**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования**

**«МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ СВЯЗИ И ИНФОРМАТИКИ»**

**(МТУСИ)**

Кафедра «Информационная безопасность»

ЛАБОРАТОРНАЯ РАБОТА №0

по дисциплине

«Разработка безопасного ПО»  
на тему

«Разработка консольных приложений на языке программирования C»

Выполнил:

студент группы БПЗ1902

Висягина М.О.

Проверил:

старший преподаватель кафедры ИБ

Барков В.В.

Москва, 2021

**Цель практикума:**

Овладеть навыками создания однофайловых и многофайловых проектов и научится создавать программы на языке C с применением арифметических операций, инструкций выбора, итерации, массивов, строк и структур.

Овладеть навыками создания разветвляющихся алгоритмов на языке C с применением инструкций выбора, условных выражений, логических операций и операций отношения.

Овладеть навыками создания циклических алгоритмов на языке C с применением инструкций цикла на примере алгоритмов работы с последовательностями чисел.

Овладеть навыками работы с указателями и массивами в языке программирования C.

Овладеть навыками работы со строками в языке программирования C.

Овладеть навыками работы со структурами, объединениями, перечислениями и битовыми полями в языке C.

Овладеть навыками выделения динамической памяти в языке программирования C.

**Задание:**

На языке программирования C разработать консольное приложение для сортировки и вывода на экран списка учащихся.

**Задание 1**

Разработать функцию лексикографического сравнения двух строк.

Функция должна иметь следующий прототип:

int CompareStrings(const char \*lhs, const char \*rhs);

Функция возвращает:

• 0, если строки равны

• -1, если строка lhs предшествует строке rhs

• 1, если строка rhs предшествует строке lhs

**Задание 2**

Разработать структуру Person, которая включает в семя следующие поля:

• Фамилия (строка)

• Имя (строка)

• Отчество (строка)

• Год рождения (беззнаковое число)

**Задание 3**

Разработать перечисление SortType, включающее два значения:

• Asc – сортировка по возрастанию

• Desc – сортировка по убыванию

Разработать функцию сортировки массива структур Person.

Функция должна иметь следующий прототип:

void Sort(struct Person \*persons, int count, enum SortType sortType);

Функция сортирует массив persons по возрастанию или убыванию в зависимости от значения параметра sortType. Сортировка производится по следующим полям:

• Фамилия

• Имя

• Отчество

• Возраст

Сравнение строк организовать с помощью функции, разработанной в задании 1.

**Задание 4**

Разработать функцию, которая получает аргументы командной строки и создаёт на его основе динамический массив структур. Размер массива равен количеству аргументов -person в аргументах командной строки.

**Задание 5**

Разработать функцию, которая получает на вход массив структур Person и выводит его на экран в виде таблицы.

**Задание 6**

Разработать функцию main, которая вызывает функцию из задания 5 для создания массива стуктур Person из аргументов командной строки, запрашивает тип сортировки, сортирует массив с помощью функции из задания 3 и выводит на экран с помощью функции из задания 5.

**Задание 7**

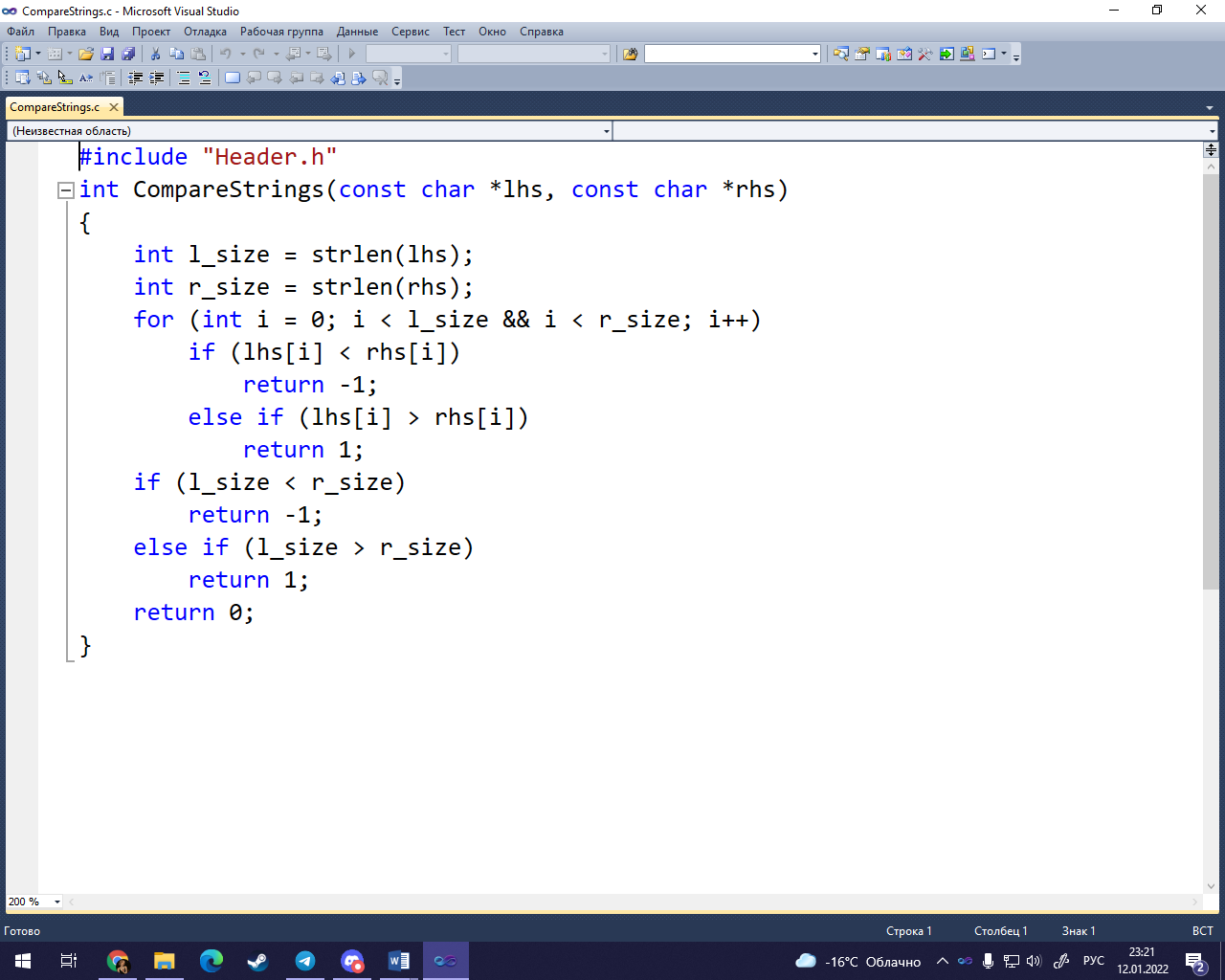
Скомпилировать разработанные в заданиях 1-6 исходные файлы. Заголовочные файлы не передаются на вход компилятору, а включаются препроцессором в ходе компиляции исходных файлов. Создать на основе откомпилированных файлов (объектных файлов) из заданий 1- 5 статическую библиотеку. Скомпоновать откомпилированный файл из задания 6 и статическую библиотеку в исполняемый файл.

**Выполнение:**

ссылка на репозиторий: https://github.com/migonet/RBPO

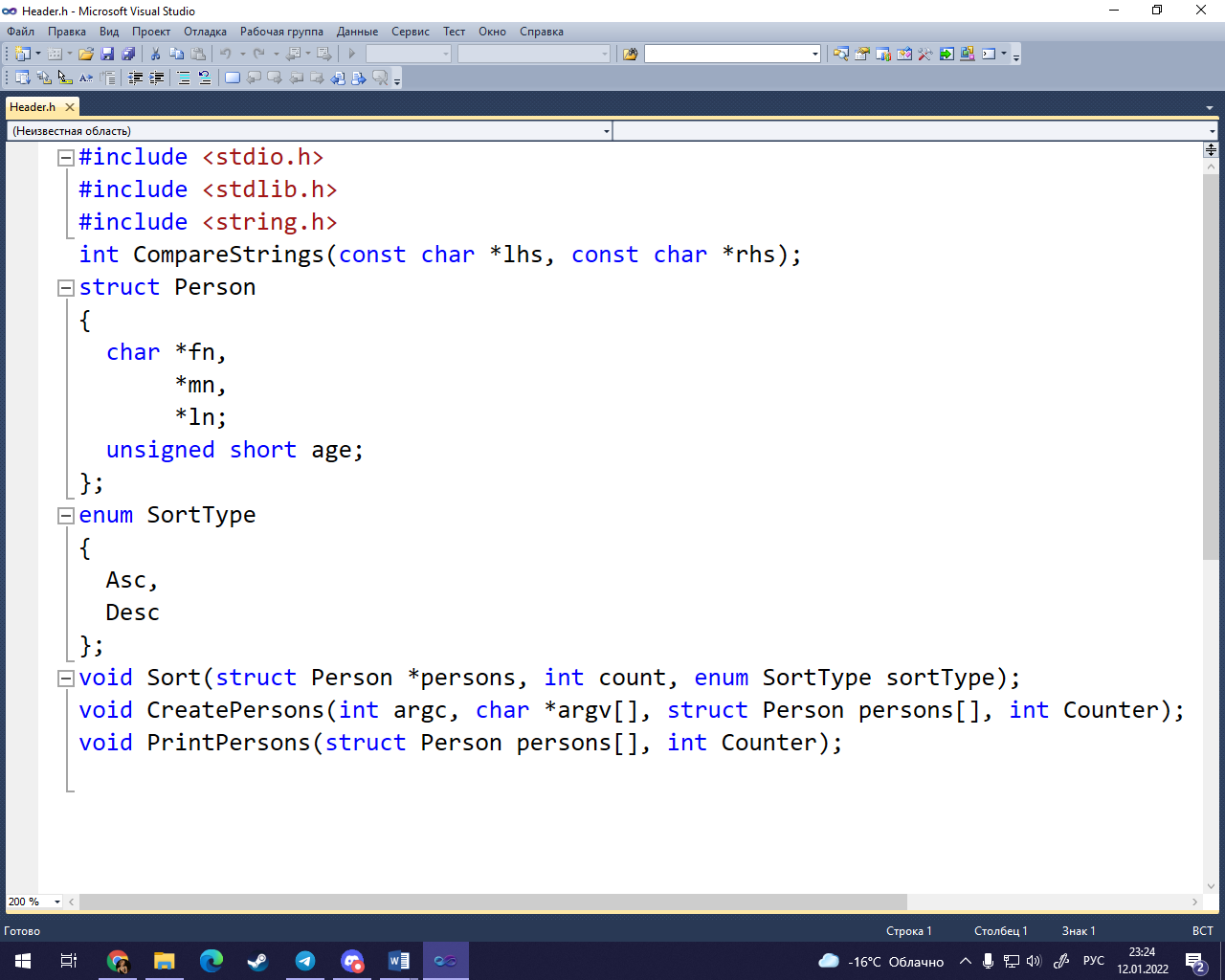
Задание 1:

Разработка функции лексикографического сравнения двух строк.



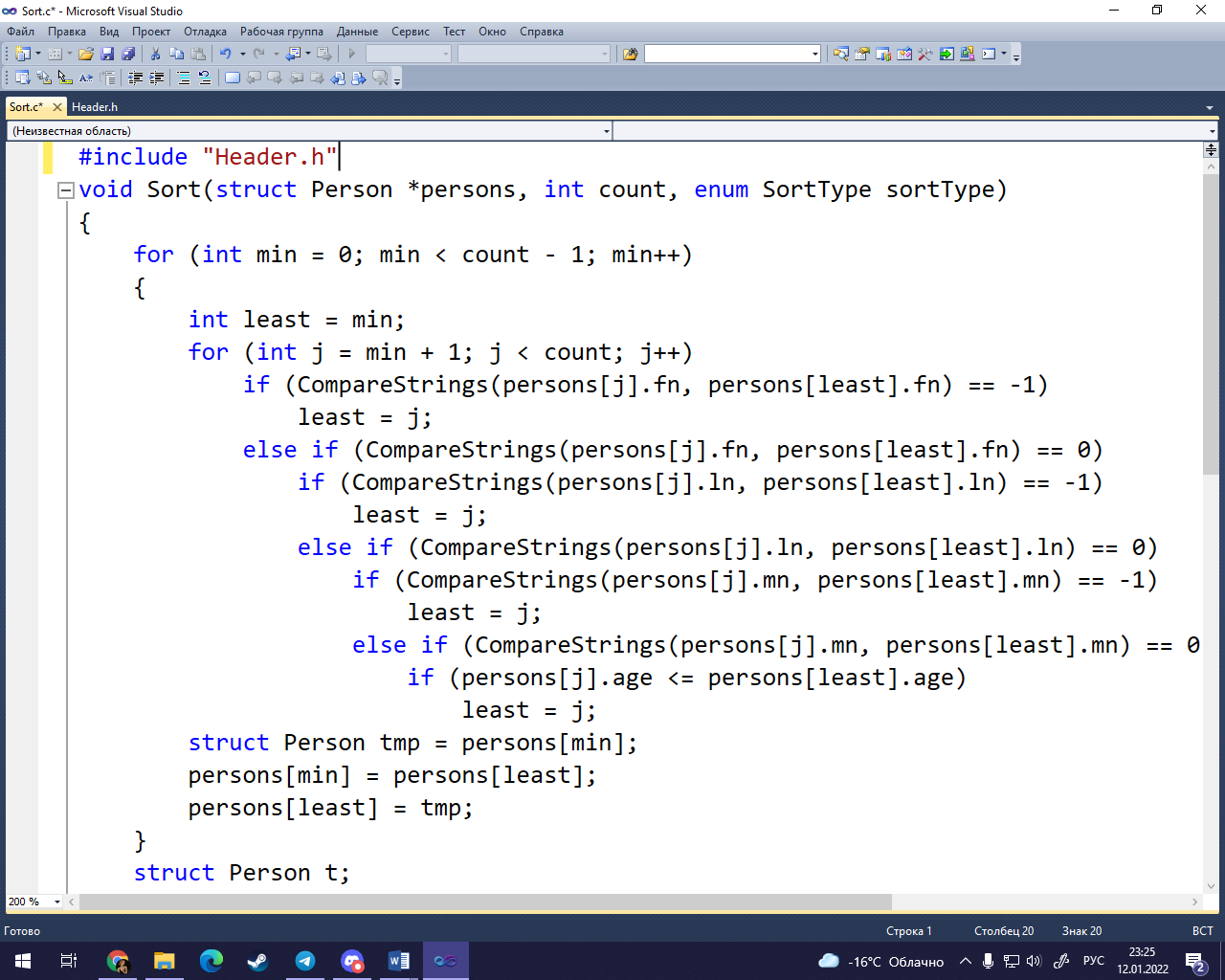
**Задание 2:**

Разработать структуру Person.



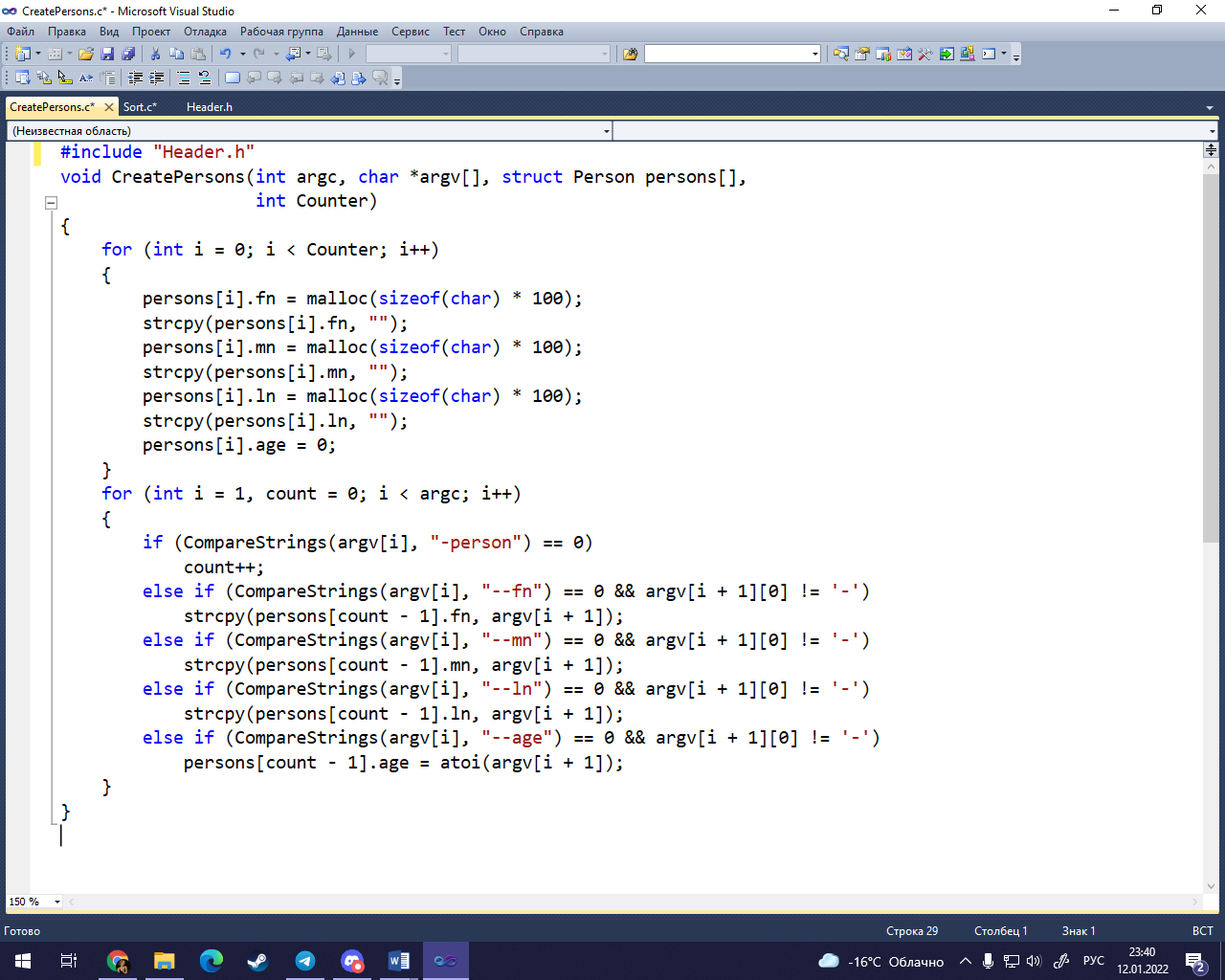
**Задание 3:**

Разработать перечисление SortType.



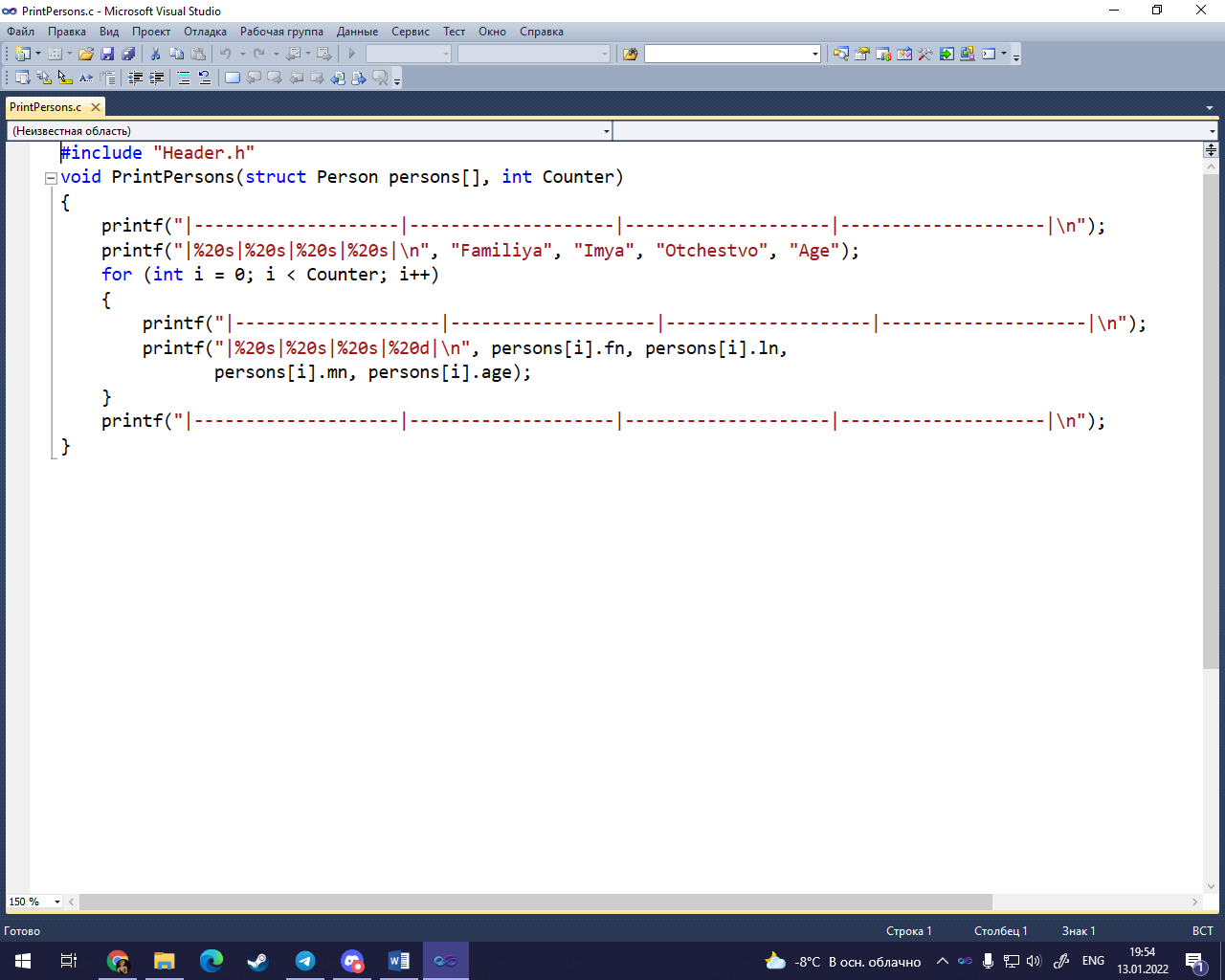
**Задание 4:**

Разработать функцию, которая получает аргументы командной строки и создаёт на его основе динамический массив структур.

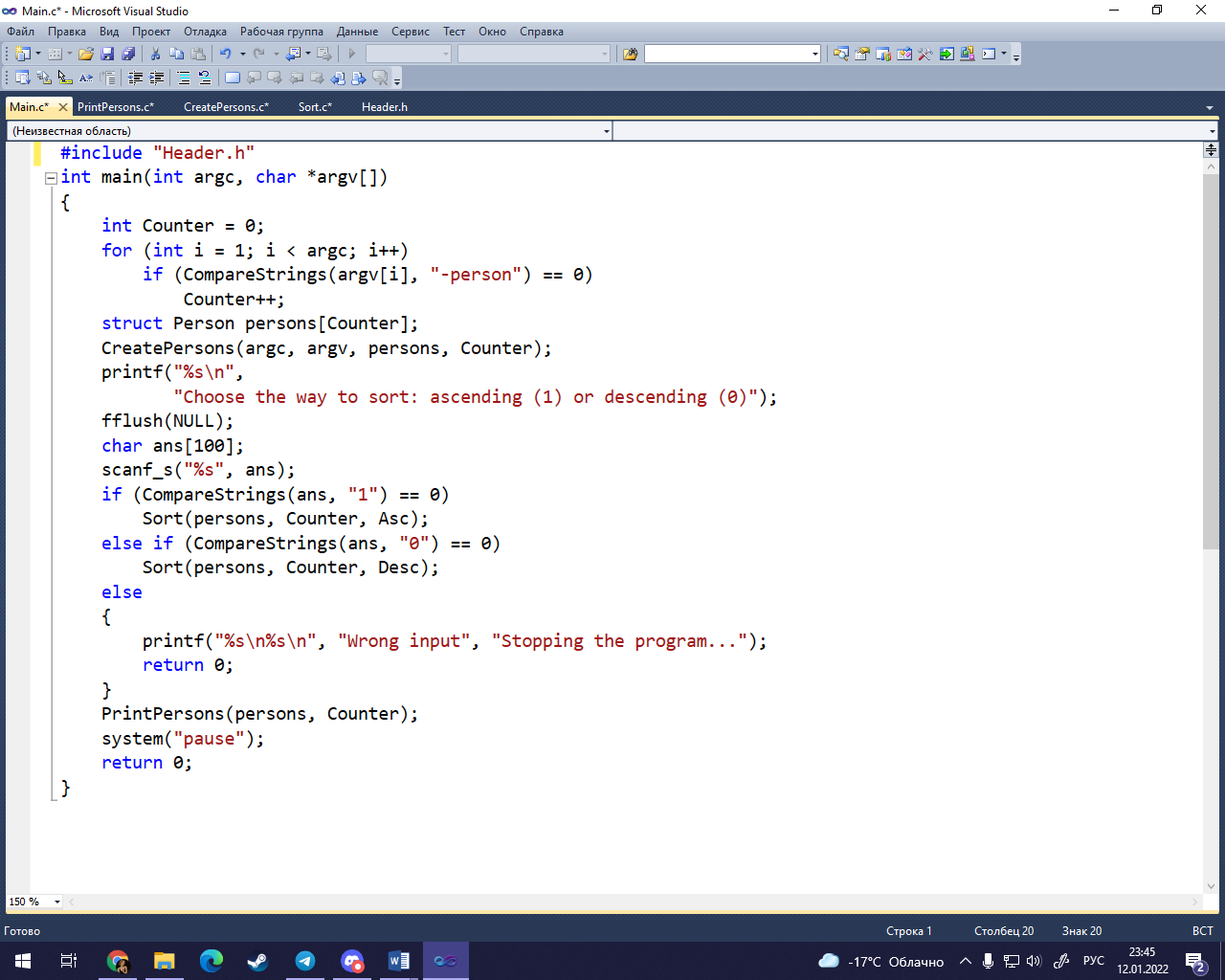


**Задание 5:**

Разработать функцию, которая получает на вход массив структур Person и выводит его на экран в виде таблицы.



**Задание 6:**  
Разработать функцию main, которая вызывает функцию из задания 5 для создания массива стуктур Person из аргументов командной строки, запрашивает тип сортировки, сортирует массив с помощью функции из задания 3 и выводит на экран с помощью функции из задания 5.



Запускаем командную строку и вводим:

./app -person --fn Selutina --ln Aly --mn Alekseevna --age 20 -person --fn Hlibbova --ln Tanya --age 21 --mn Alekseevna

**Результат**:

