd[i] != '"') tmpwd += tmpcd[i++];

tmpwd = "";

}

else if (isnum(tmpcd[i]) || tmpcd[i] == '0') {

while (i < tmpcd->Length && (isnum(tmpcd[i]) || tmpcd[i] == '0' || tmpcd[i] == '.')) tmpwd += tmpcd[i++];

i--;

}

else if (allchar(tmpcd[i])) {

while(i < tmpcd->Length && allchar(tmpcd[i])) tmpwd += tmpcd[i++];

i--;

}

if (tmpwd != "") {

for (int j = 0; j < idet\_rows - 1; j++) {

if (Convert::ToString(t\_id->Rows[j]->Cells[1]->Value) == tmpwd) {

tmpwd = "(" + Cidet1->HeaderCell->Value + "," + (j + 1) + ")";

break;

}

}

for (int j = 0; j < lex\_rows - 1; j++) {

if (Convert::ToString(t\_lexme->Rows[j]->Cells[1]->Value) == tmpwd) {

tmpwd = "(" + Clex1->HeaderCell->Value + "," + (j + 1) + ")";

break;

}

}

for (int j = 0; j < cnst\_rows - 1; j++) {

if (Convert::ToString(t\_cnst->Rows[j]->Cells[1]->Value) == tmpwd) {

tmpwd = "(" + Ccnst1->HeaderCell->Value + "," + (j + 1) + ")";

break;

}

}

for (int j = 0; j < op\_rows - 1; j++) {

if (Convert::ToString(t\_op\_sings->Rows[j]->Cells[1]->Value) == tmpwd) {

tmpwd = "(" + dataGridViewTextBoxColumn1->HeaderCell->Value + "," + (j + 1) + ")";

break;

}

}

for (int j = 0; j < comp\_rows - 1; j++) {

if (Convert::ToString(t\_op\_comp->Rows[j]->Cells[1]->Value) == tmpwd) {

tmpwd = "(" + dataGridViewTextBoxColumn3->HeaderCell->Value + "," + (j + 1) + ")";

break;

}

}

for (int j = 0; j < sep\_rows - 1; j++) {

if (Convert::ToString(t\_sep->Rows[j]->Cells[1]->Value) == tmpwd) {

tmpwd = "(" + dataGridViewTextBoxColumn5->HeaderCell->Value + "," + (j + 1) + ")";

break;

}

}

desTextBox4->Text += tmpwd + " ";

tmpwd = "";

}

}

}

void psevda()

{

String^ tmpcd = desTextBox4->Text;

String^ tmpwd = "";

for (int i = 0; i < tmpcd->Length; i++) {

if (tmpcd[i] == '(') {

i++;

while (i < tmpcd->Length && tmpcd[i] != ',') tmpwd += tmpcd[i++];

i++;

if (tmpwd == Convert::ToString(Cidet1->HeaderCell->Value)) {

tmpwd = "";

while (tmpcd[i] != ')') tmpwd += tmpcd[i++];

i++;

tmpwd = Convert::ToString(t\_id->Rows[Convert::ToInt32(tmpwd) - 1]->Cells[2]->Value);

}

if (tmpwd == Convert::ToString(Clex1->HeaderCell->Value)) {

tmpwd = "";

while (tmpcd[i] != ')') tmpwd += tmpcd[i++];

i++;

tmpwd = Convert::ToString(t\_lexme->Rows[Convert::ToInt32(tmpwd) - 1]->Cells[2]->Value);

}

if (tmpwd == Convert::ToString(Ccnst1->HeaderCell->Value)) {

tmpwd = "";

while (tmpcd[i] != ')') tmpwd += tmpcd[i++];

i++;

tmpwd = Convert::ToString(t\_cnst->Rows[Convert::ToInt32(tmpwd) - 1]->Cells[2]->Value);

}

if (tmpwd == Convert::ToString(dataGridViewTextBoxColumn1->HeaderCell->Value)) {

tmpwd = "";

while (tmpcd[i] != ')') tmpwd += tmpcd[i++];

i++;

tmpwd = Convert::ToString(t\_op\_sings->Rows[Convert::ToInt32(tmpwd) - 1]->Cells[2]->Value);

}

if (tmpwd == Convert::ToString(dataGridViewTextBoxColumn3->HeaderCell->Value)) {

tmpwd = "";

while (tmpcd[i] != ')') tmpwd += tmpcd[i++];

i++;

tmpwd = Convert::ToString(t\_op\_comp->Rows[Convert::ToInt32(tmpwd) - 1]->Cells[2]->Value);

}

if (tmpwd == Convert::ToString(dataGridViewTextBoxColumn5->HeaderCell->Value)) {

tmpwd = "";

while (tmpcd[i] != ')') tmpwd += tmpcd[i++];

i++;

tmpwd = Convert::ToString(t\_sep->Rows[Convert::ToInt32(tmpwd) - 1]->Cells[2]->Value);

}

}

pseTextBox3->Text += tmpwd + " ";

tmpwd = "";

}

}

void cl()

{

richTextBox1->Clear();

richTextBox2->Clear();

pseTextBox3->Clear();

desTextBox4->Clear();

t\_lexme->RowCount = 1;

t\_cnst->RowCount = 1;

t\_id->RowCount = 1;

t\_op\_sings->RowCount = 1;

t\_sep->RowCount = 1;

t\_op\_comp->RowCount = 1;

S = 0;

a = 0;

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e)

{

prework();

int lex\_row = 1, cnst\_row = 1, idet\_row = 1, op\_row = 1, comp\_row = 1, sep\_row = 1;

// work(lex\_row, cnst\_row, idet\_row, op\_row, comp\_row, sep\_row);

// descriptor(lex\_row, cnst\_row, idet\_row, op\_row, comp\_row, sep\_row);

// psevda();

};

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e)

{

cl();

}

};

}