Министерство образования Республики Беларусь

Учреждение образования

"Брестский государственный технический университет"

Кафедра ИИТ

Лабораторная работа №9

За 2ой семестр

По дисциплине: "ОАиП"

Тема: « Структуры, перечисления, объединения »

Выполнил:

Студенты 1-го курса АС-59

(подгруппа 2)

Левоцкий Н.

Цыганов Е.

Проверила:

Гирель Т. Н.

Брест 2021

**Цель работы:** изучить синтаксис и правила работы со структурами, реализовать программу с применением структур, перечислений и объединений.

**Вариант 12**

**Задание:**

Создать тип структуры согласно варианту, организовать поля этой структуры так, чтобы они содержали объединение, перечисление (можно добавить дополнительные поля) и битовое поле.

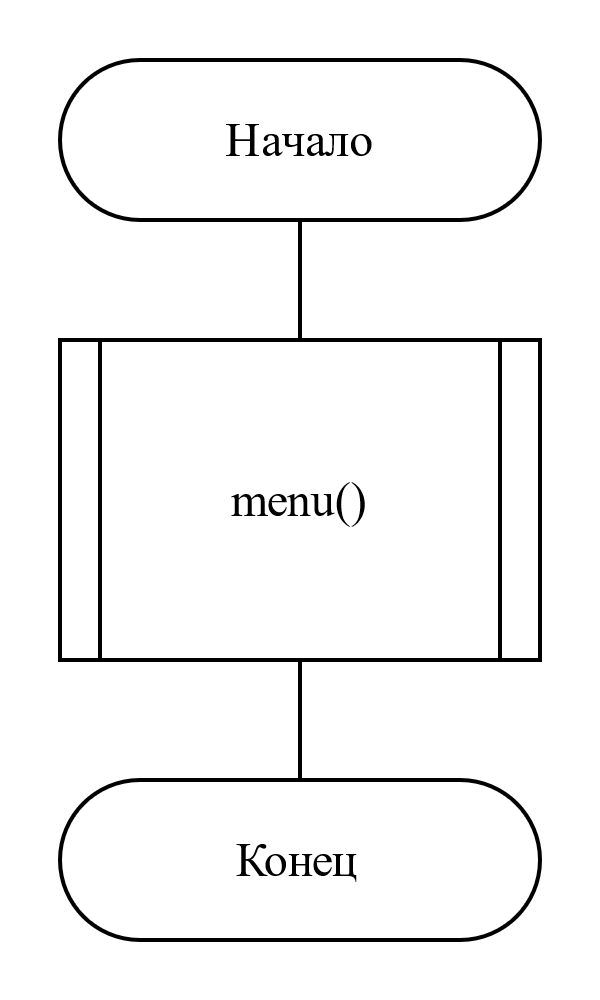
Создать массив структур, содержащий информацию согласно варианту индивидуального задания.

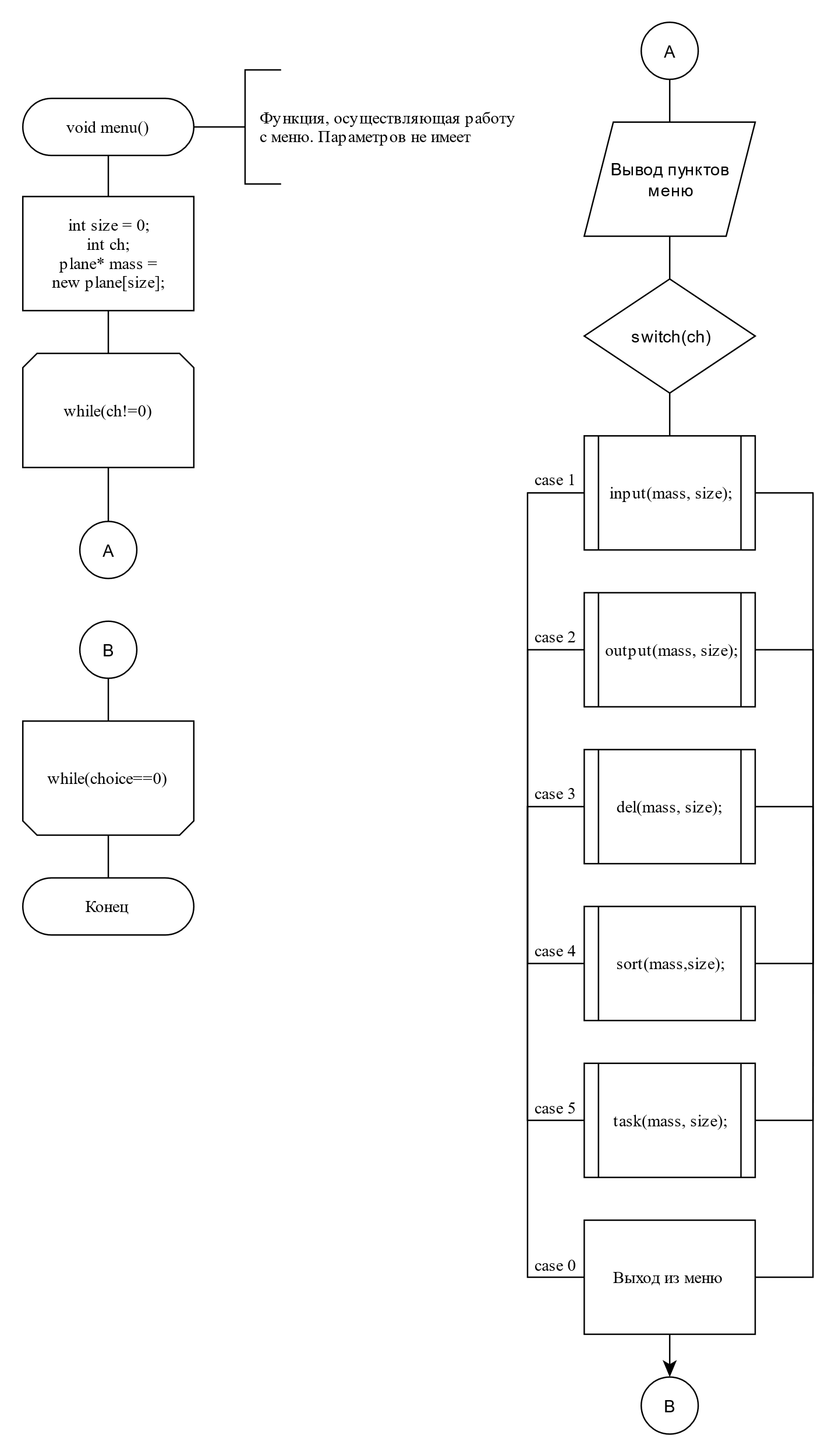
Реализовать работу с массивом структур через меню: ввод данных в массив, вывод содержимого массива на экран, сортировка по одному полю, удаления записи по заданному значению поля, выборка записей согласно индивидуального задания.

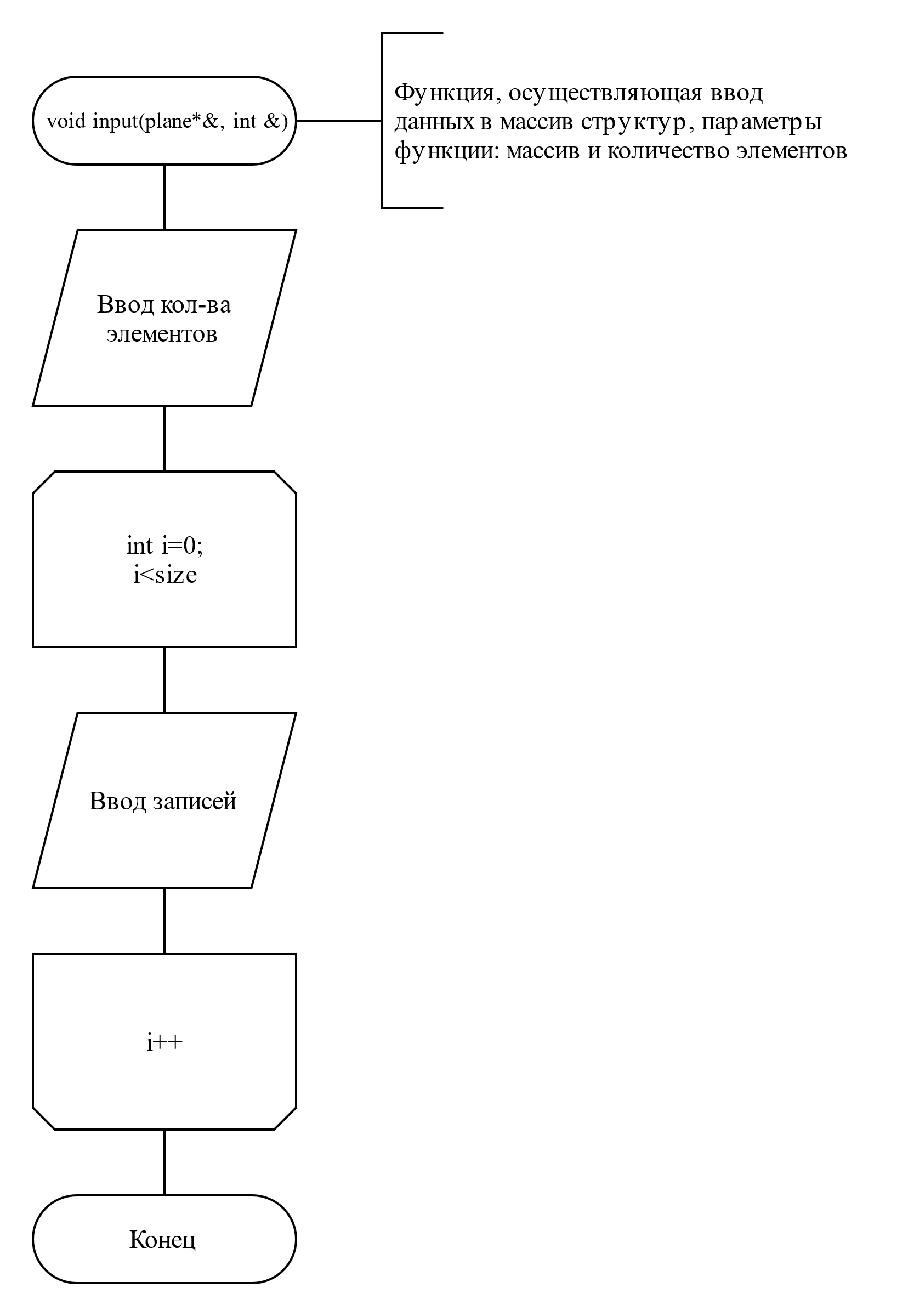
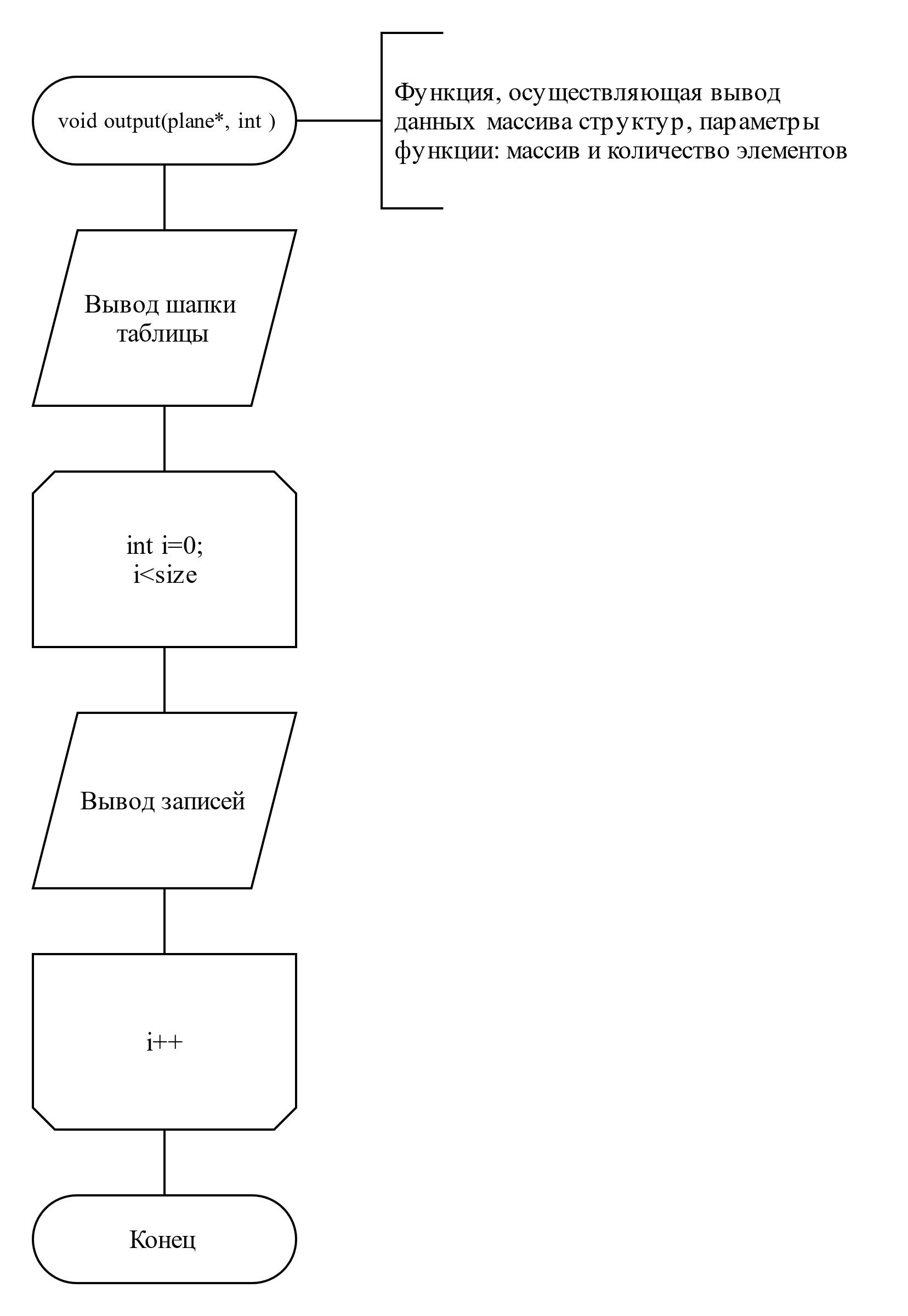
В справочной аэропорта имеется расписание вылета самолетов.

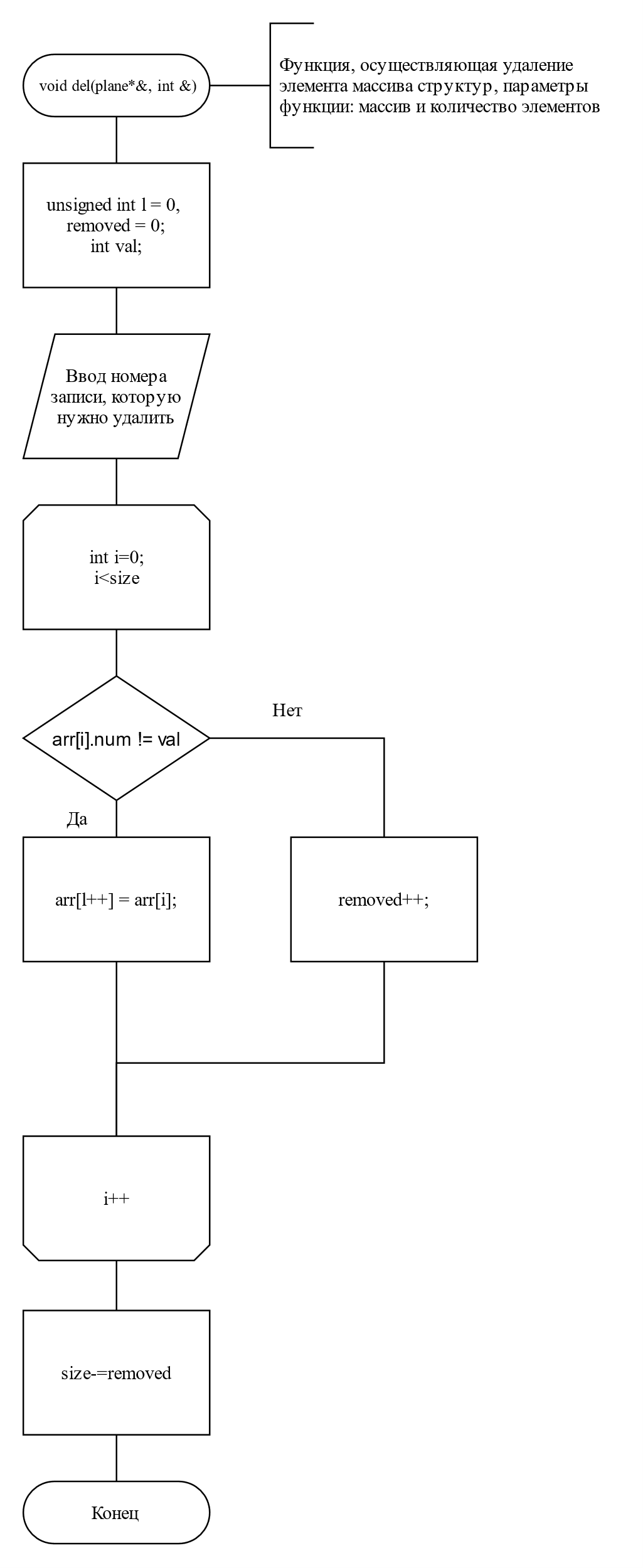
Для каждого рейса указаны его номер, тип самолета, пункт назначения, время вылета.Вывести все номера рейсов, вылетающих в заданный пункт назначения.

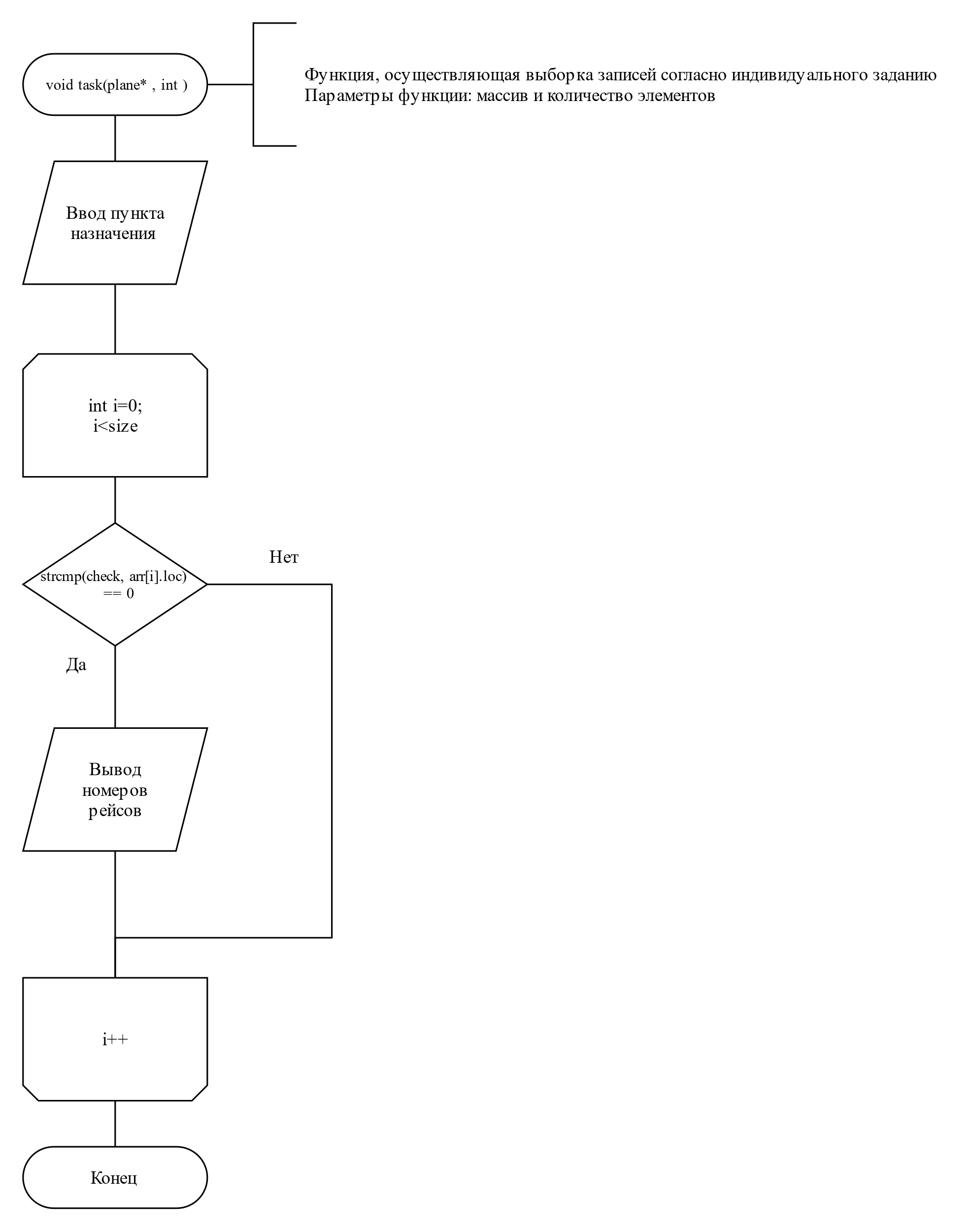
**Блок-схемы:**

****

****

****

****

****

**Код программы:**

//В справочной аэропорта имеется расписание вылета самолетов.

//Для каждого рейса указаны его номер, тип самолета, пункт назначения, время вылета.Вывести все номера рейсов, вылетающих в заданный пункт назначения.

//Реализовать работу с массивом структур через меню :

//ввод данных в массив, вывод собержимого массива на экран, сортировка по одному полю, удаления записи по заданному значению поля, выборка записей согласно индивидуального задания.

#include<iostream>

#include<Windows.h>

#include<iomanip>

#include<stdlib.h>

using namespace std;

struct plane

{

int num;

union inFlight //объединение

{

int key;

char ans[4];

}inf;

enum Type //перечисление

{

pass = 1,

trans,

mail,

cour

};

char typedes[15];

unsigned short hour : 5; //5 бит для 24-х часов //битовые поля

unsigned short min : 6; //6 для минут

char loc[30];

short time;

};

void menu(); //меню

void input(plane\*& arr, int &size); //ввод

void output(plane\* arr, int size); //вывод

void del(plane\*& arr, int& size); //удаление

void sort(plane\* arr, int size); //сортировка

void task(plane\* arr, int size); //задание

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

menu();

return 0;

}

void menu()

{

int size = 0;

int ch;

plane\* mass = new plane[size];

do {

cout << "Меню: \n";

cout << "1 - ввод данных\n";

cout << "2 - вывод данных\n";

cout << "3 - удаление записи по номеру\n";

cout << "4 - сортировать по времени вылета\n";

cout << "5 - Вывод номеров рейсов, вылетающих в данный пункт назначения\n";

cout << "0 - завершить работу с меню\n";

cout << "Выберите пункт: "; cin >> ch;

switch (ch)

{

case 1:

system("cls");

input(mass, size);

break;

case 2:

system("cls");

output(mass, size);

break;

case 3:

system("cls");

del(mass, size);

break;

case 4:

system("cls");

sort(mass,size);

break;

case 5:

system("cls");

task(mass, size);

break;

case 0:break;

default: cout << "Неверный ввод";

break;

}

} while (ch != 0);

/\*delete[] mass;\*/

}

void input(plane\*& arr, int& size)

{

int N;

cout << "Введите кол-во записей: "; cin >> N;

size = N;

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

{

arr[i].num = i + 1;

cout << "Номер самолёта: " << i+1 << endl;

int type;

cout << "Тип самолёта (1-Пассажирский, 2-Транспортный, 3-Почтовый, 4-Курьерский): "; cin >>type;

switch (type)

{

case (arr[i].pass):

{

strcpy(arr[i].typedes, "Пассажирский");

}break;

case (arr[i].trans):

{

strcpy(arr[i].typedes, "Транспортный");

}break;

case (arr[i].mail):

{

strcpy(arr[i].typedes, "Почтовый");

}break;

case (arr[i].cour):

{

strcpy(arr[i].typedes, "Курьерский");

}break;

}

cin.ignore();

cout << "Пункт назначения: ";

gets\_s(arr[i].loc);

cout << "Время вылета (часы : минуты): \n";

cin >> arr[i].time;

arr[i].hour = arr[i].time;

cin >> arr[i].time;

arr[i].min = arr[i].time;

cout << "В полёте (1-Да, 2-Нет): "; cin >> arr[i].inf.key;

if (arr[i].inf.key == 1)

strcpy(arr[i].inf.ans, "Да");

if (arr[i].inf.key == 2)

strcpy(arr[i].inf.ans, "Нет");

cout << endl;

}

}

void output(plane\* arr, int size)

{

cout << left << setw(8) << "|Номер" << setw(14) << left << "|Тип самолёта" << setw(19) << left << "|Пункт назначения" << setw(14) << left << "|Время вылета" << setw(10) << left << "|В полёте|" << endl;

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

{

cout << left << setw(9) << arr[i].num << setw(15) << left << arr[i].typedes << setw(20) << left << arr[i].loc << setw(2) << left <<arr[i].hour<<":"<<setw(12)<<left<<arr[i].min << setw(10) << left << arr[i].inf.ans << endl;

}

}

void del(plane\*& arr, int& size)

{

int val;

cout << "Номер записи, которую удалить- "; cin >> val;

unsigned int l = 0, removed = 0;

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

{

if (arr[i].num != val)

arr[l++] = arr[i];

else

removed++;

}

size -= removed;

}

void sort(plane\* arr, int size)

{

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

{

for (int j = i; j < size; j++)

{

if (arr[i].hour > arr[j].hour)

{

swap(arr[i], arr[j]);

}

}

}

}

void task(plane\* arr, int size)

{

char check[30];

cout << "Введит пункт назначения- ";

cin.ignore();

gets\_s(check);

cout << "Номера рейсов:\n";

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

{

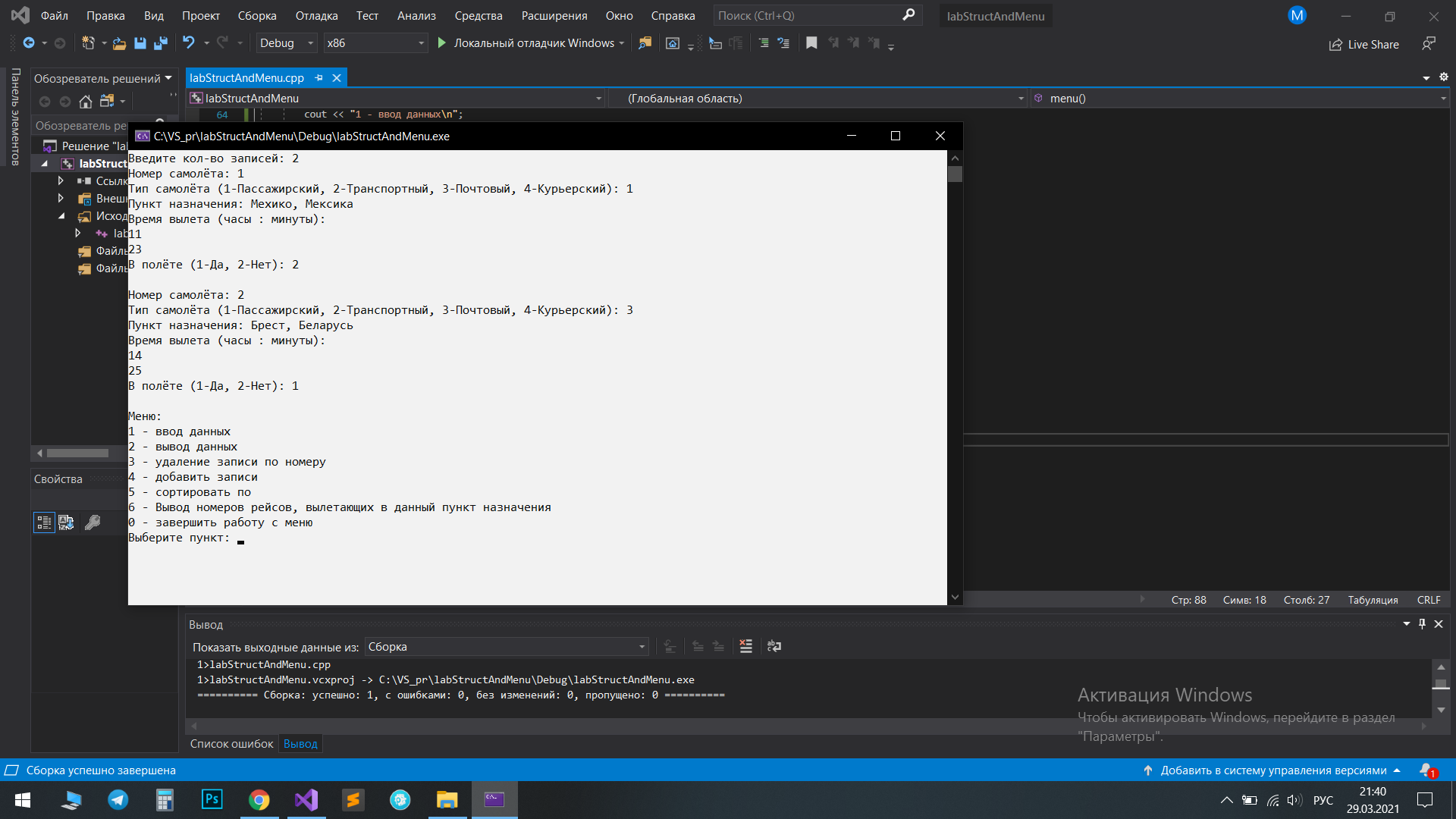
if (strcmp(check, arr[i].loc) == 0)

cout << arr[i].num << endl;

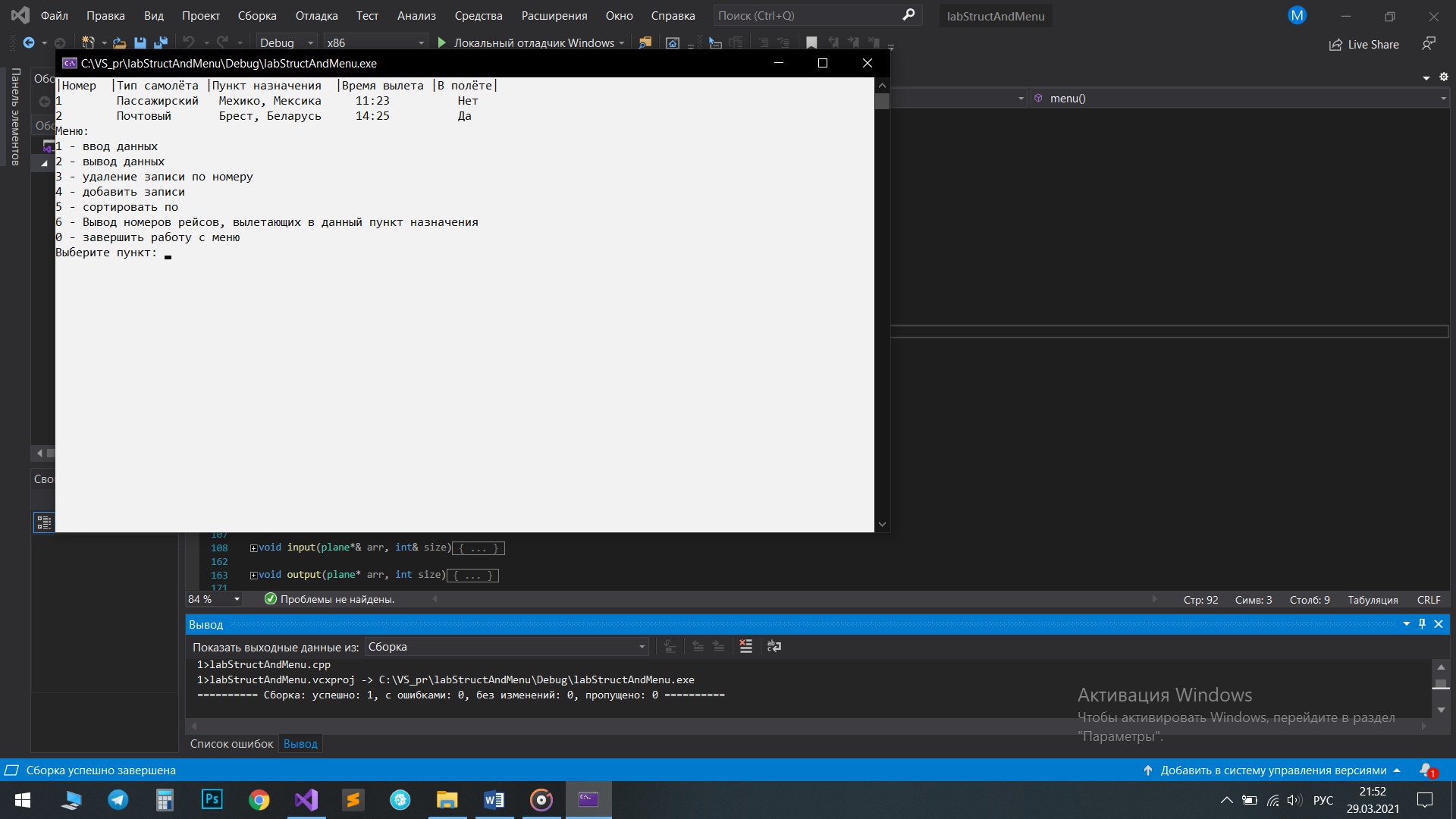
}

}

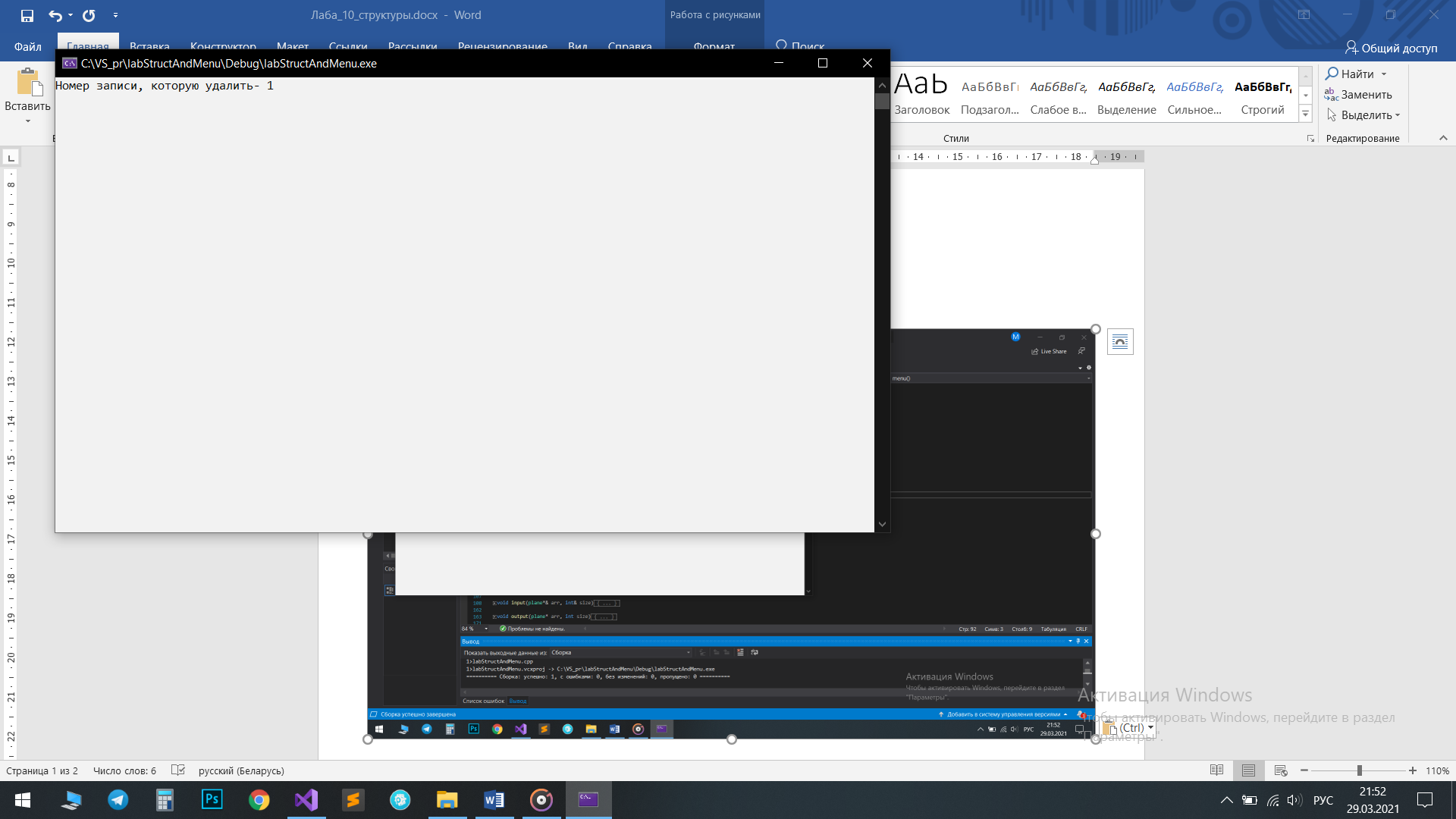
Ввод

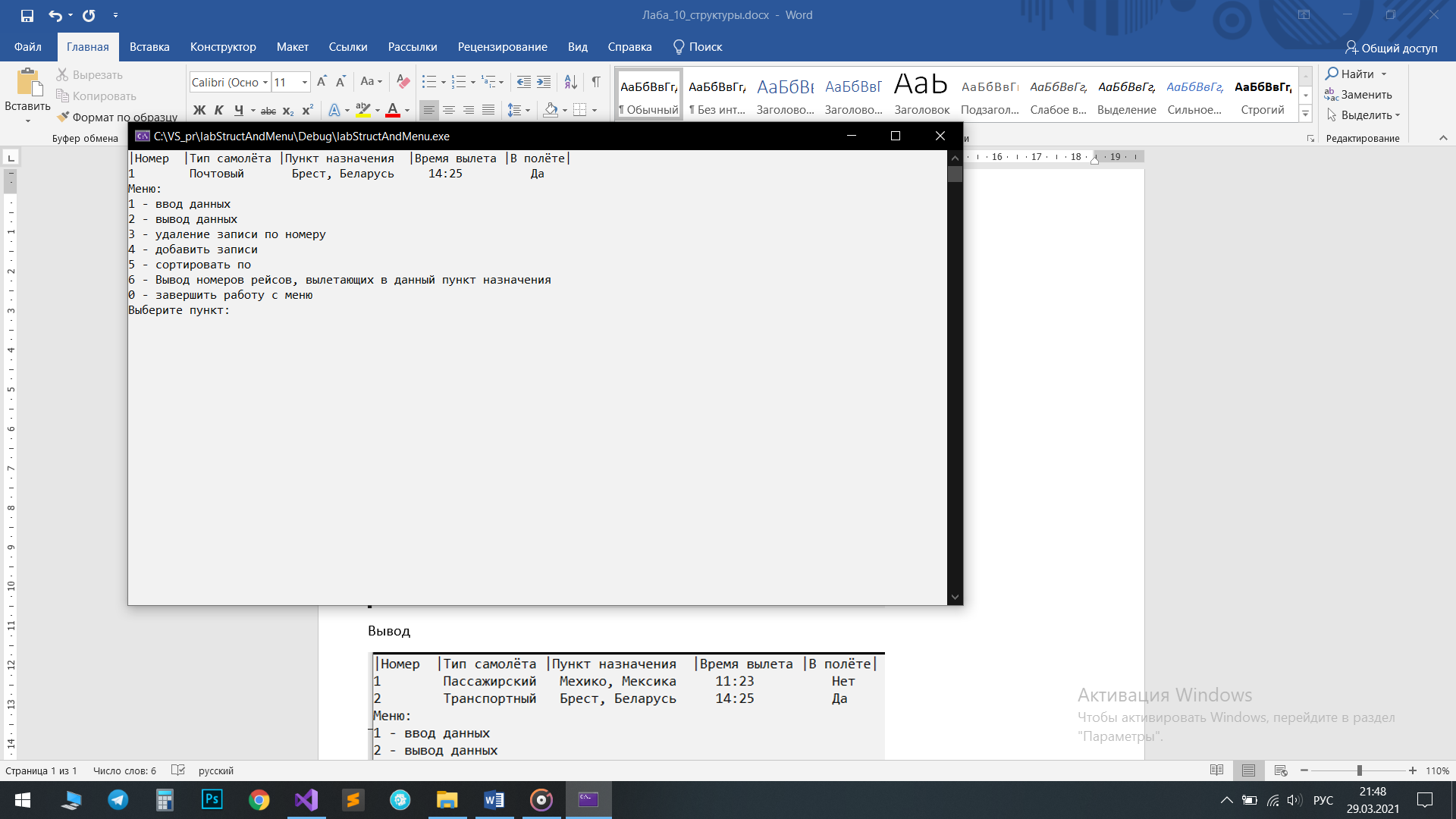


Вывод

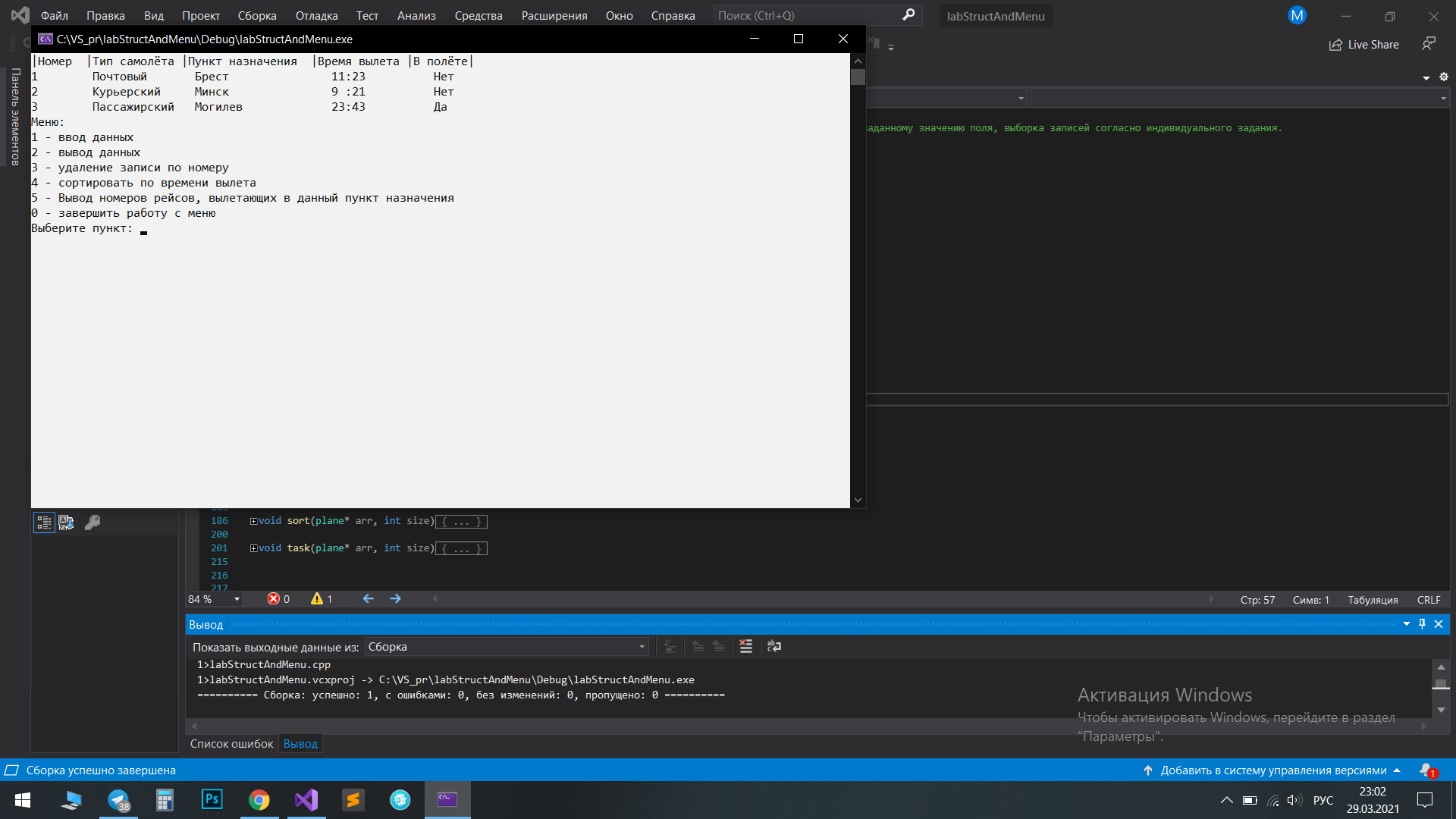


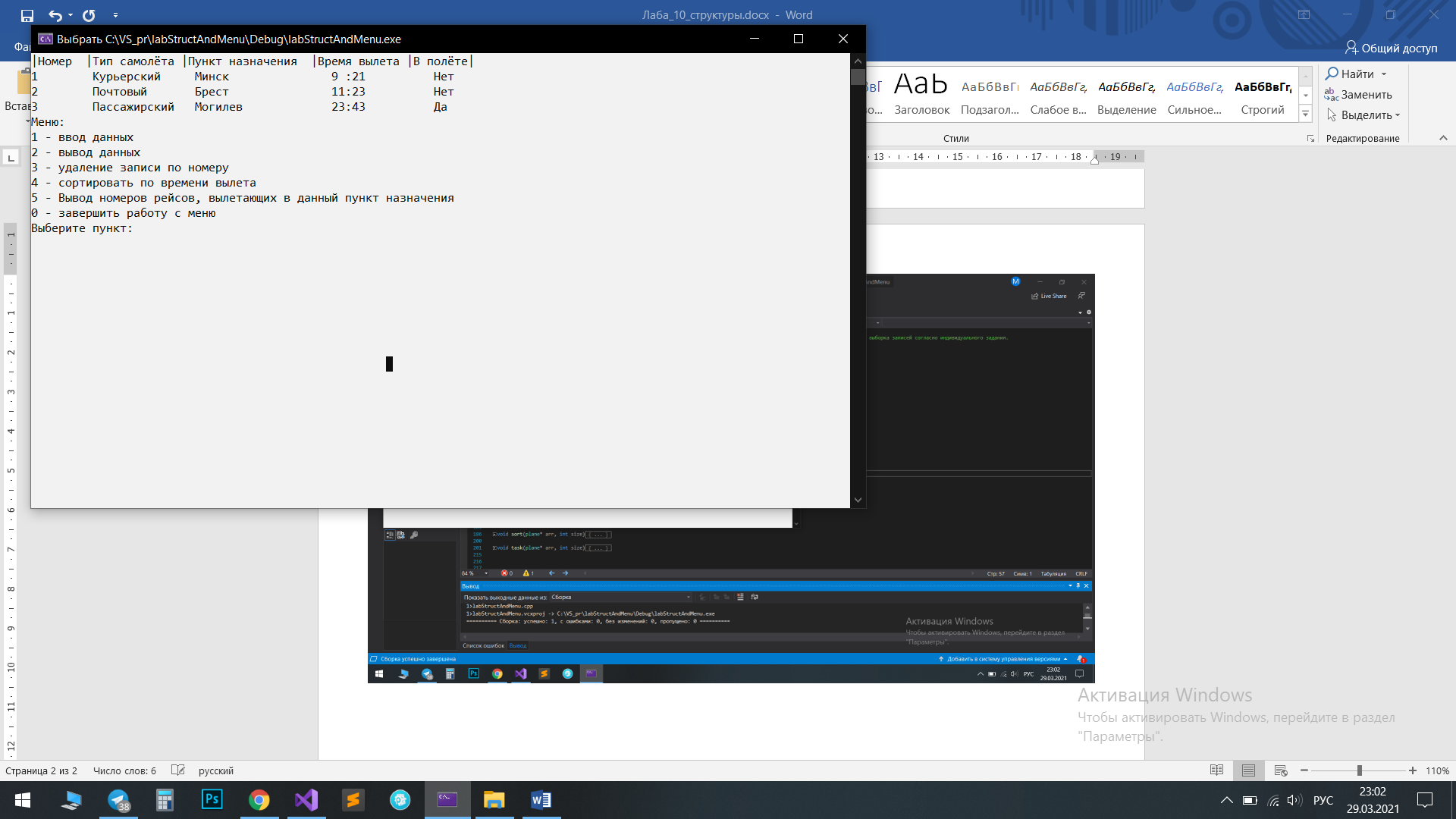
Удаление и снова вывод



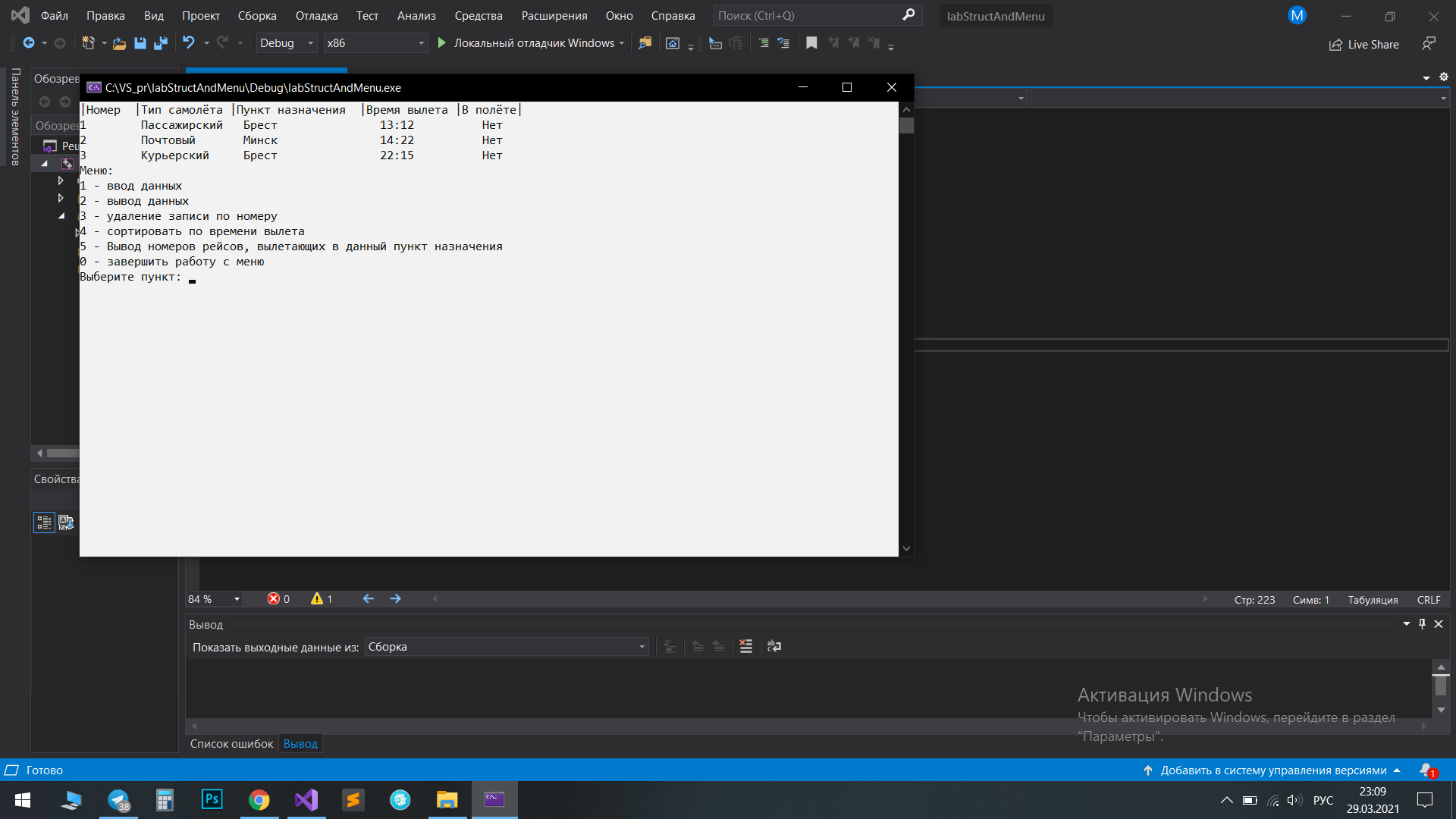


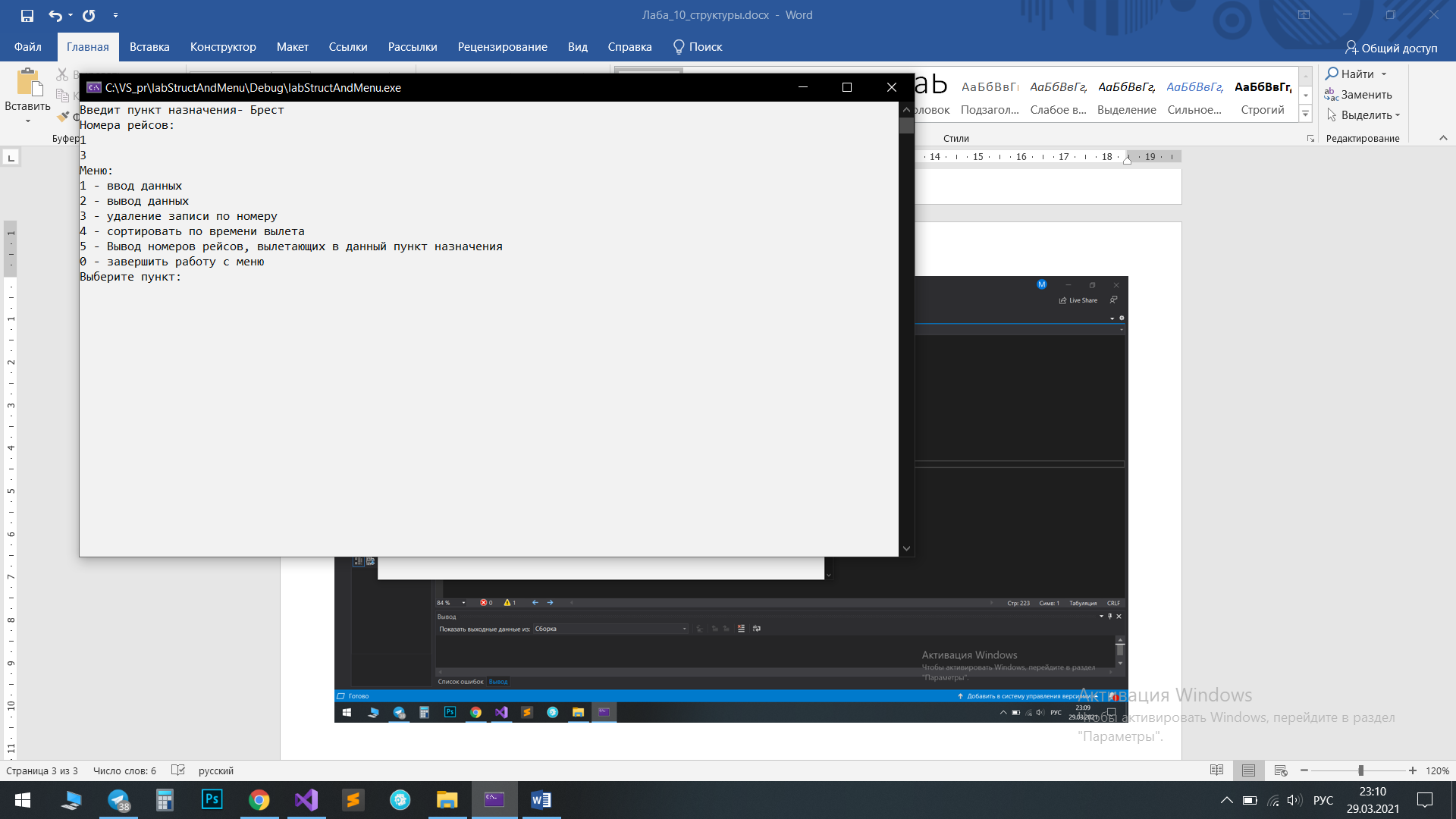
Сортировка





Поиск





**Вывод:** Я изучил синтаксис и правила работы со структурами, реализовал программу с применением структур, перечислений и объединений

.