**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Основы информатики»

Отчет по лабораторной работе №9

«Ввод, сортировка и двоичный поиск в массиве структур»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-11 |  | преподаватель каф. ИУ5 |
| Михалёв Ярослав |  | Козлов А.Д. |
| Подпись и дата: |  | Подпись и дата: |
|  |  |  |

Москва, 2021 г.

Постановка задачи

**Задание.**

Англо-русский словарь построен в виде массива структур Dictionary. Структура содержит английское слово и соответствующее ему русское слово. Максимальный размер словаря – 100 пар слов.

Разработать программу, которая:

* обеспечивает формирование словаря (добавление и удаление записей);
* записывает словарь, отсортированный по английским значениям слов, в файл;
* обеспечивает просмотр словаря;
* выполняет перевод слов с английского на русский, используя для поиска слова в словаре метод двоичного поиска в отсортированном массиве;
* выполняет перевод слов с русского на английский, используя для поиска слов в словаре метод перебора;

Программа должна обеспечивать диалог с помощью меню.

Начальное число слов в словаре равно 10.

Для исключения проблем, связанных с вводом кириллицы, вводите русские слова латинскими буквами, например: kot (cat), sobaka(dog) и т.п.

Разработка алгоритма

**Описание входных, выходных и вспомогательных данных:**

Входные данные:

Нет

Выходные данные:

dict.txt – текстовый файл со словарем

Вспомогательные данные:

const int max\_size = 100 – размер массива dict

const int l\_word = 31 – длинна одного слова

**Список функций:**

int Menu()

Создаёт единичную матрицу размером N - 1

Выходные данные:

* int choose – целое число от 1 до 7

bool Equals(char word1[], char word2[])

Возвращает true, если слова совпадают, false в обратном случае

Входные данные:

* char word1[] – первое слово
* char word2[] – второе слово

Выходные данные:

* true / false

bool Compare(char word1[], char word2[])

Сравнивает два слова (посимвольно)

Входные данные:

* char word1[] – первое слово
* char word2[] – второе слово

Выходные данные:

* true – если word1[] > word2[]
* false – в обратном случае

void SortDict()

Сортирует словарь по возрастанию методом пузырька

int Find(char word[], int mode = 2)

Ищет слово в словаре

* английское, если mode = 0,
* русское, если mode = 1,
* любое, если mode = 2

Входные данные:

* char word[] – слово
* int mode – режим работы функции

Выходные данные:

* int i – индекс строки c найденным слово
* -1 – если слова нет в словаре

void Translate(int mode)

Переводит слово

Входные данные

* char word[] – слово
* int mode - режим работы программы

Выходные данные

* Переведённое слово

void AddWord()

Добавляет слово в словарь

Входные данные

* char word[] – слово
* char translate[] – перевод слова на английский

void AddTestWord(const char word[], const char translate[])

Входные данные

* const char word[] – слово
* const char translate[] – перевод слова на английский

void RemoveWord()

Удаляет слово из словаря

Входные данные

* char word[] – слово

void ShowDict()

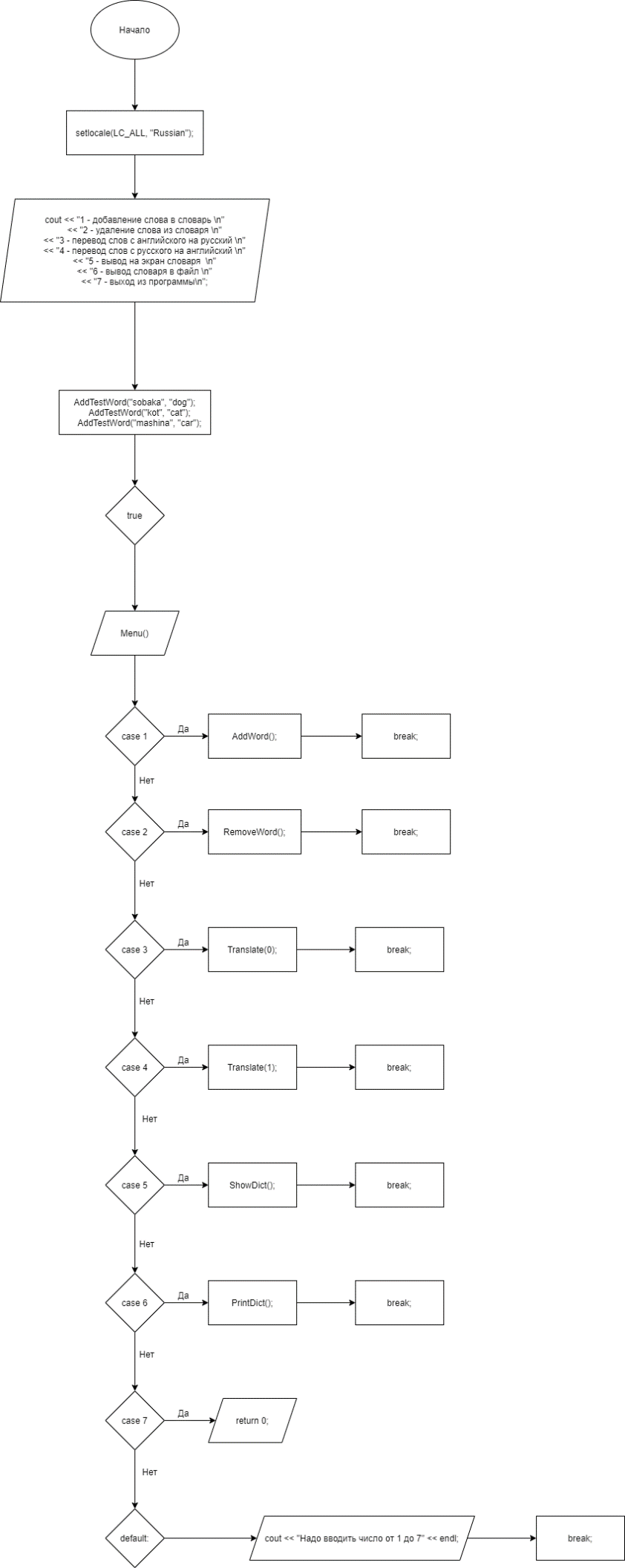
Выводит словарь на экран

void PrintDict()

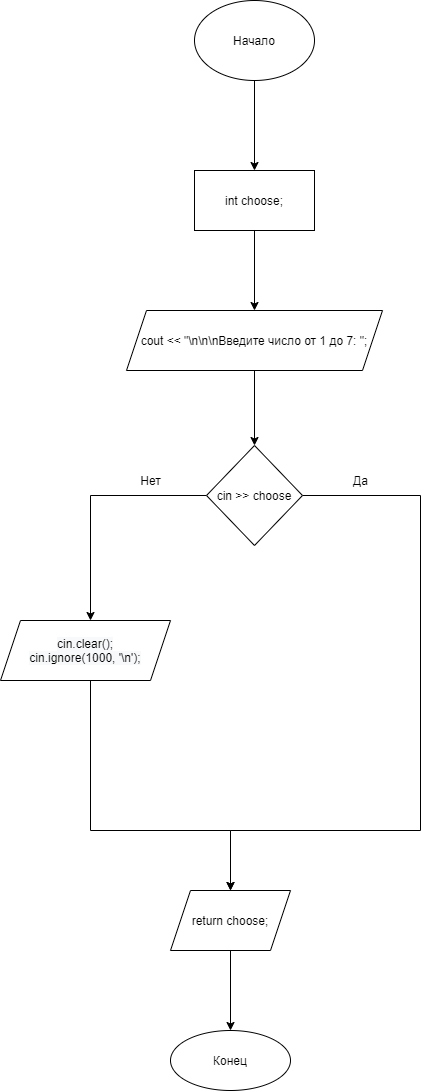
Печатает словарь в файл

Схема алгоритма

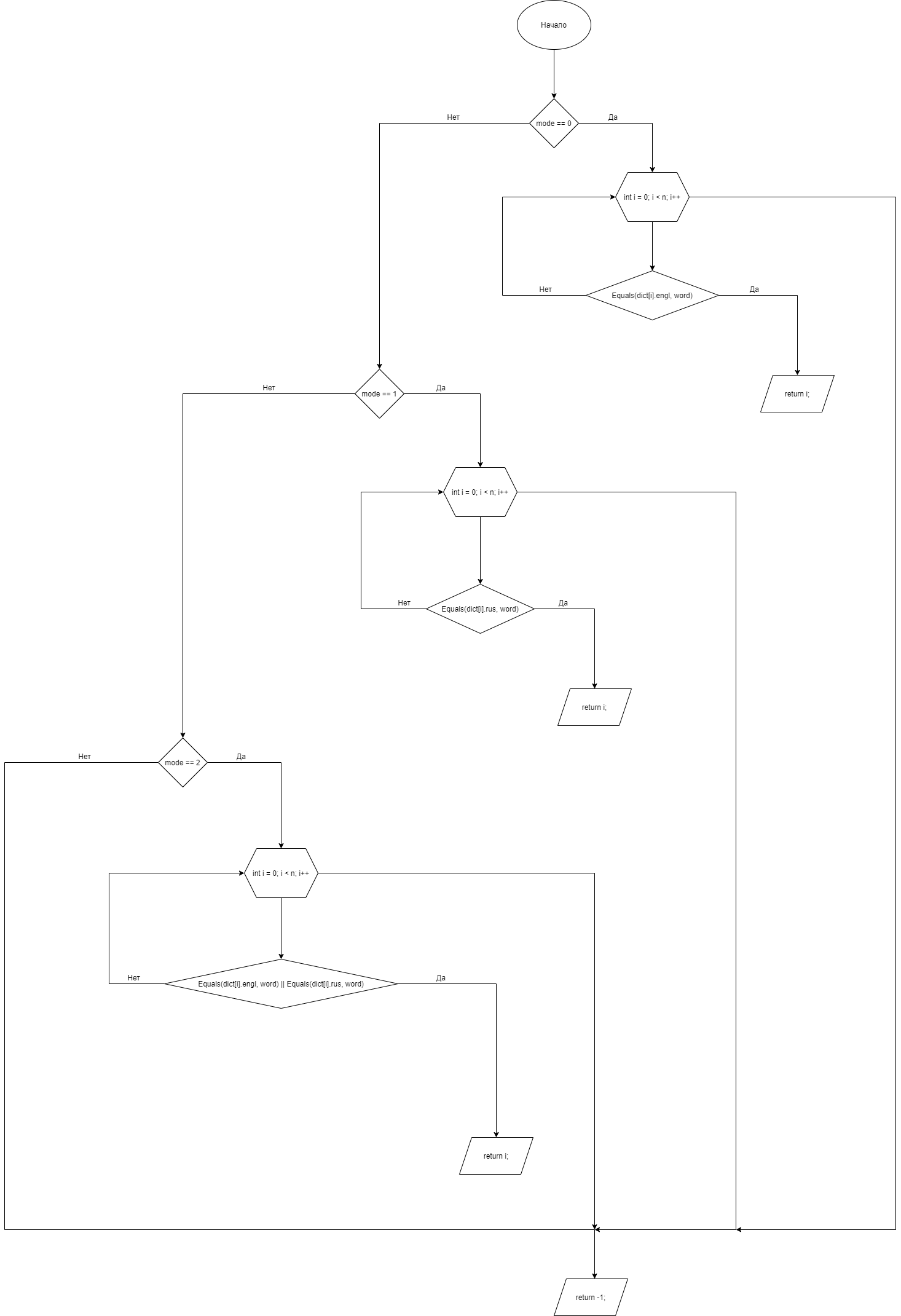
Main



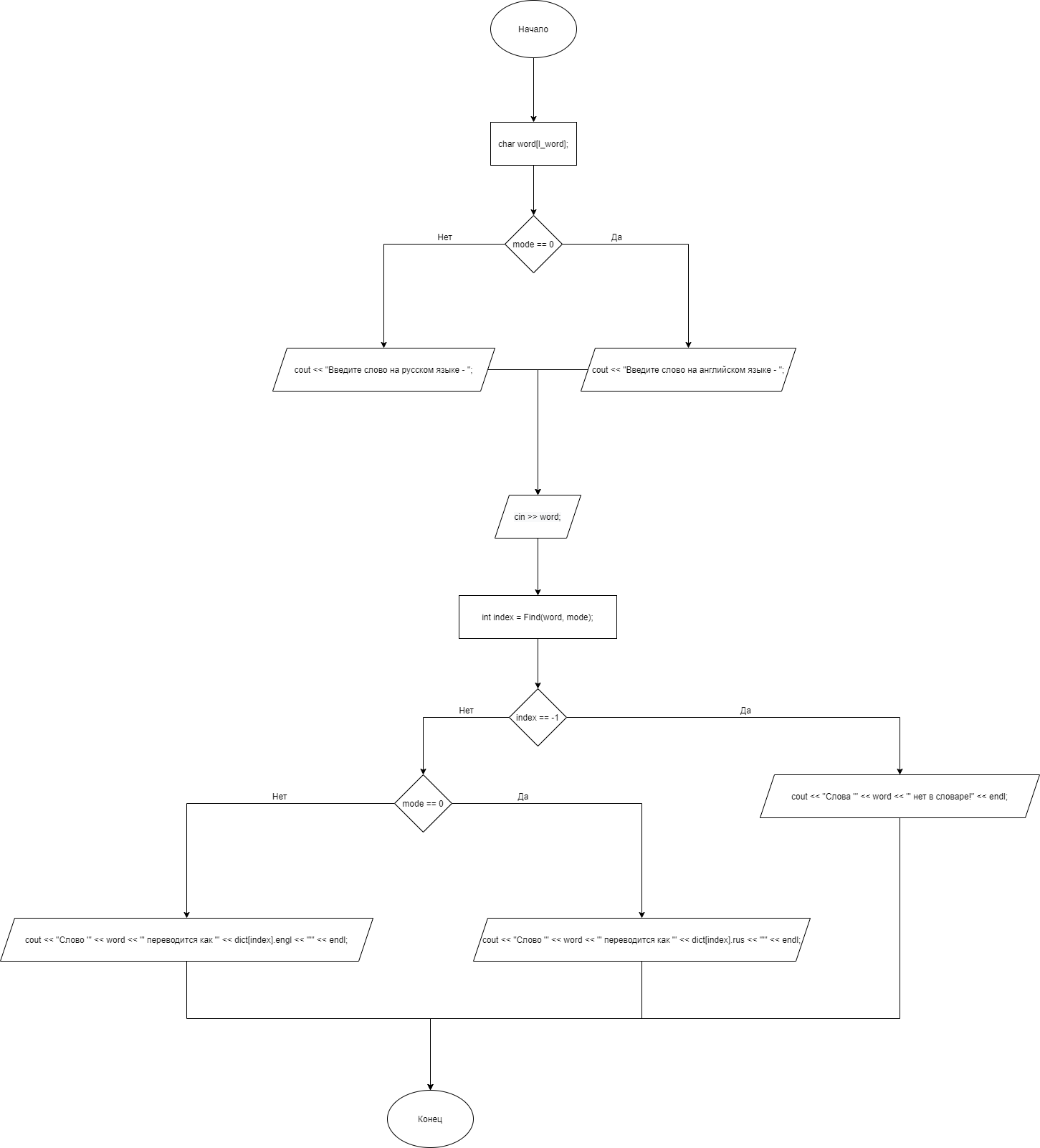
Menu



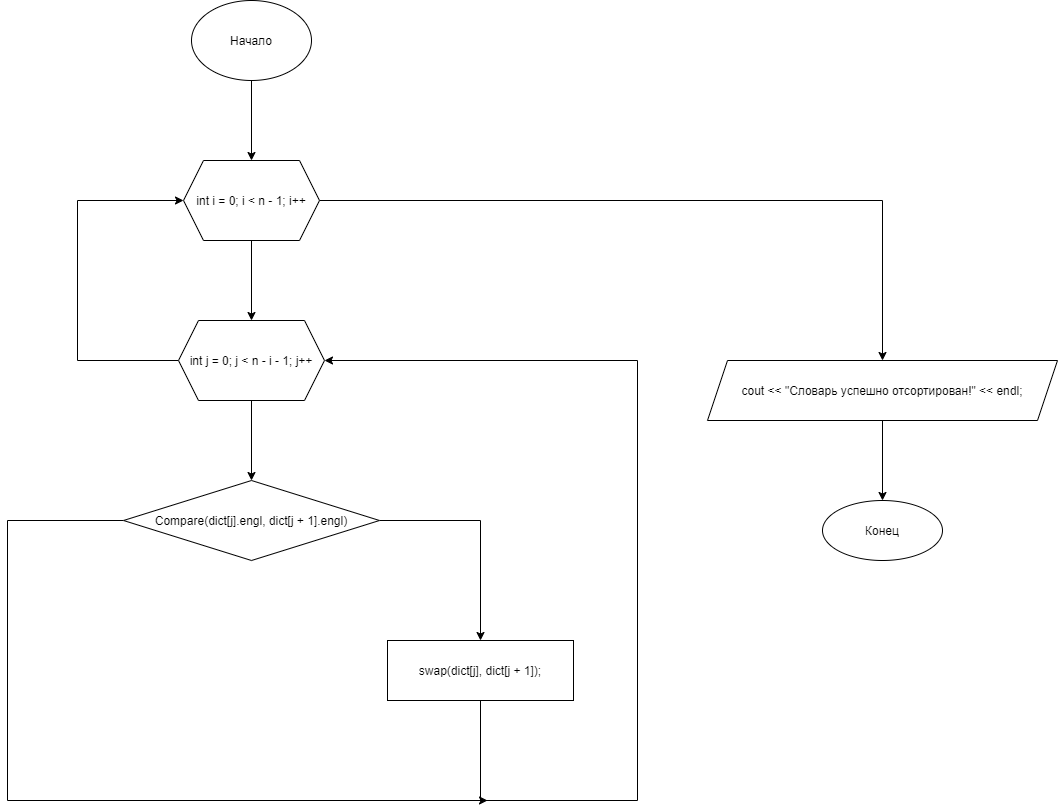
Find



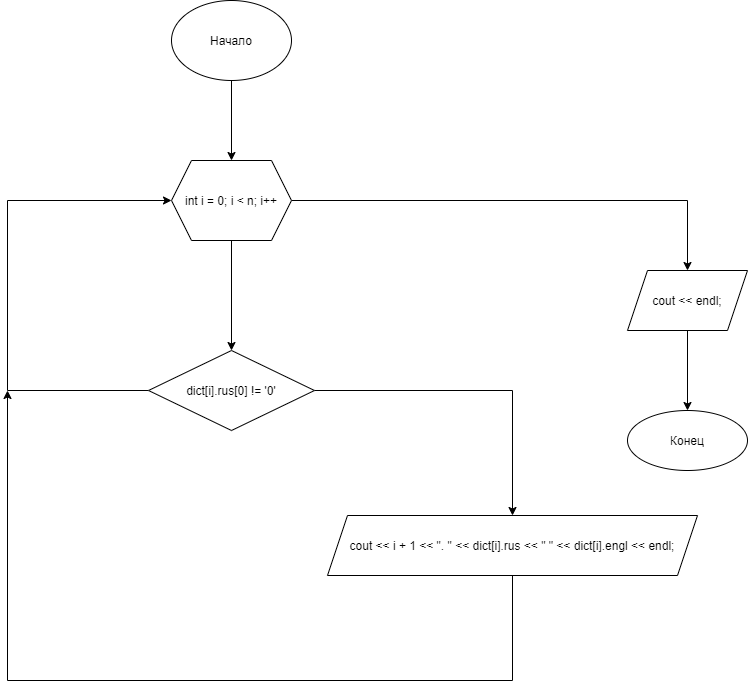
Translate



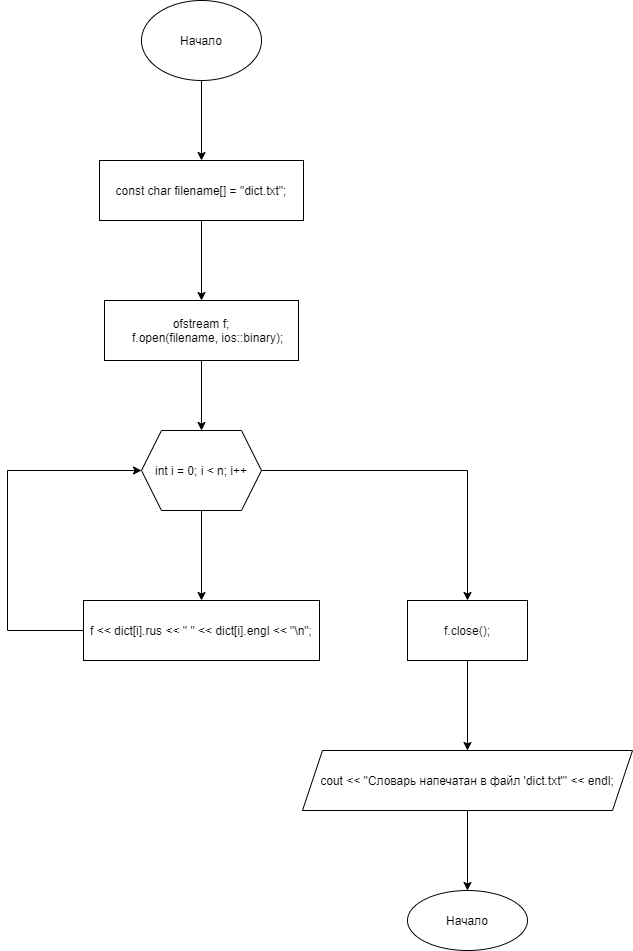
SortDict



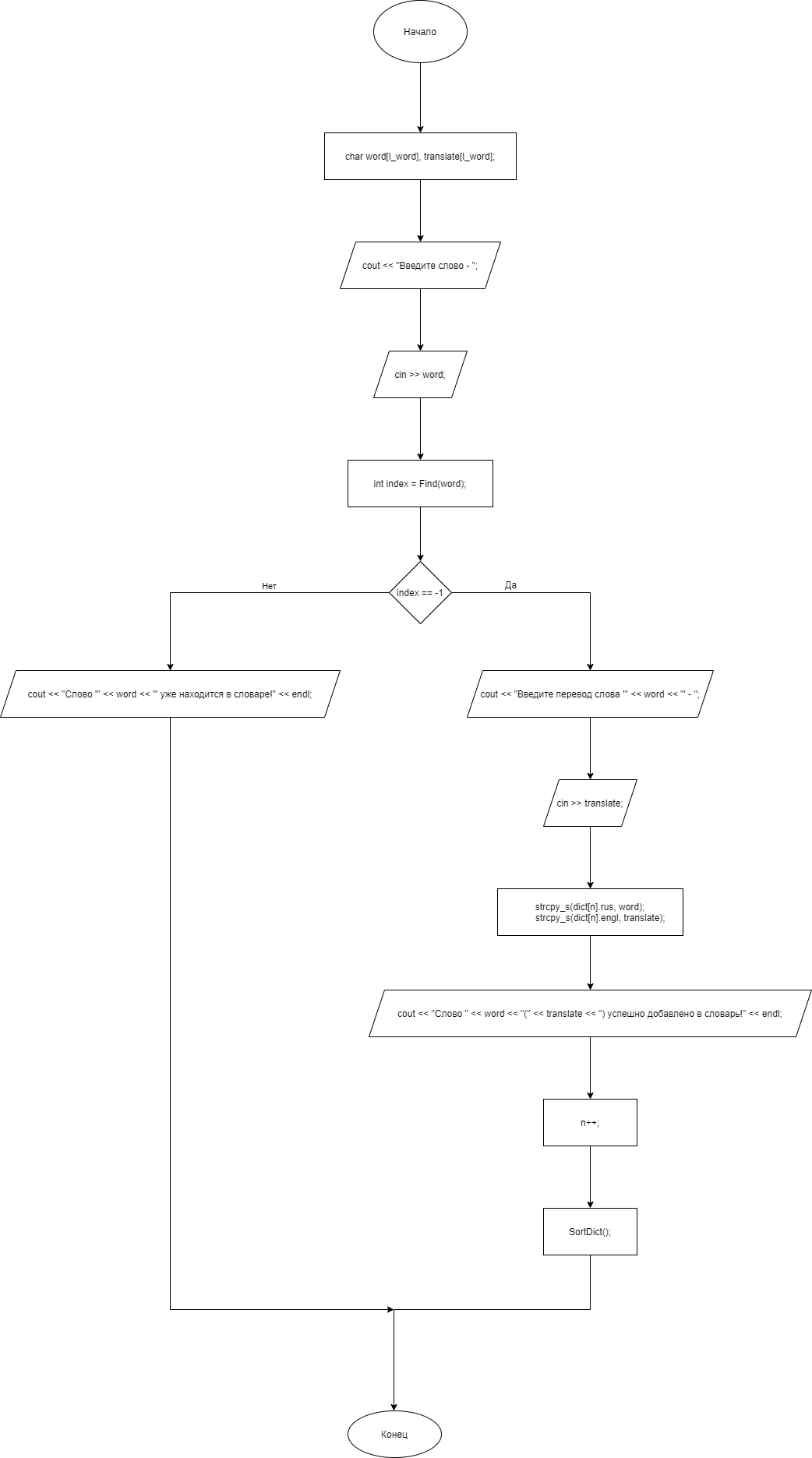
ShowDict



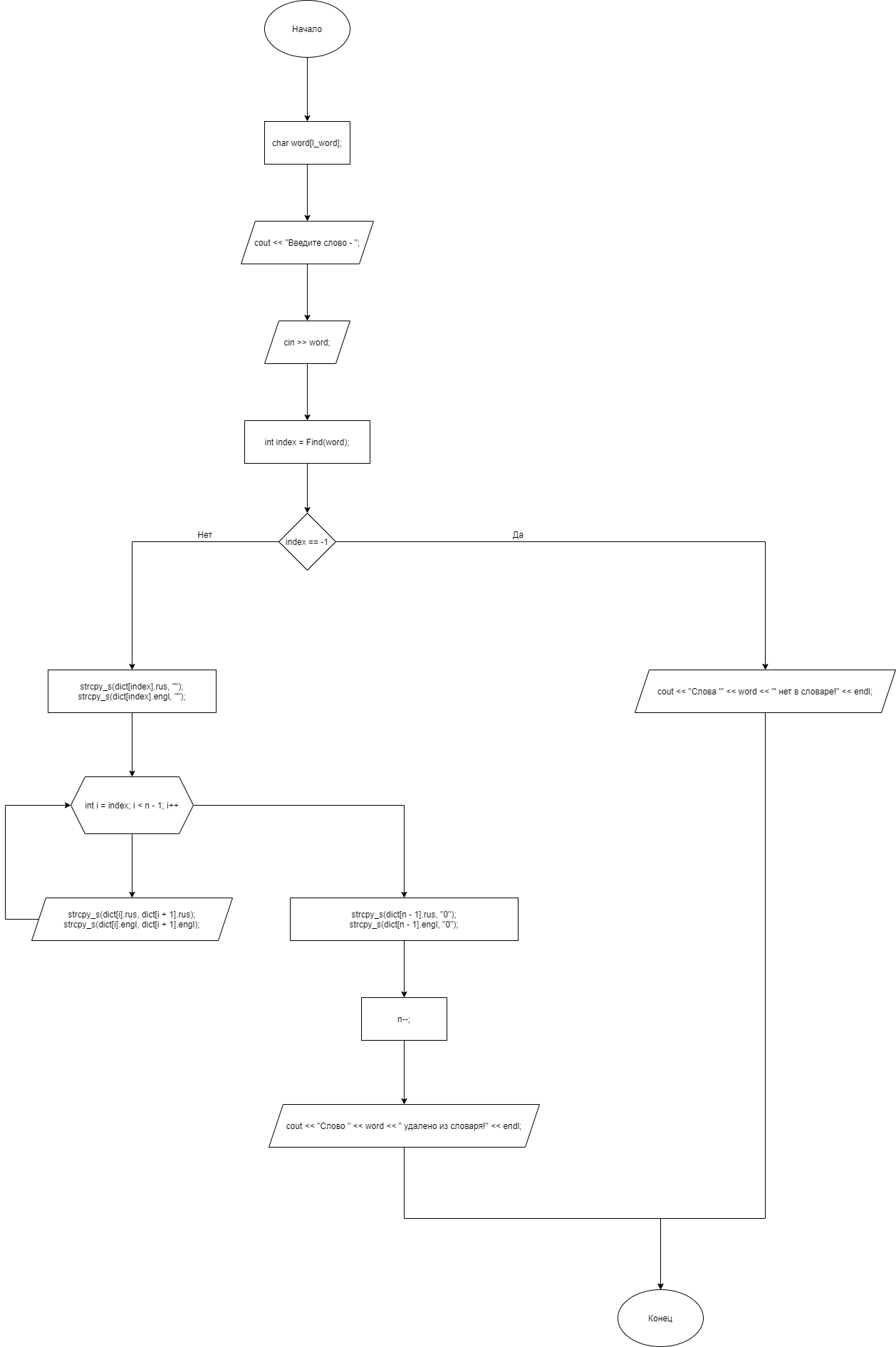
PrintDict



AddWord



RemoveWord



Текст программы

Main.cpp

#include <iostream>

#include <fstream>

**using** **namespace** std**;**

const int max\_size **=** 100**;**

const int l\_word **=** 31**;**

struct Dictionary **{**

char engl**[**l\_word**];** // слово по-английски

char rus**[**l\_word**];** // слово по-русски

**};**

Dictionary dict**[**max\_size**];** // массив структур для хранения словаря в оперативной памяти

int n **=** 0**;** // фактическое число записей в словаре

// Эта функция (в качестве подсказки) выводит пронумерованный перечень возможных

// действий пользователя и выполняет ввод номера выбранного действия.

// Это число она должна вернуть в вызвавшую функцию.

int Menu**()**

**{**

int choose**;**

cout **<<** "\n\n\nВведите число от 1 до 7: "**;**

**if** **(**cin **>>** choose**)**

**{**

**}**

**else**

**{**

cin**.**clear**();** // put the stream back into a good state

cin**.**ignore**(**1000**,** '\n'**);** // and remove the bad input remaining in the input buffer

**}**

**return** choose**;**

**}**

// Возвращает true, если слова совпадают, false в обратном случае

bool Equals**(**char word1**[],** char word2**[])**

**{**

**for** **(**int i **=** 0**;** i **<** l\_word**;** i**++)**

**{**

**if** **(**word1**[**i**]** **!=** word2**[**i**])**

**return** **false;**

**if** **(**word1**[**i**]** **==** '\0' **&&** word2**[**i**]** **==** '\0'**)**

**return** **true;**

**}**

return true;

}

// Сравнивает два слова (посимвольно)

bool Compare(char word1[], char word2[])

{

for (int i = 0; word1[i] != '\0' || word2[i] != '\0'; i++)

if (word1[i] > word2[i])

return true;

else if (word1[i] < word2[i])

return false;

}

// Сортирует словарь по возрастанию методом пузырька

void SortDict()

{

for (int i = 0; i < n - 1; i++)

for (int j = 0; j < n - i - 1; j++)

if (Compare(dict[j].engl, dict[j + 1].engl))

swap(dict[j], dict[j + 1]);

cout << "Словарь успешно отсортирован!" << endl;

}

// Ищет слово в словаре

int Find(char word[], int mode = 2)

{

if (mode == 0)

{

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

if (Equals(dict[i].engl, word))

return i;

}

else if (mode == 1)

{

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

if (Equals(dict[i].rus, word))

return i;

}

else if (mode == 2)

{

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

if (Equals(dict[i].engl, word) || Equals(dict[i].rus, word))

return i;

}

return -1;

}

// Переводит слово

void Translate(int mode)

{

char word[l\_word];

if (mode == 0)

cout << "Введите слово на английском языке - ";

else

cout << "Введите слово на русском языке - ";

cin >> word;

int index = Find(word, mode);

if (index == -1)

cout << "Слова '" << word << "' нет в словаре!" << endl;

else

{

if (mode == 0)

cout << "Слово '" << word << "' переводится как '" << dict[index].rus << "'" << endl;

else

cout << "Слово '" << word << "' переводится как '" << dict[index].engl << "'" << endl;

}

}

// Добавляет слово в словарь

void AddWord()

{

char word[l\_word], translate[l\_word];

cout << "Введите слово - ";

cin >> word;

int index = Find(word);

if (index == -1)

{

cout << "Введите перевод слова '" << word << "' - ";

cin >> translate;

strcpy\_s(dict[n].rus, word);

strcpy\_s(dict[n].engl, translate);

cout << "Слово " << word << "(" << translate << ") успешно добавлено в словарь!" << endl;

n++;

SortDict();

}

else

{

cout << "Слово '" << word << "' уже находится в словаре!" << endl;

}

}

// Добавляет слово в словарь

void AddTestWord(const char word[], const char translate[])

{

strcpy\_s(dict[n].rus, word);

strcpy\_s(dict[n].engl, translate);

n++;

}

// Удаляет слово из словаря

void RemoveWord()

{

char word[l\_word];

cout << "Введите слово - ";

cin >> word;

int index = Find(word);

if (index == -1)

cout << "Слова '" << word << "' нет в словаре!" << endl;

else

{

strcpy\_s(dict[index].rus, "");

strcpy\_s(dict[index].engl, "");

for (int i = index; i < n - 1; i++)

{

strcpy\_s(dict[i].rus, dict[i + 1].rus);

strcpy\_s(dict[i].engl, dict[i + 1].engl);

}

strcpy\_s(dict[n - 1].rus, "0");

strcpy\_s(dict[n - 1].engl, "0");

n--;

cout << "Cлово " << word << " удалено из словаря!" << endl;

}

}

// Выводит словарь на экран

void ShowDict()

{

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

if (dict[i].rus[0] != '0')

cout << i + 1 << ". " << dict[i].rus << " " << dict[i].engl << endl;

cout << endl;

}

// Печатает словарь в файл

void PrintDict()

{

const char filename[] = "dict.txt";

ofstream f;

f.open(filename, ios::binary);

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

f << dict[i].rus << " " << dict[i].engl << "\n";

f.close();

cout << "Словарь напечатан в файл 'dict.txt'" << endl;

}

int main()

{

setlocale(LC\_ALL, "Russian");

cout << "1 - добавление слова в словарь \n"

<< "2 - удаление слова из словаря \n"

<< "3 - перевод слов с английского на русский \n"

<< "4 - перевод слов с русского на английский \n"

<< "5 - вывод на экран словаря \n"

<< "6 - вывод словаря в файл \n"

<< "7 - выход из программы\n";

AddTestWord("sobaka", "dog");

AddTestWord("kot", "cat");

AddTestWord("mashina", "car");

while (true) {

switch (Menu()) {

case 1: AddWord(); break;

case 2: RemoveWord(); break;

case 3: Translate(0); break;

case 4: Translate(1); break;

case 5: ShowDict(); break;

case 6: PrintDict(); break;

case 7: return 0;

default:

cout << "Надо вводить число от 1 до 7" << endl;

break;

}

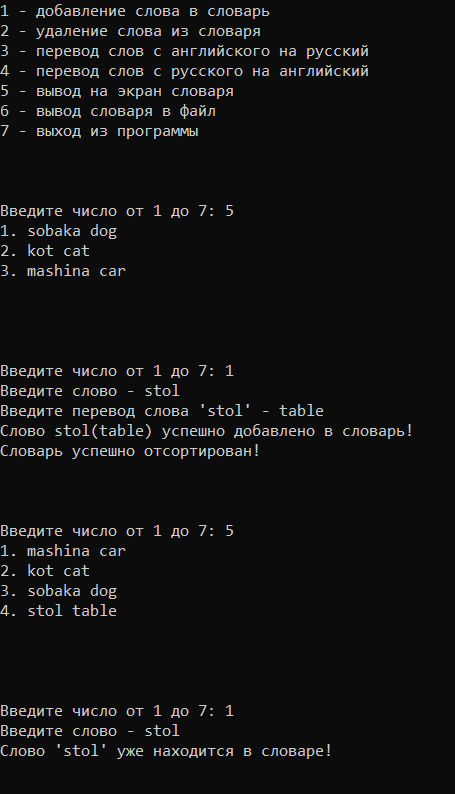
}

return 0;

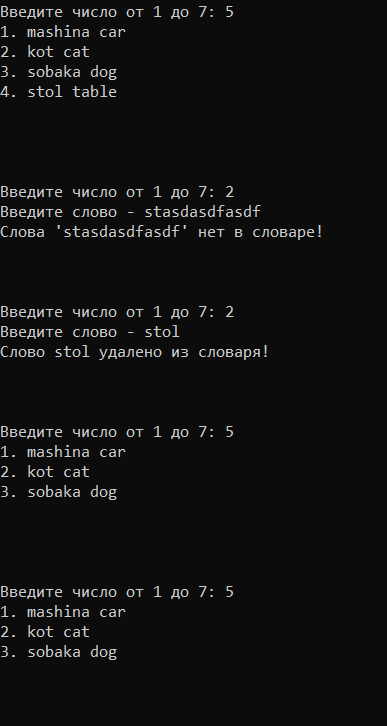
}

Анализ результатов

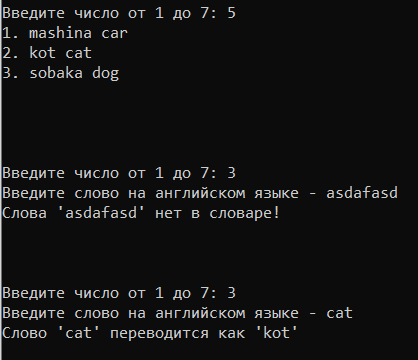
Добавление новых слов



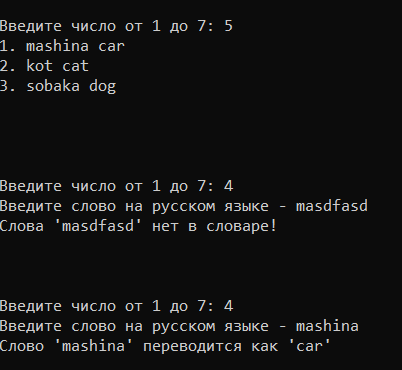
Удаление слов



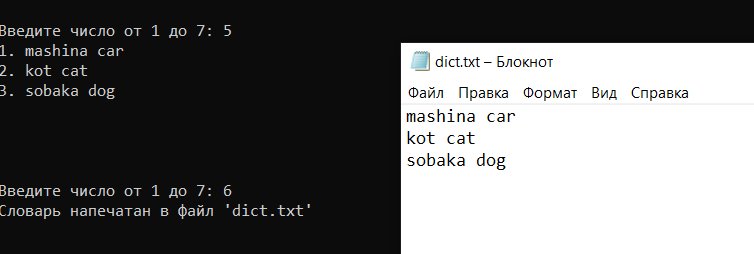
Перевод с английского на русский



Перевод с русского на английский



Печать словаря в файл



Вывод

Я научился

* Работать со структурами
* Работать с оператором switch case
* Записывать данные в текстовые файлы