**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»**

Кафедра «Информационная безопасность»

ЛАБОРАТОРНАЯ РАБОТА №2

по дисциплине

«Разработка безопасного ПО»  
на тему

«Компиляция файлов исходного кода и компоновка полученных объектных файлов в исполняемый модуль. Объявление и определение переменных, функций и пространств имён. Линейные алгоритмы»

Выполнил:

студент группы БПЗ1902

Кварацхелия Д.Т.

Проверил:

старший преподаватель  
 кафедры ИБ

Барков В.В.

Москва 2021

Цель практикума:

Овладеть навыками создания однофайловых и многофайловых проектов в интегрированной среде разработки Microsoft Visual Studio 2015 Community Edition и научится создавать линейные программы на языке C с применением арифметических операций.

Задание:

По номеру Вашего варианта выбрать задачу, решаемую в этой лабораторной работе, и выполнить для нее следующие задания.

Задание 1

Составить программу, имеющую линейный алгоритм и состоящую из одной функции void main(). Программу записать в файл с именем task1.cpp. Скомпилировать, скомпоновать и выполнить.

В функции main организовать вычисление задачи вашего варианта дважды:

• для исходных данных, значения которых задать в виде литералов в тексте функции main;

• для исходных произвольных данных, значения которых пользователь Вашей программы должен ввести с клавиатуры в процессе выполнения программы.

Проанализировать результаты работы и сделать выводы. Перенести полученные результаты в отчет. Вывод действительных чисел осуществлять с точностью до 0.0001

Задание 2

Линейный алгоритм функции из задания 1 разделить на две процедуры, выделив в одну вычислительные операции этого алгоритма, а в другую все операции ввода-вывода.

Вычислительную часть алгоритма оформить в виде функции f.

Входные данные необходимо передавать через параметры функции по значению.

Выходные данные необходимо возвращать через возвращаемое значение функции.

Прототип функции:

double f(double x) // Если функция имеет один параметр

double f(double x, double y) // Если функция имеет два параметра

Другую часть алгоритма оформить как функцию void main(), вызывающую функцию f нужное количество раз.

Записать тексты функций файл с именем task2.сpp в следующем порядке: функция f, функция main.

Cкомпилировать, скомпоновать и выполнить.

Задание 3

Создать файл task3.сpp, в котором изменить порядок записи текстов функций, созданных в задании 2. Функции записать в следующем порядке: функция main, функция f.

Внести требуемые дополнения, добиться успешной компиляции, скомпоновать и выполнить.

Задание 4

Линейный алгоритм функции из задания 1 разделить на две процедуры, выделив в одну вычислительные операции этого алгоритма, а в другую все операции ввода-вывода.

Вычислительную часть алгоритма оформить в виде функции f.

Входные данные необходимо передавать через параметры функции по ссылке на константный объект.

Выходные данные необходимо возвращать через дополнительный параметр функции по ссылке. Функция не имеет возвращаемого значения.

Прототип функции:

void f(const double& x, double& result) // или

void f(const double& x, const double& y, double& result)

Другую часть алгоритма оформить как функцию void main(), вызывающую функцию f нужное количество раз.

Записать тексты функций в файл с именем task4.сpp в следующем порядке: функция main, затем функция f. Cкомпилировать, скомпоновать и выполнить.

Задание 5

Линейный алгоритм функции из задания 1 разделить на две процедуры, выделив в одну вычислительные операции этого алгоритма, а в другую все операции ввода-вывода.

Вычислительную часть алгоритма оформить в виде функции f.

Входные данные необходимо передавать через глобальные объекты x и y.

Выходные данные необходимо возвращать через глобальный объект result. Функция не имеет возвращаемого значения.

Прототип функции:

void f();

Другую часть алгоритма оформить как функцию void main(), вызывающую функцию f нужное количество раз.

Записать тексты функций в файл с именем task5.сpp в следующем порядке: функция main, затем функция f. Cкомпилировать, скомпоновать и выполнить.

Задание 6

В этом задании необходимо разделить текст файла task2.сpp на два файла. В первый файл с именем task6\_main.cpp поместить текст функции main. Скомпилировать только файл task6\_main.cpp.

Во второй файл с именем task6\_func.сpp поместить текст функции f. Скомпилировать только файл task6\_func.cpp.

После раздельной компиляции осуществить совместную компоновку. Полученный исполняемый файл выполнить.

Проанализировать результаты работы и сделать выводы.

Задание 7

В этом задании необходимо разделить текст файла task5.сpp на два файла.

В первый файл с именем task7\_main.cpp поместить текст функции main. Скомпилировать только файл task7\_main.cpp.

Во второй файл с именем task7\_func.сpp поместить определения глобальных объектов x, y и result и текст функции f.

Скомпилировать только файл task7\_func.cpp.

После раздельной компиляции осуществить совместную компоновку. Разобраться в проблемах, возникающих при совместной компиляции и компоновки. Полученный исполняемый файл выполнить.

Проанализировать результаты работы и сделать выводы.

Задание 8

В этом задании необходимо модифицировать тексты файлов из задания 7.

Описание функции и внешних переменных выделить в отдельный заголовочный файл func.h, включить его содержимое в файлы task8\_main.cpp и task8\_func.cpp. Определение внешних переменных произвести в файле task8\_func.cpp.

Скомпилировать по отдельности файлы task8\_main.cpp и task8\_func.cpp. Произвести компоновку, выполнить полученный исполняемый файл.

Задание 9

В этом задании необходимо модифицировать тексты файлов из задания 8.

Перенести все объявления и определения функции f в пространство имен, содержащее Вашу фамилию.

Скомпилировать по отдельности файлы task9\_main.cpp и task9\_func.cpp. Произвести компоновку, выполнить полученный исполняемый файл.

Задание 10

В этом задании необходимо собрать статическую библиотеку task10\_lib.lib. В библиотеку включить объектный файл task9\_func.obj из задания 9.

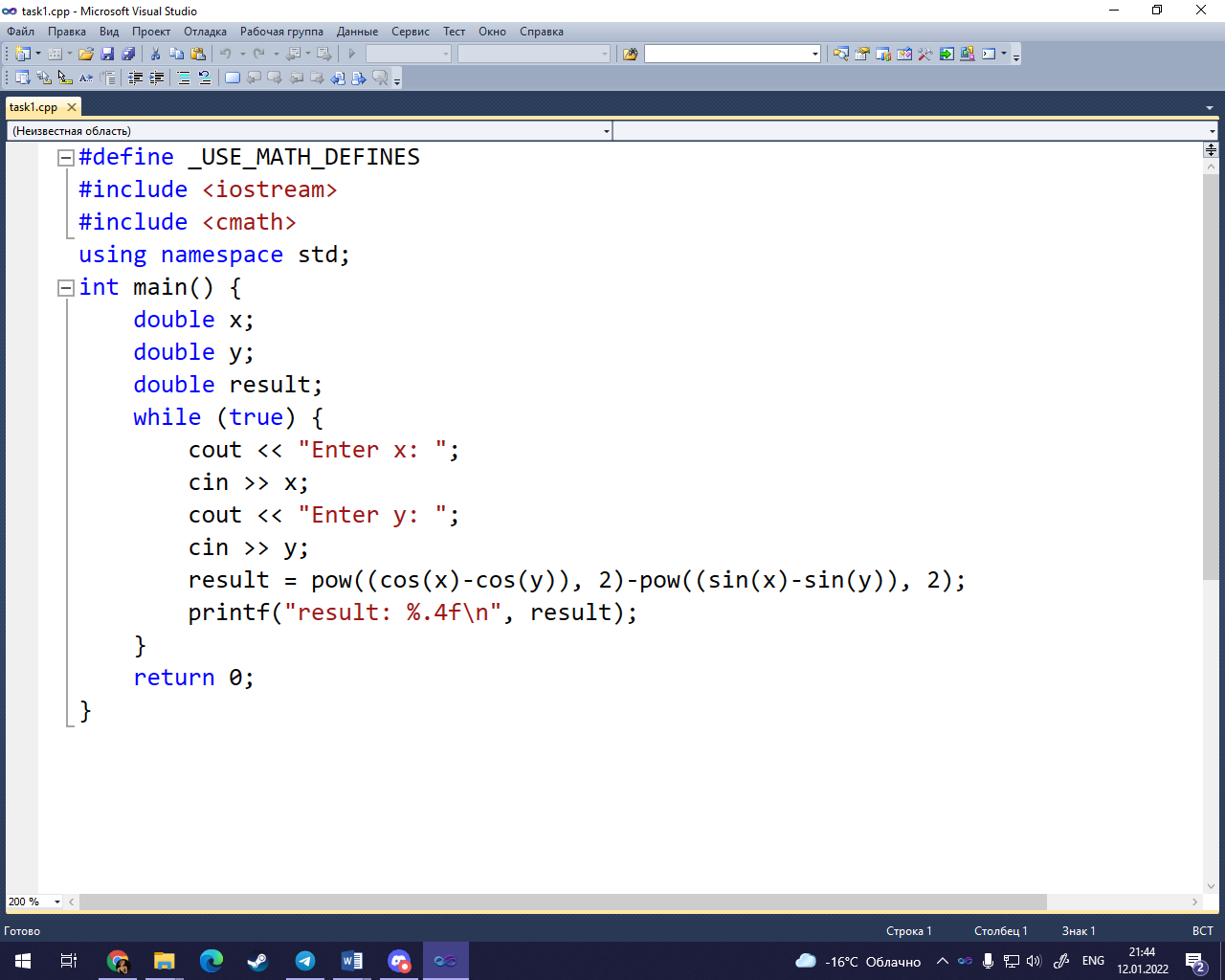
Произвести компоновку объектного файла task9\_main.obj из задания 9 и полученную статическую библиотеку task10\_lib.lib. Выполнить полученный исполняемый файл.

Выполнение:

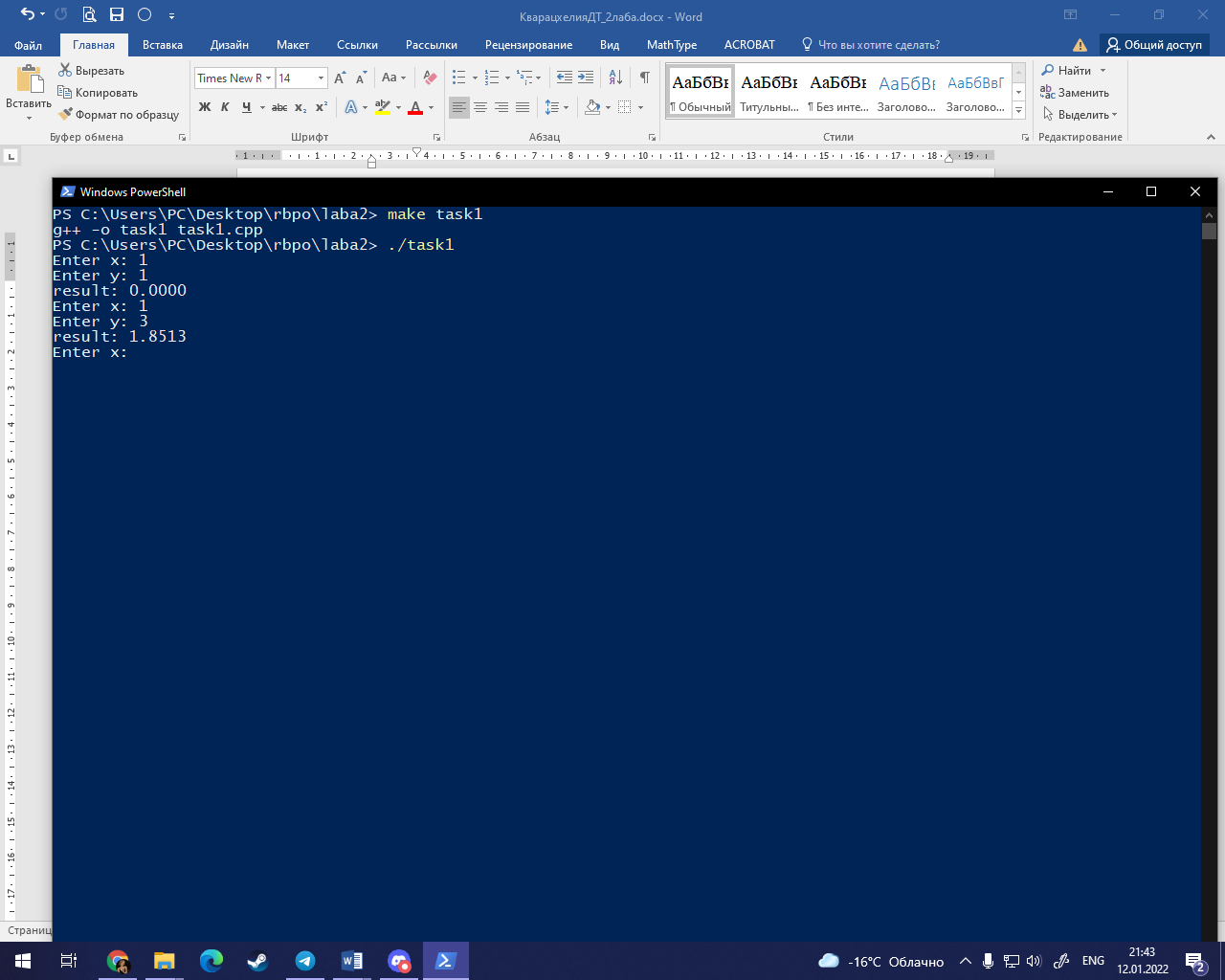
Индивидуальный вариант – 9:

ссылка на репозиторий: https://github.com/Cutie0906/RbPO\_2.git

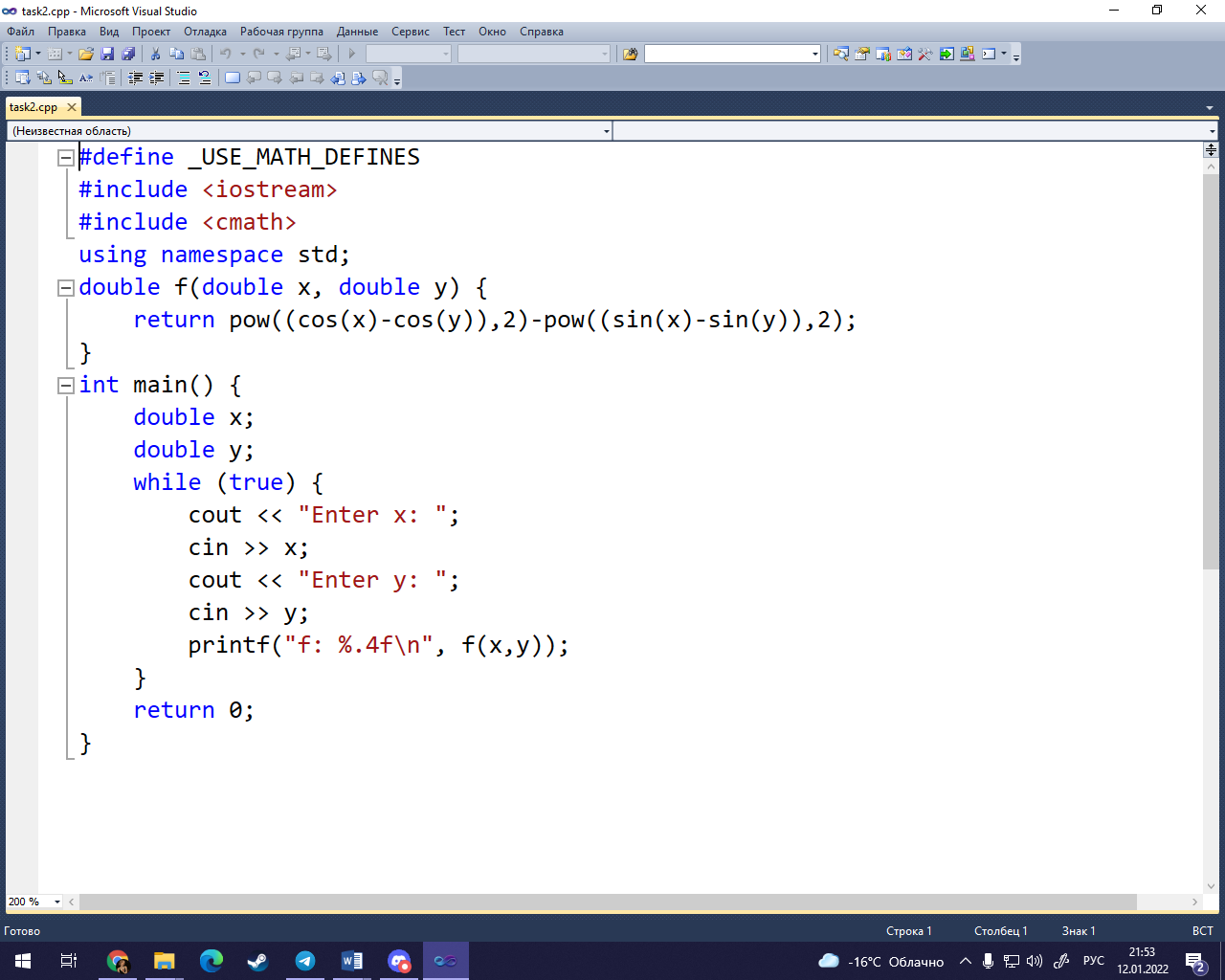
Задание 1:



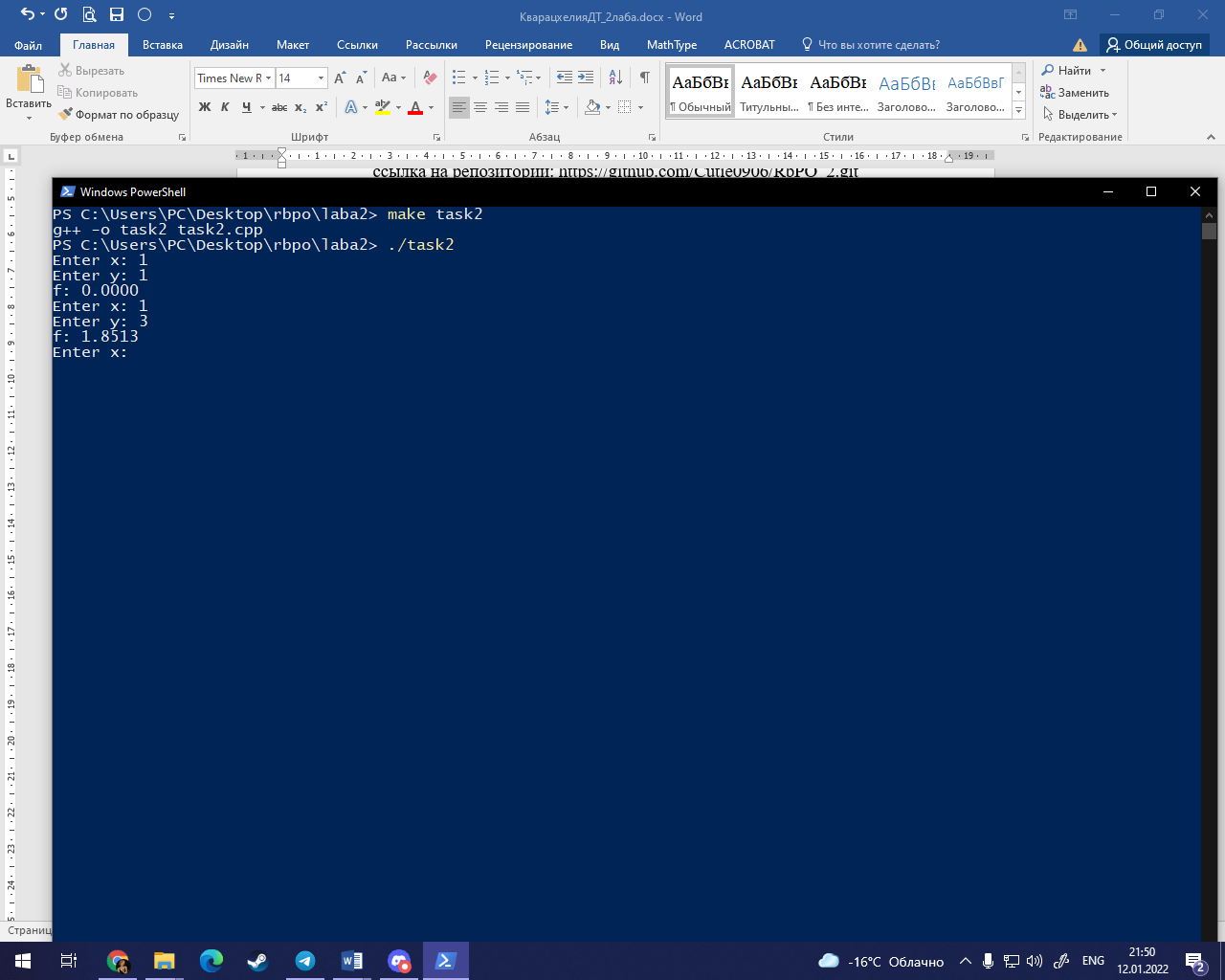
Результат:



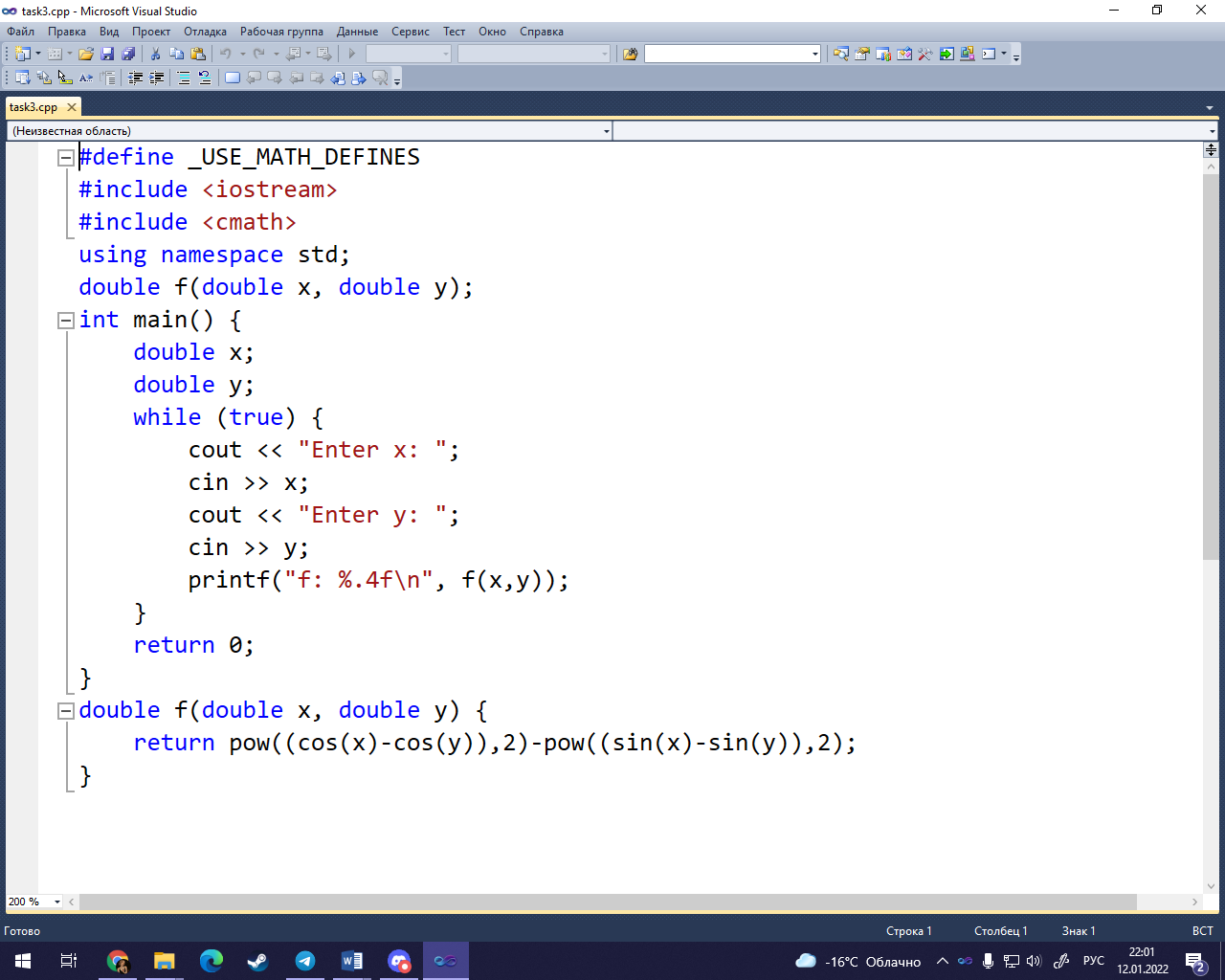
Задание 2:



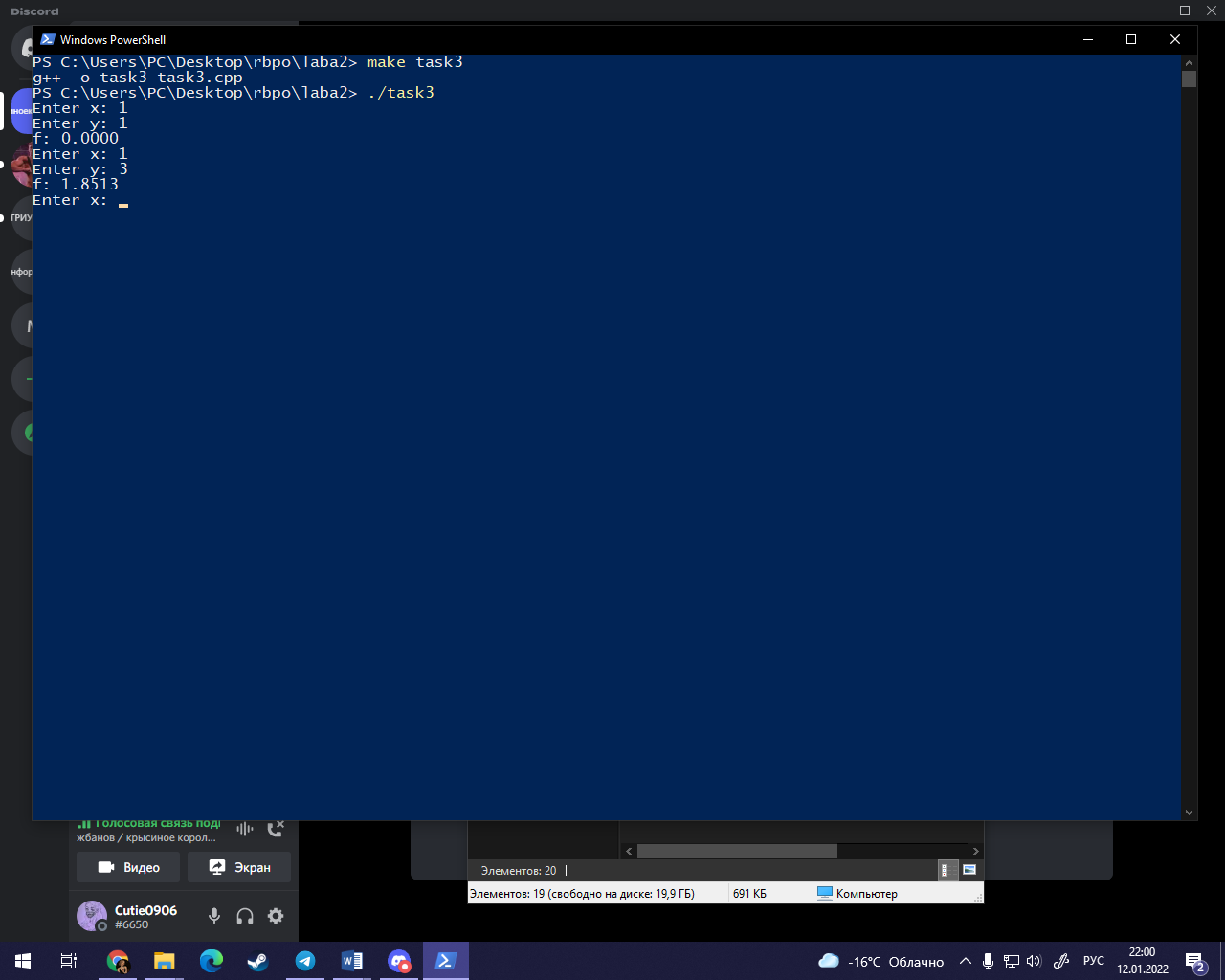
Результат:



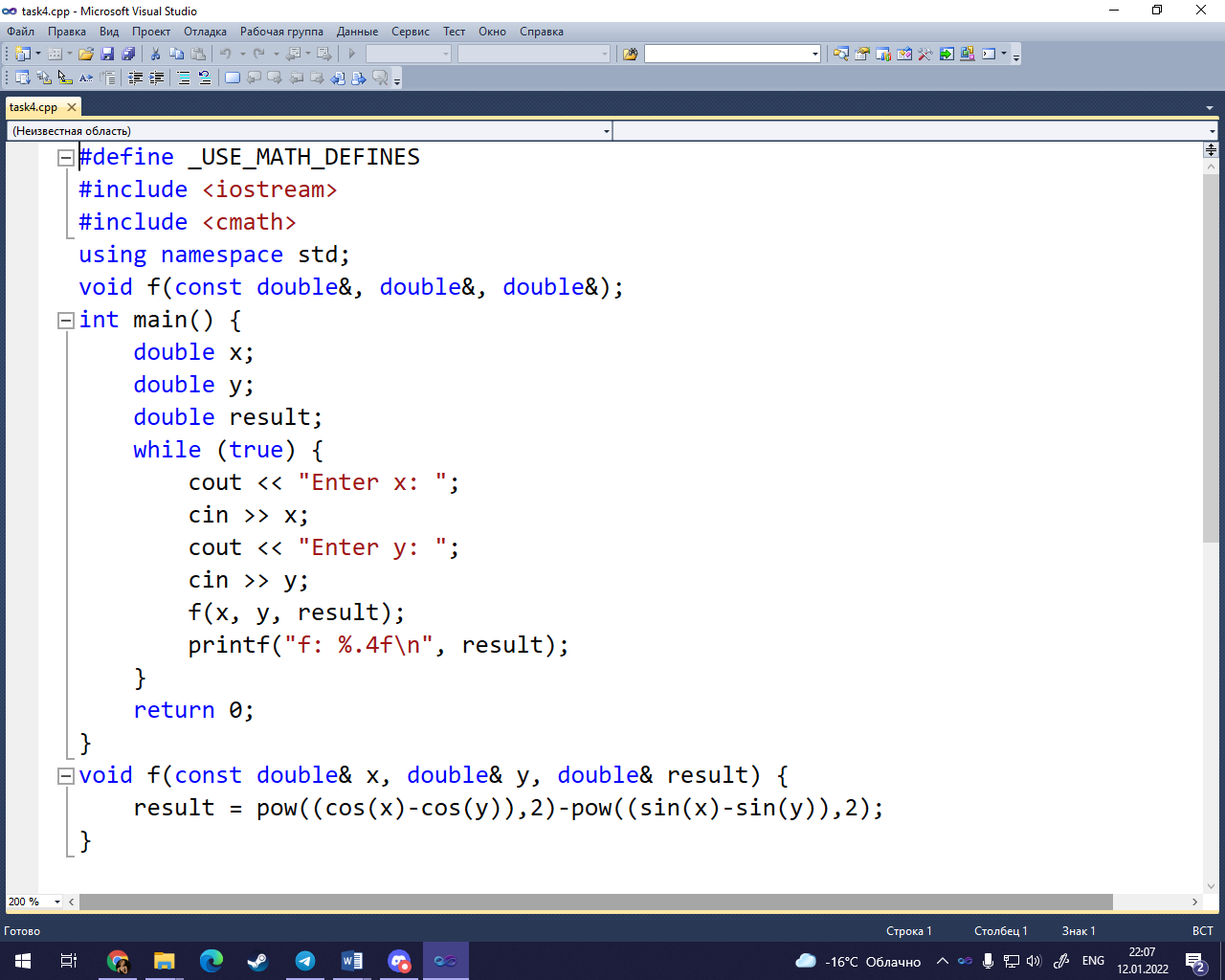
Задание 3:



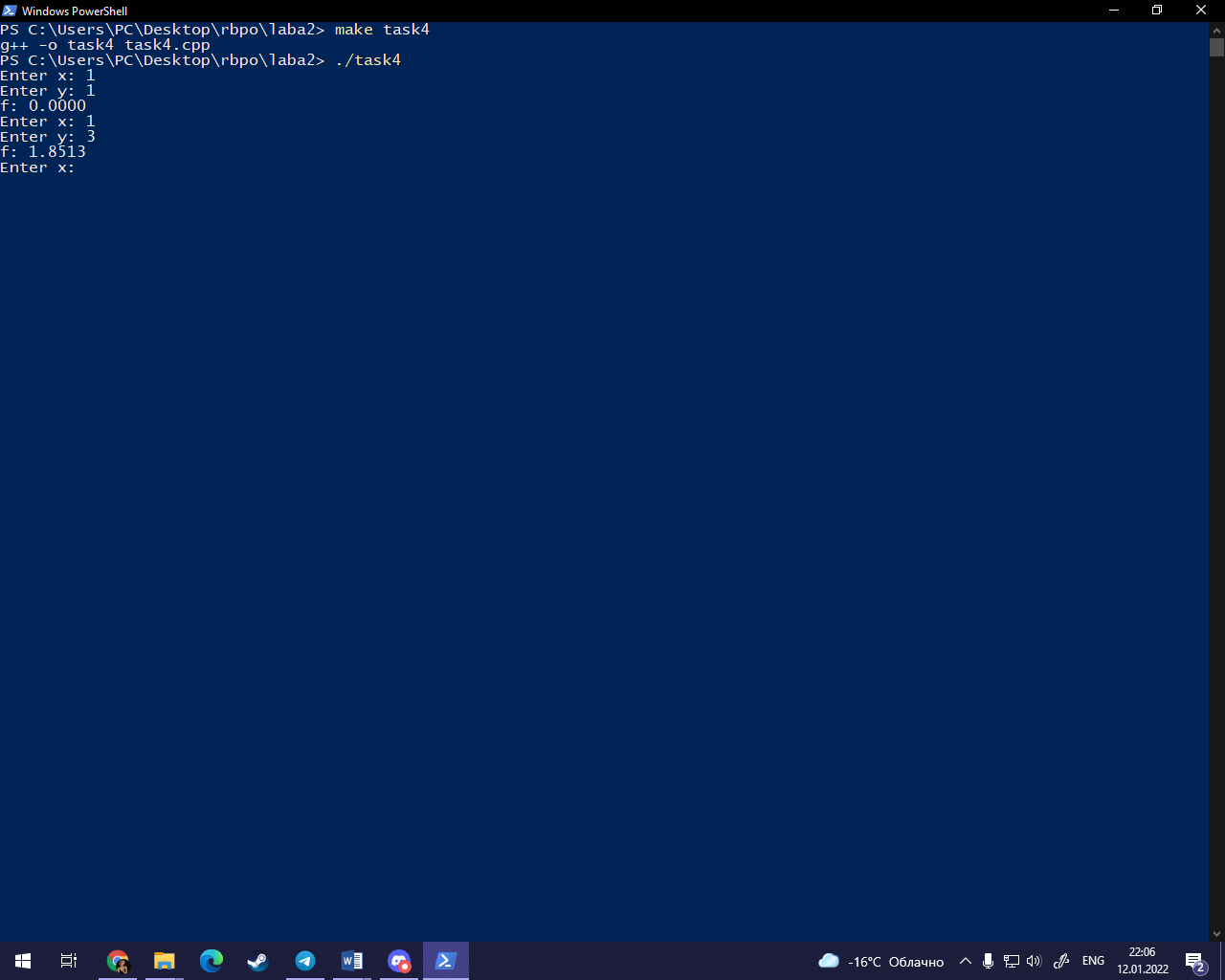
Результат:



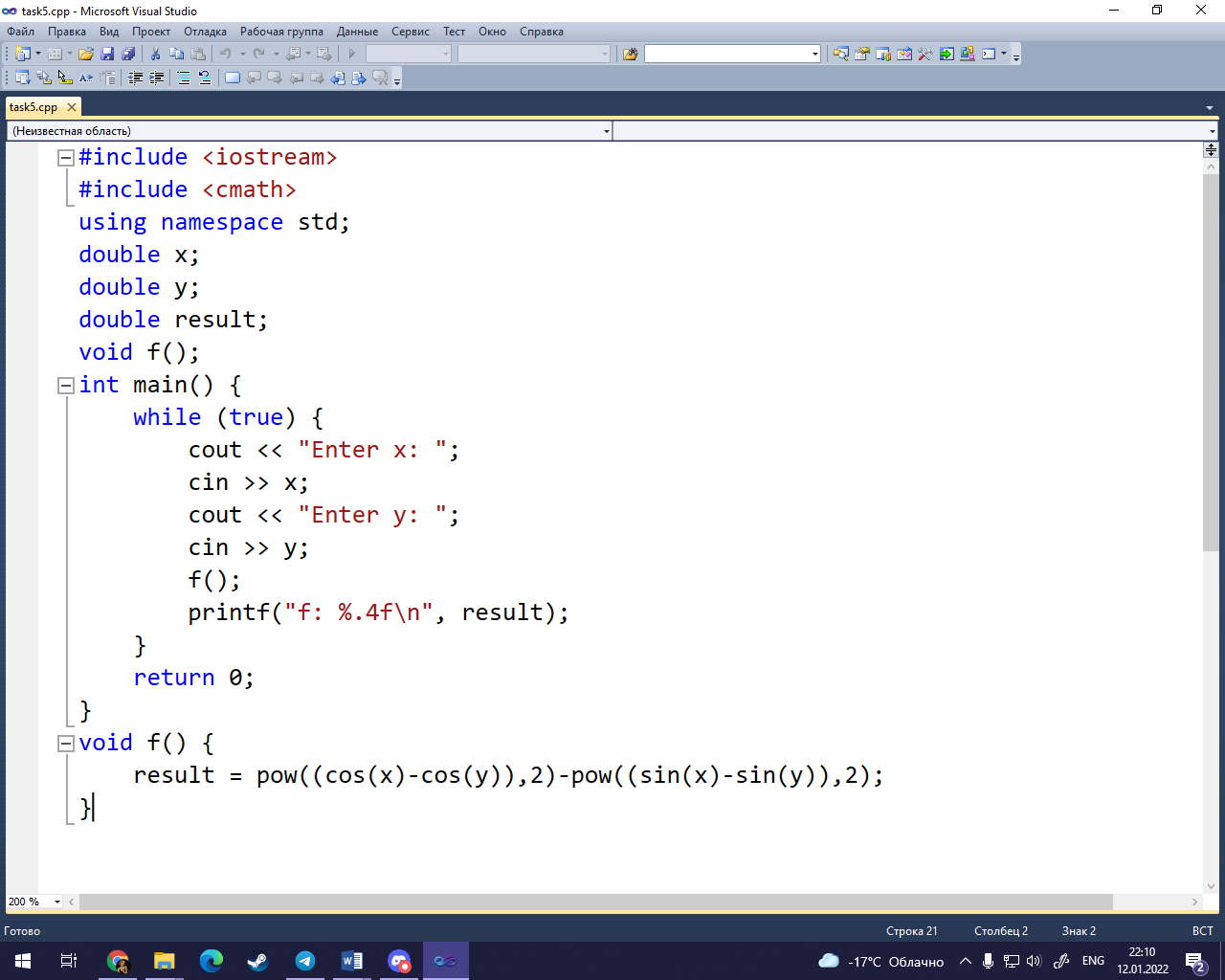
Задание 4:



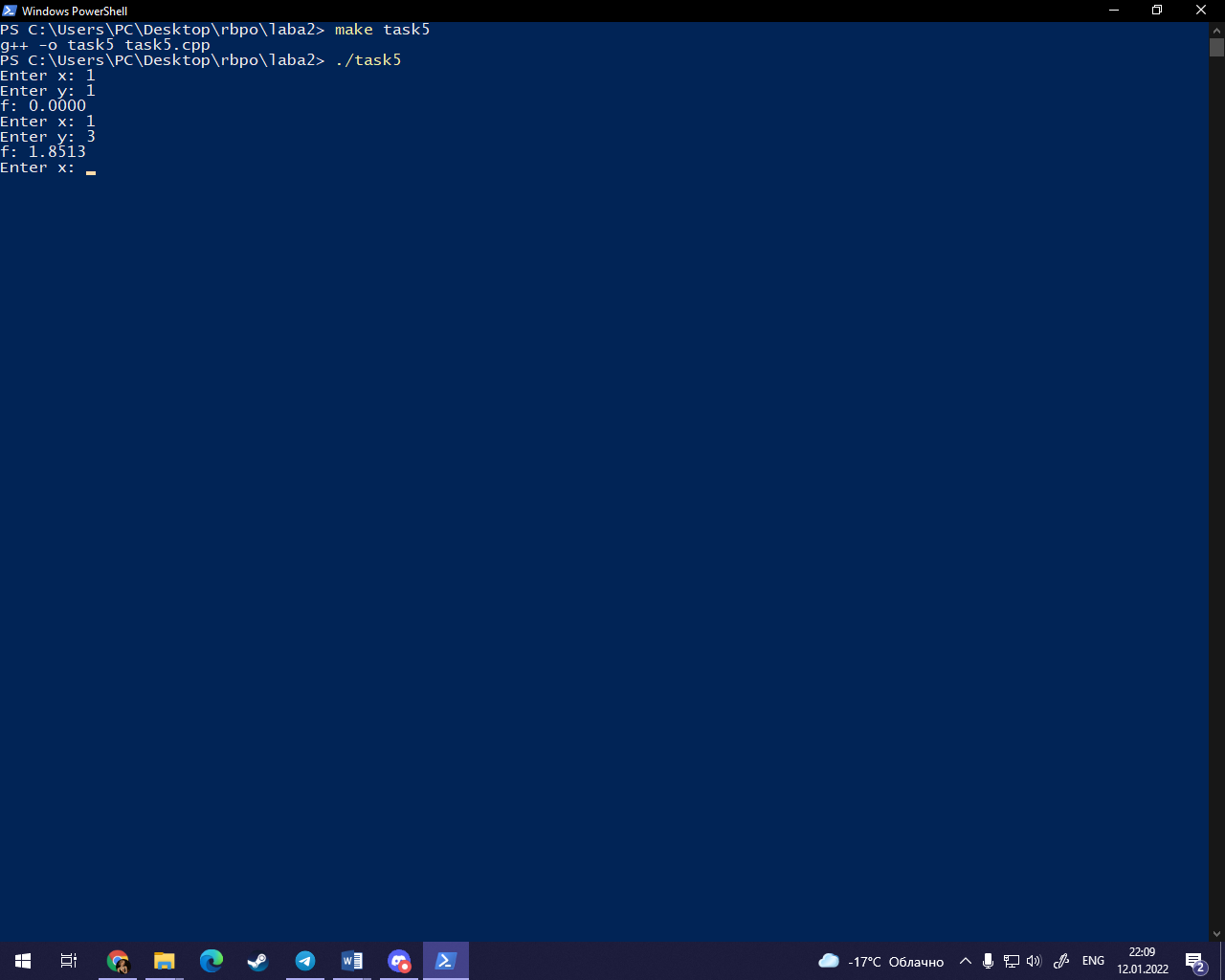
Результат:



Задание 5:

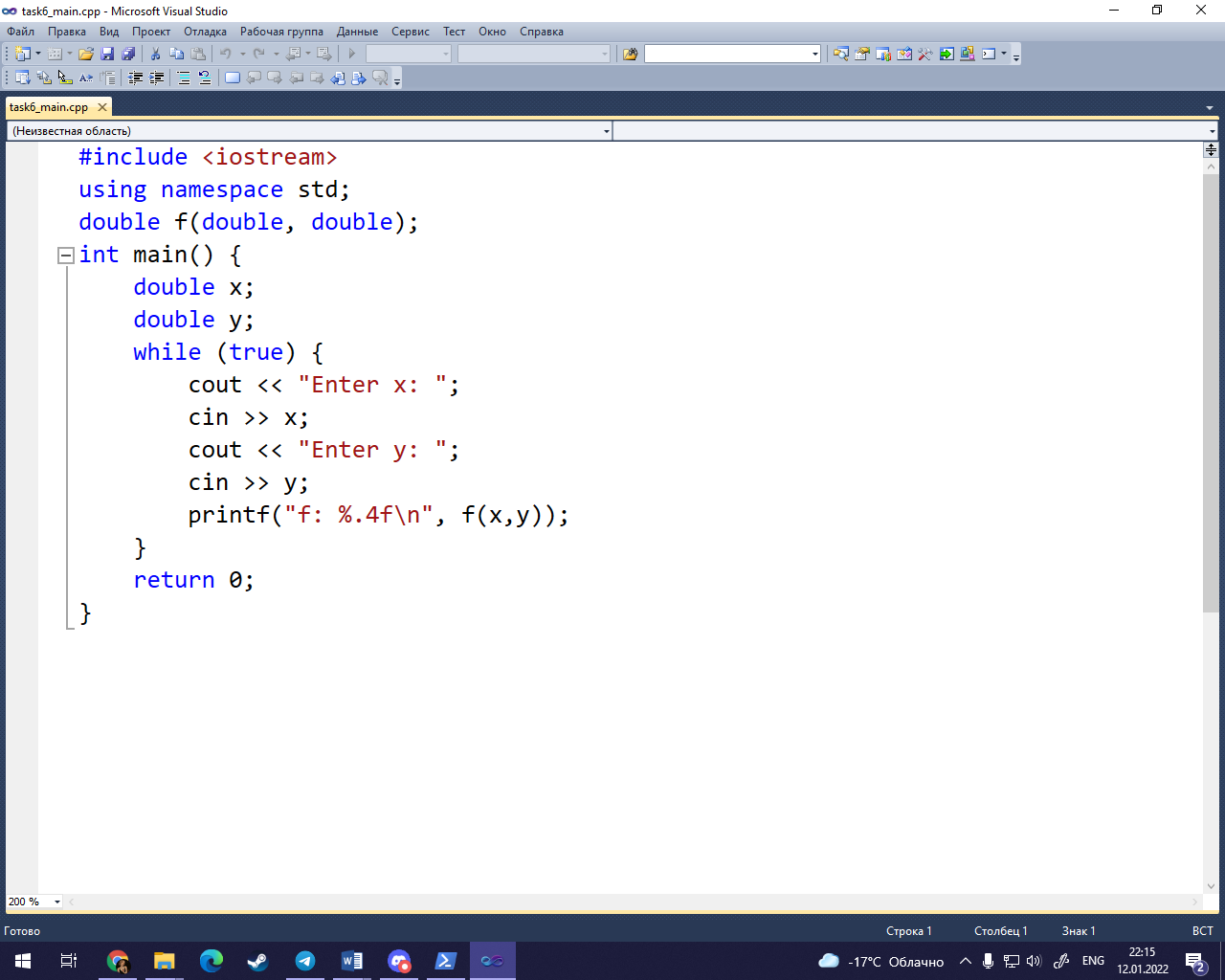


Результат:

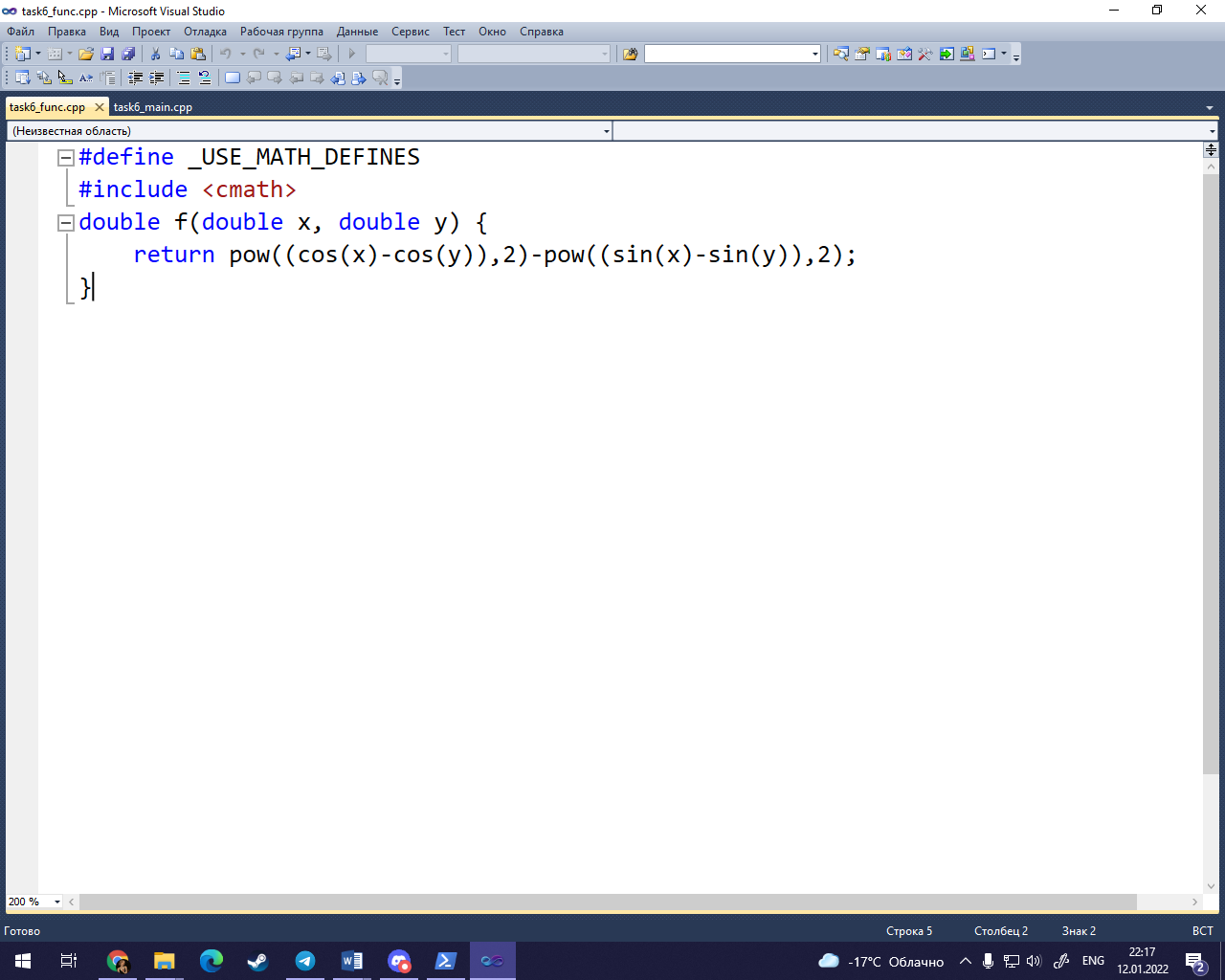


Задание 6:

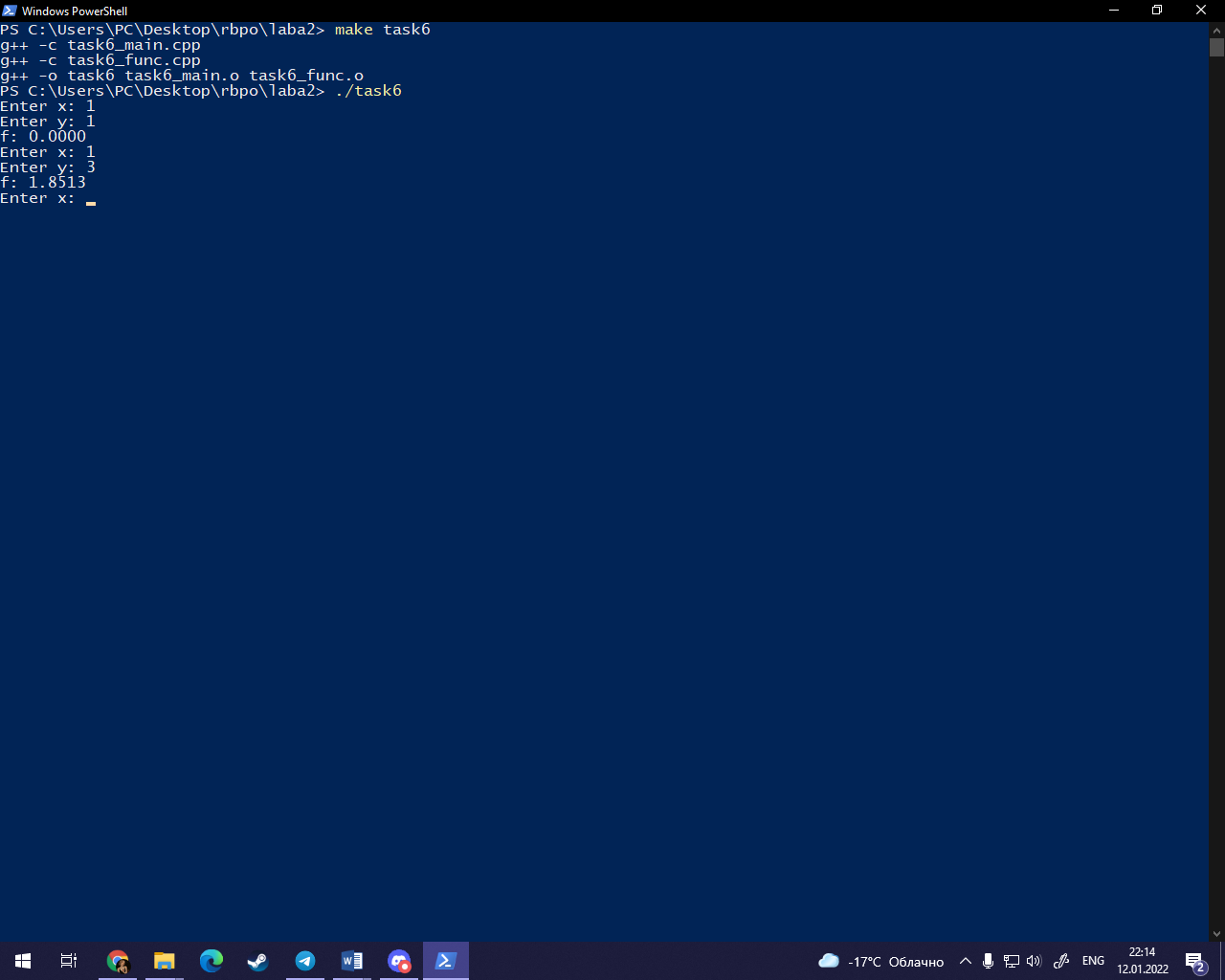
main:



func:

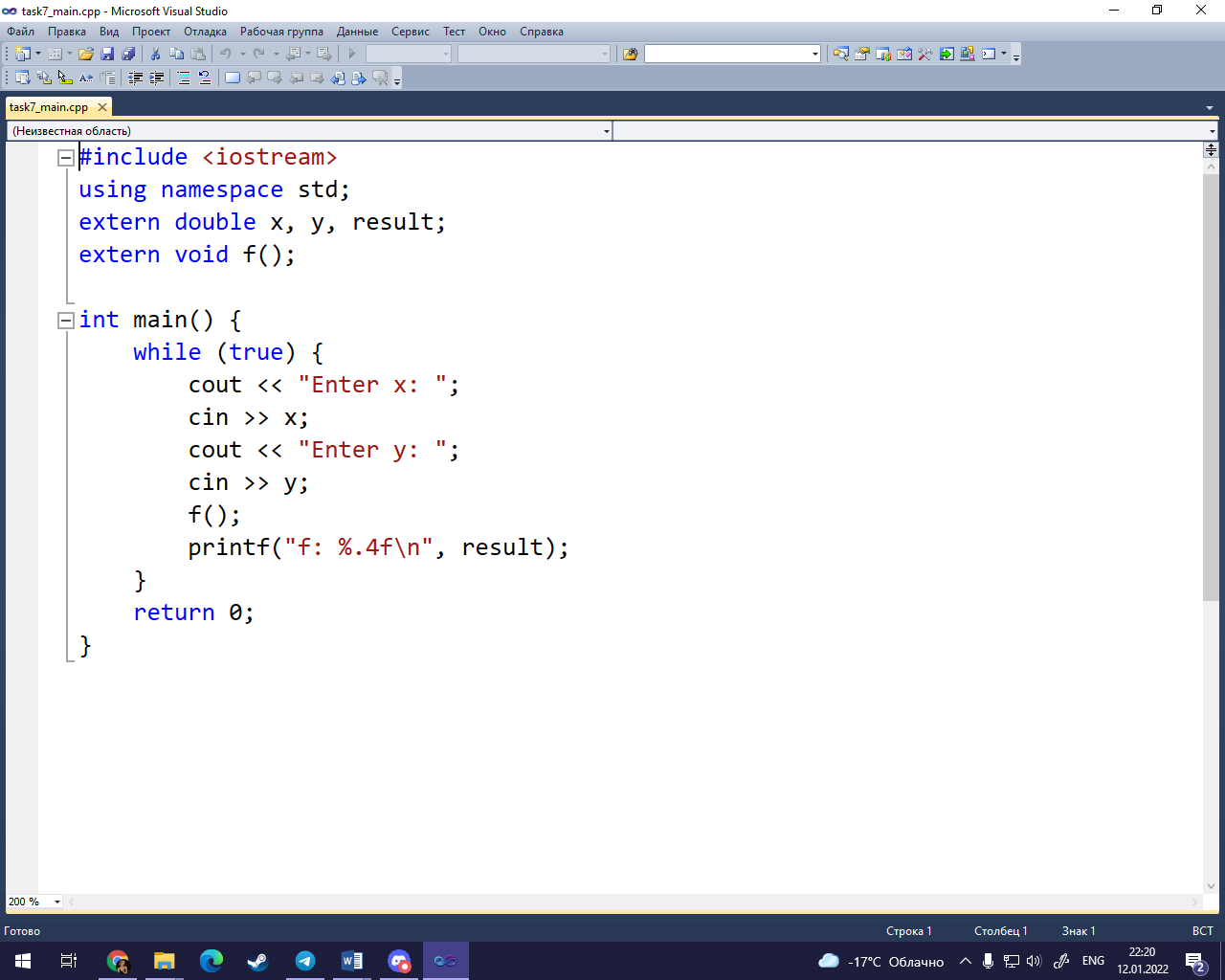


Результат:

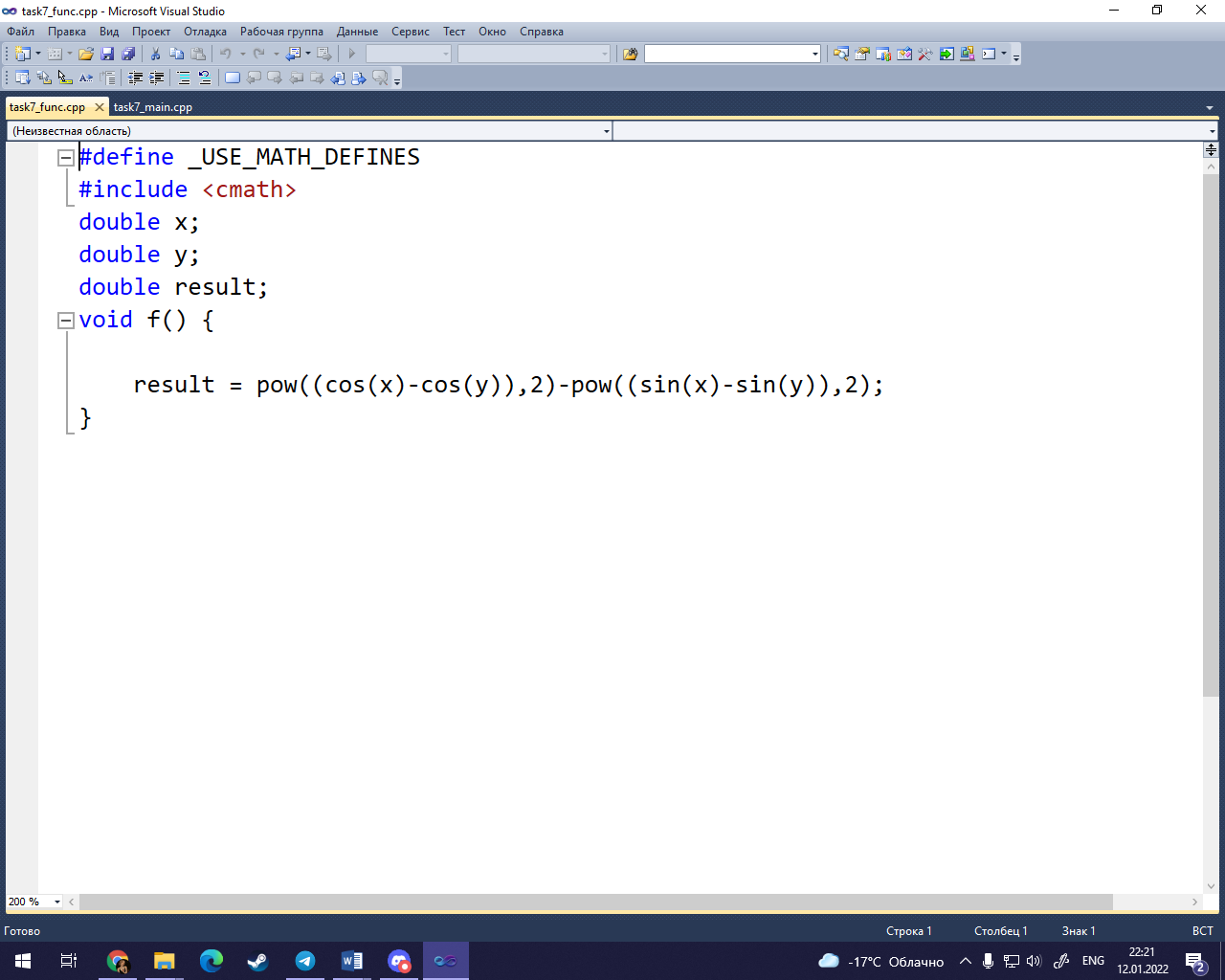


Задание 7:

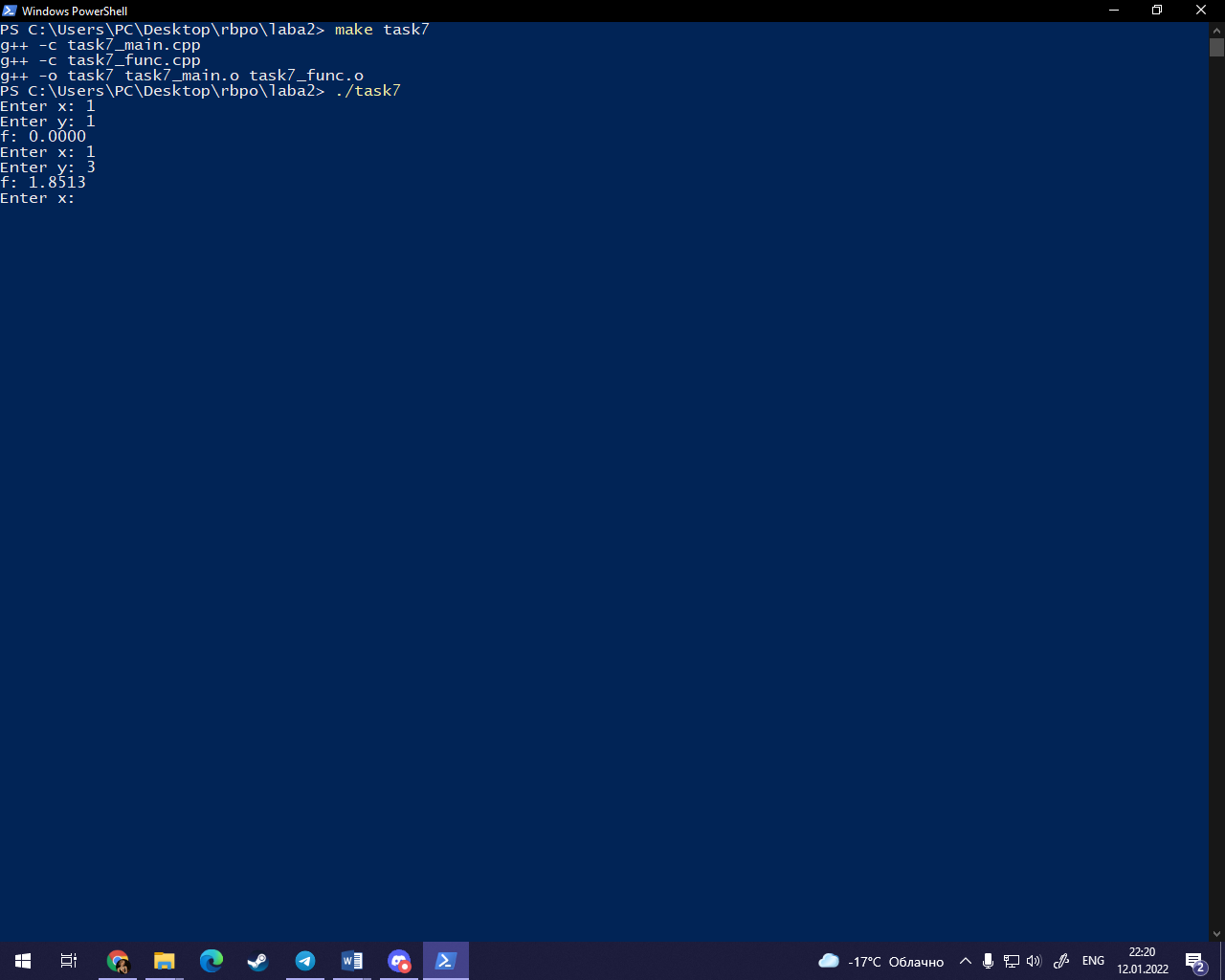
main:



func:

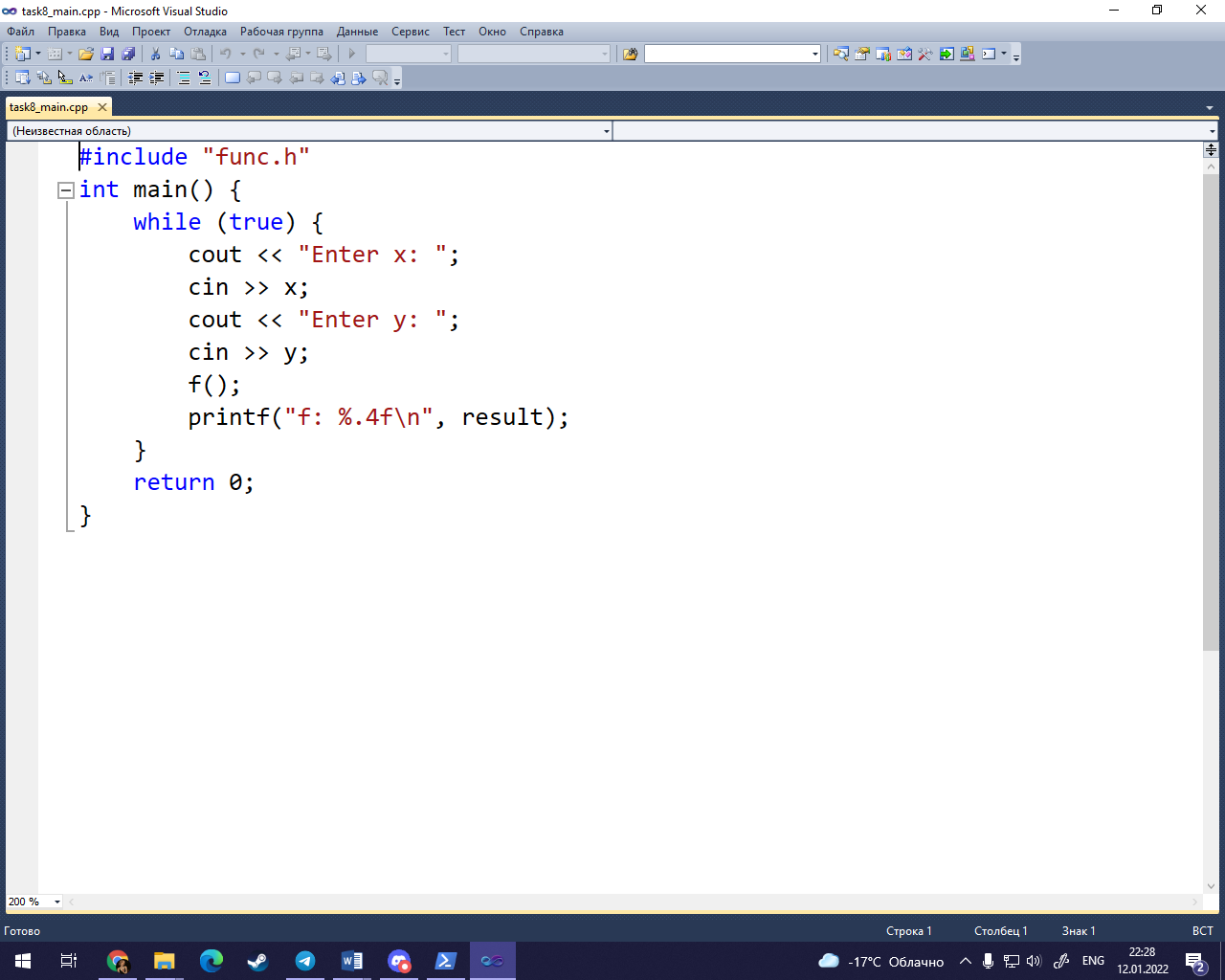


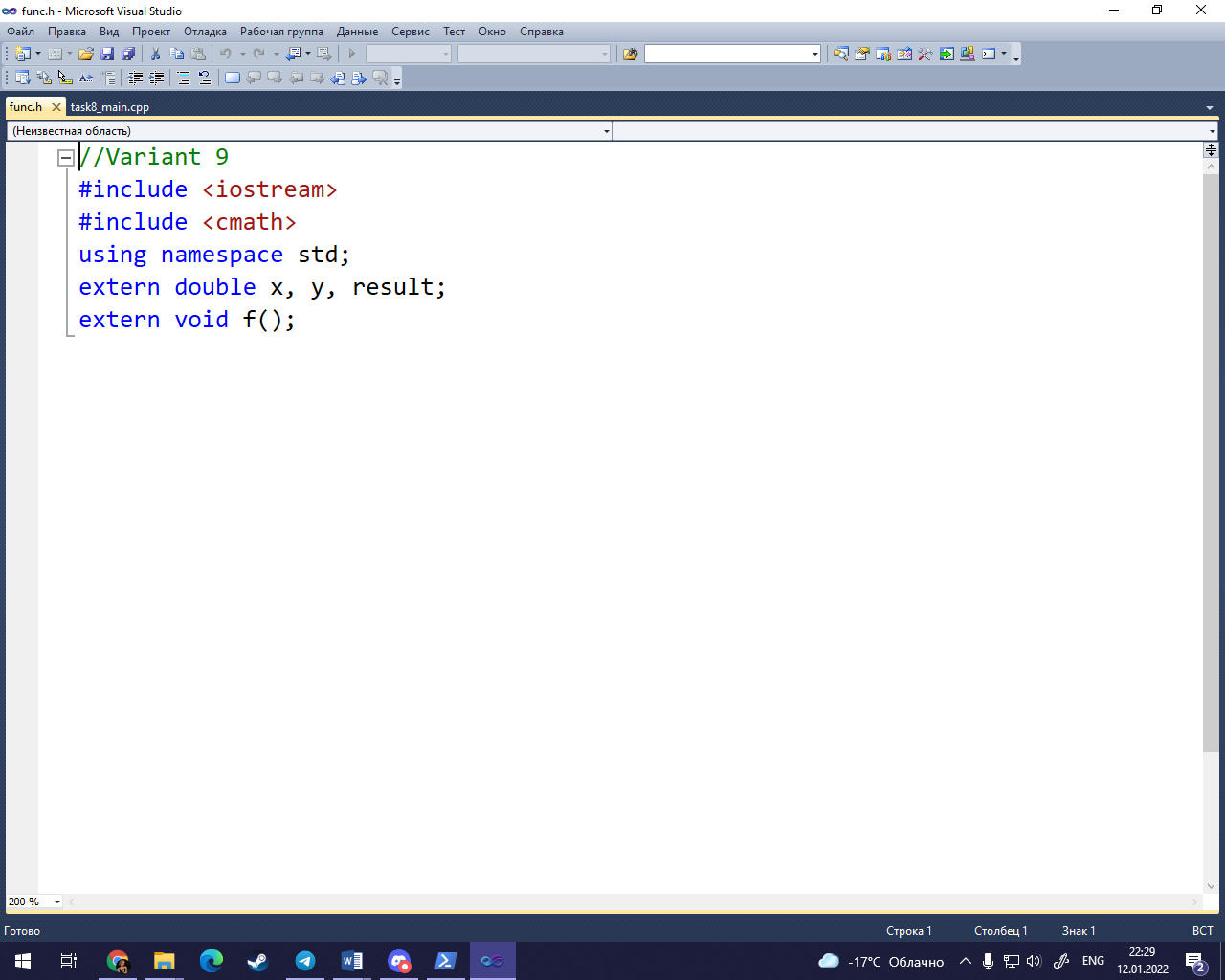
Результат:



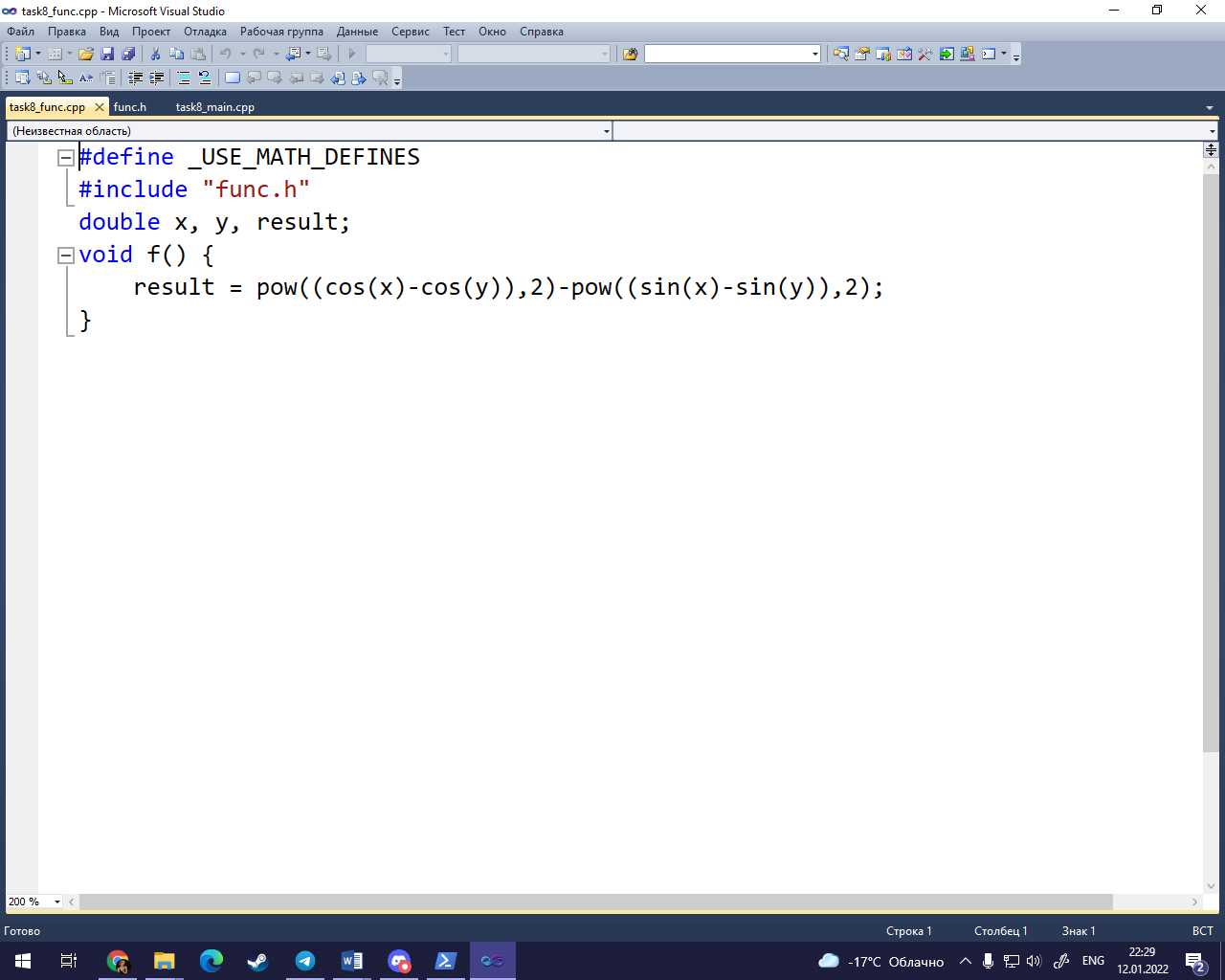
Задание 8:

main:

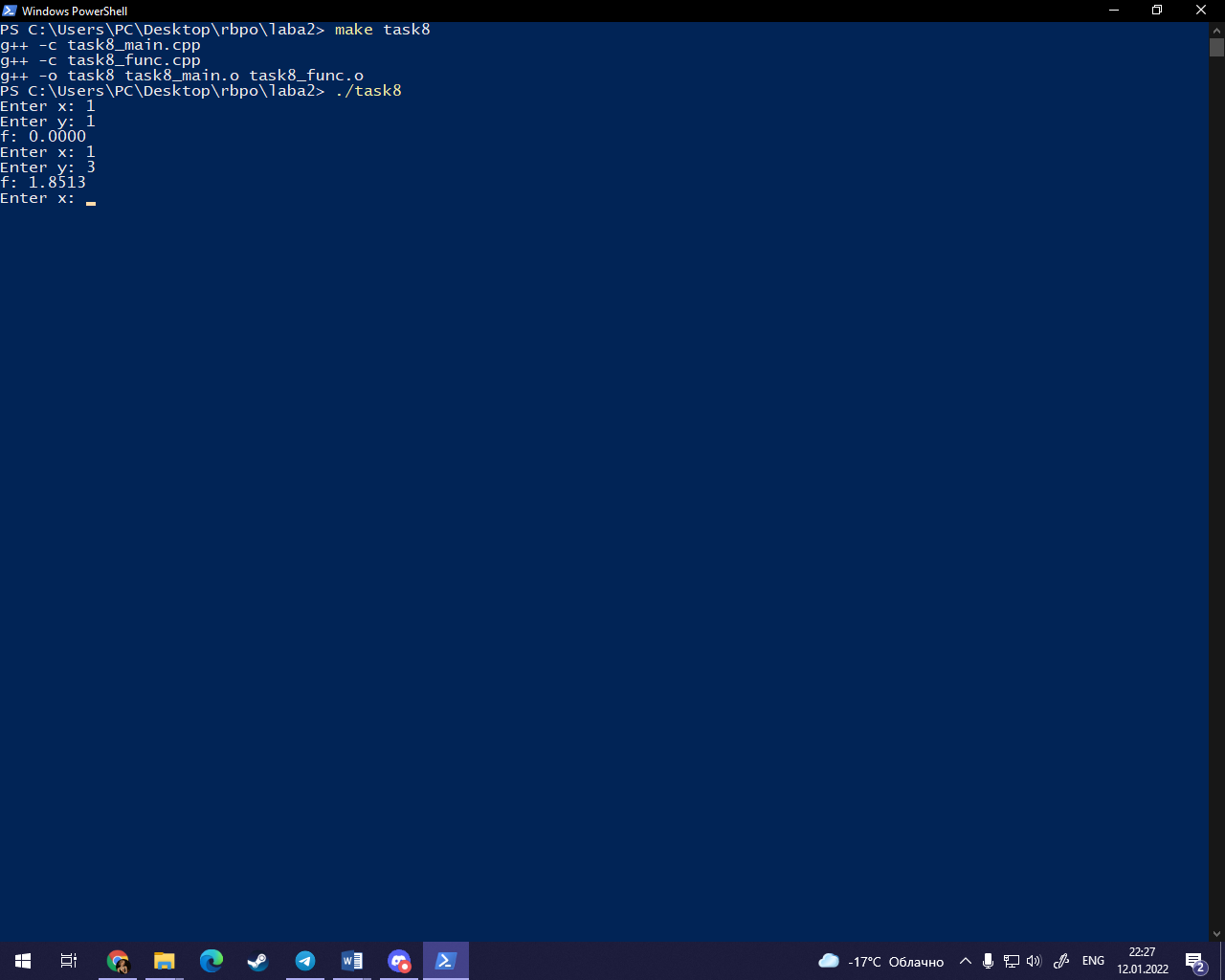




func:

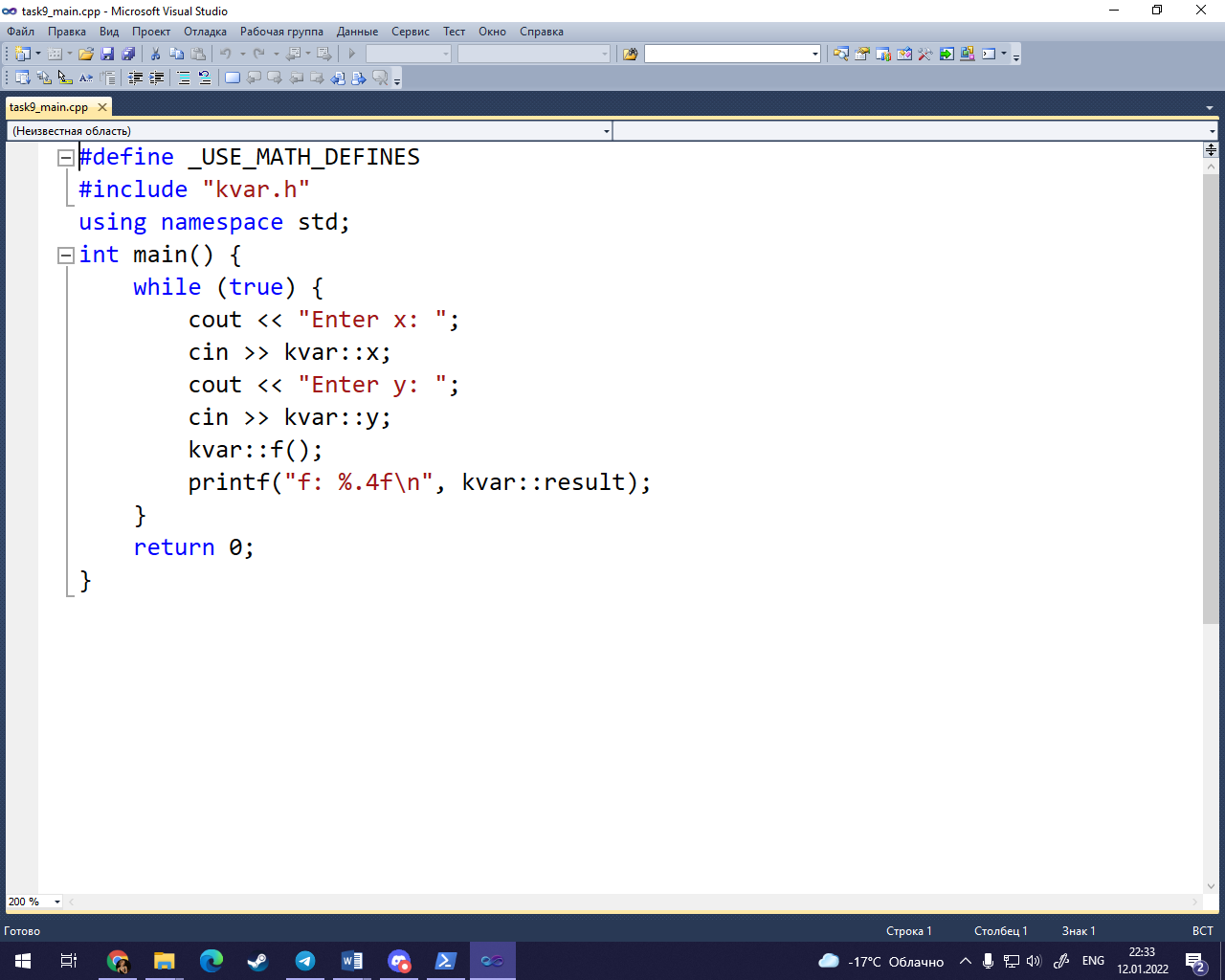


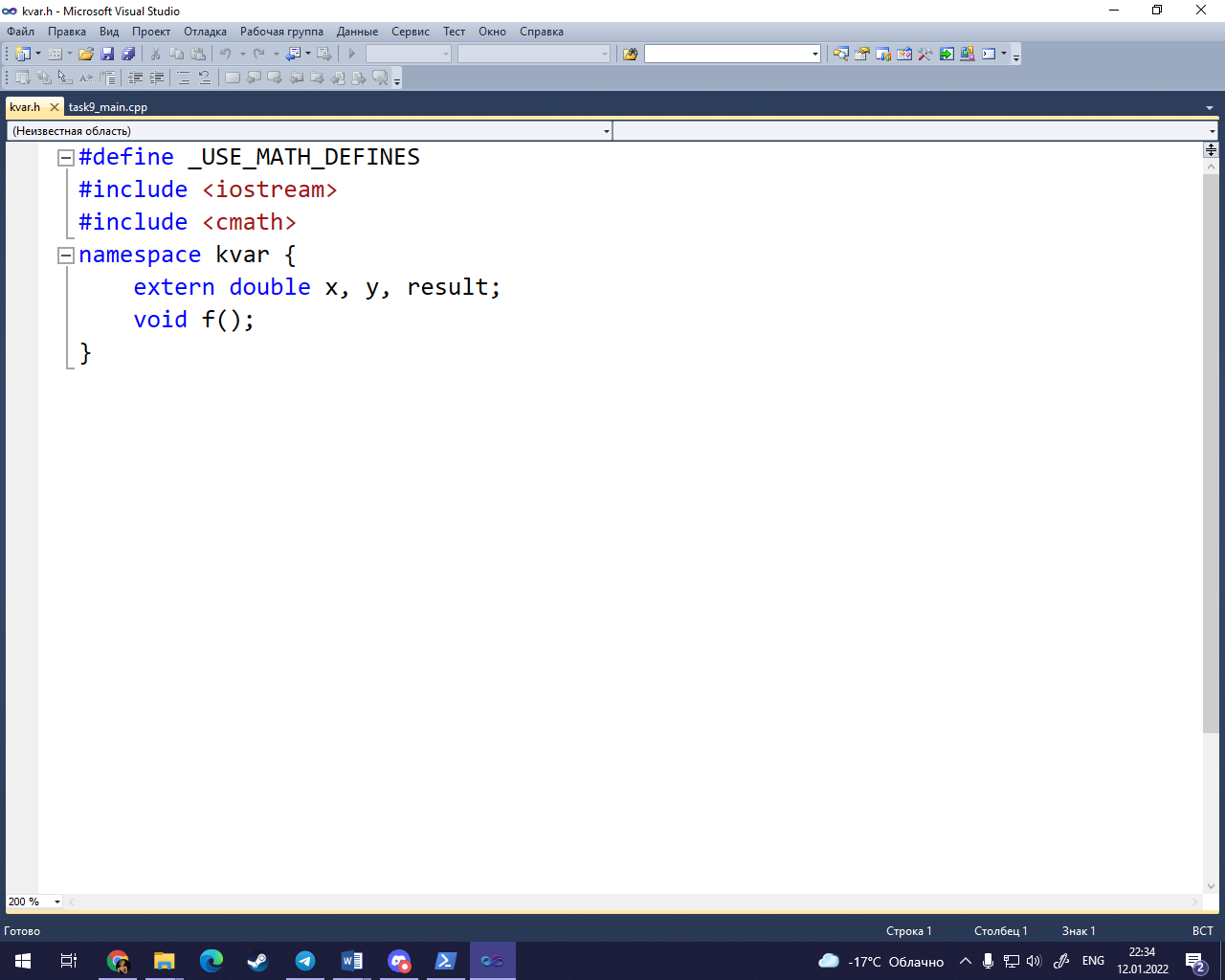
Результат:



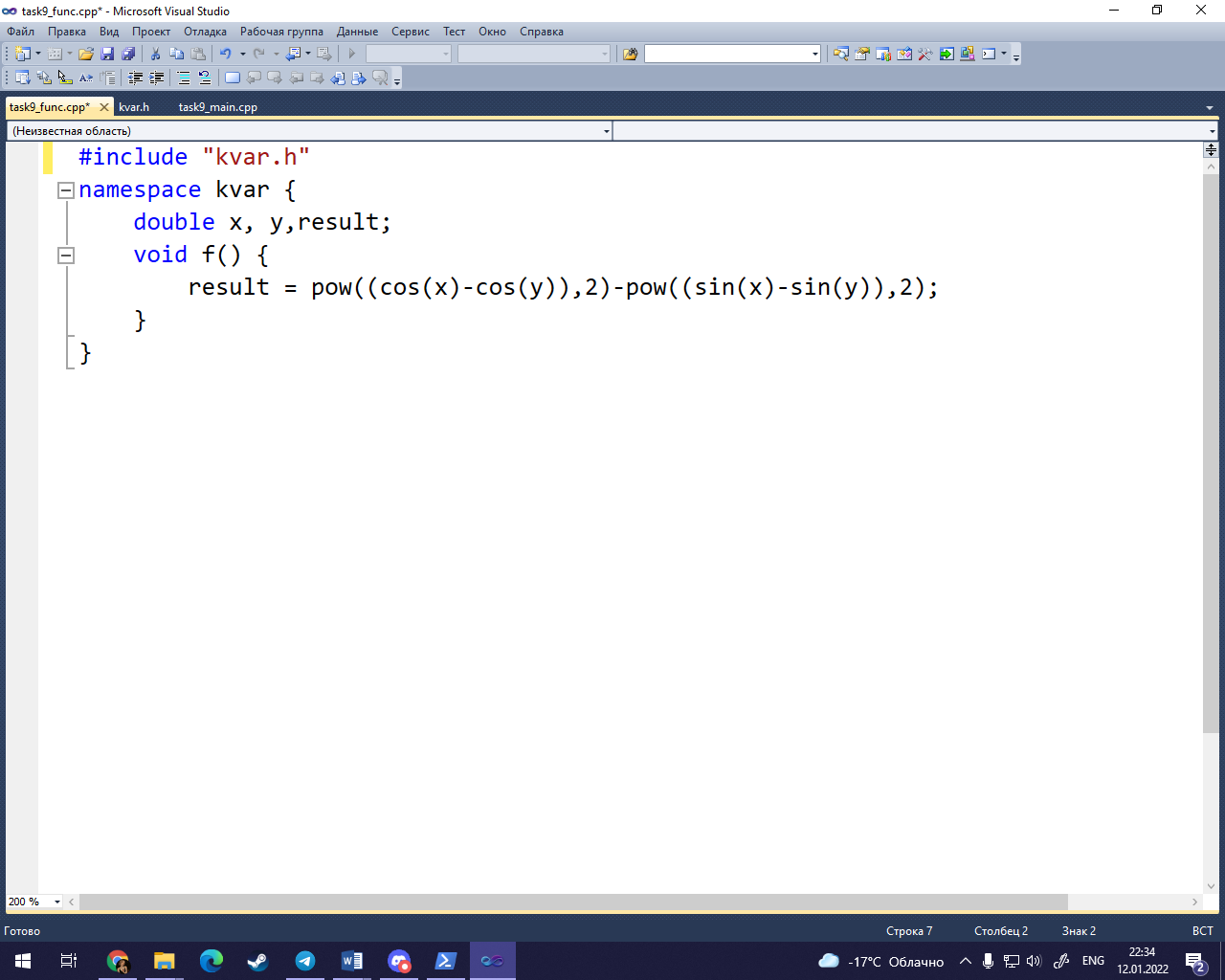
Задание 9:

main:

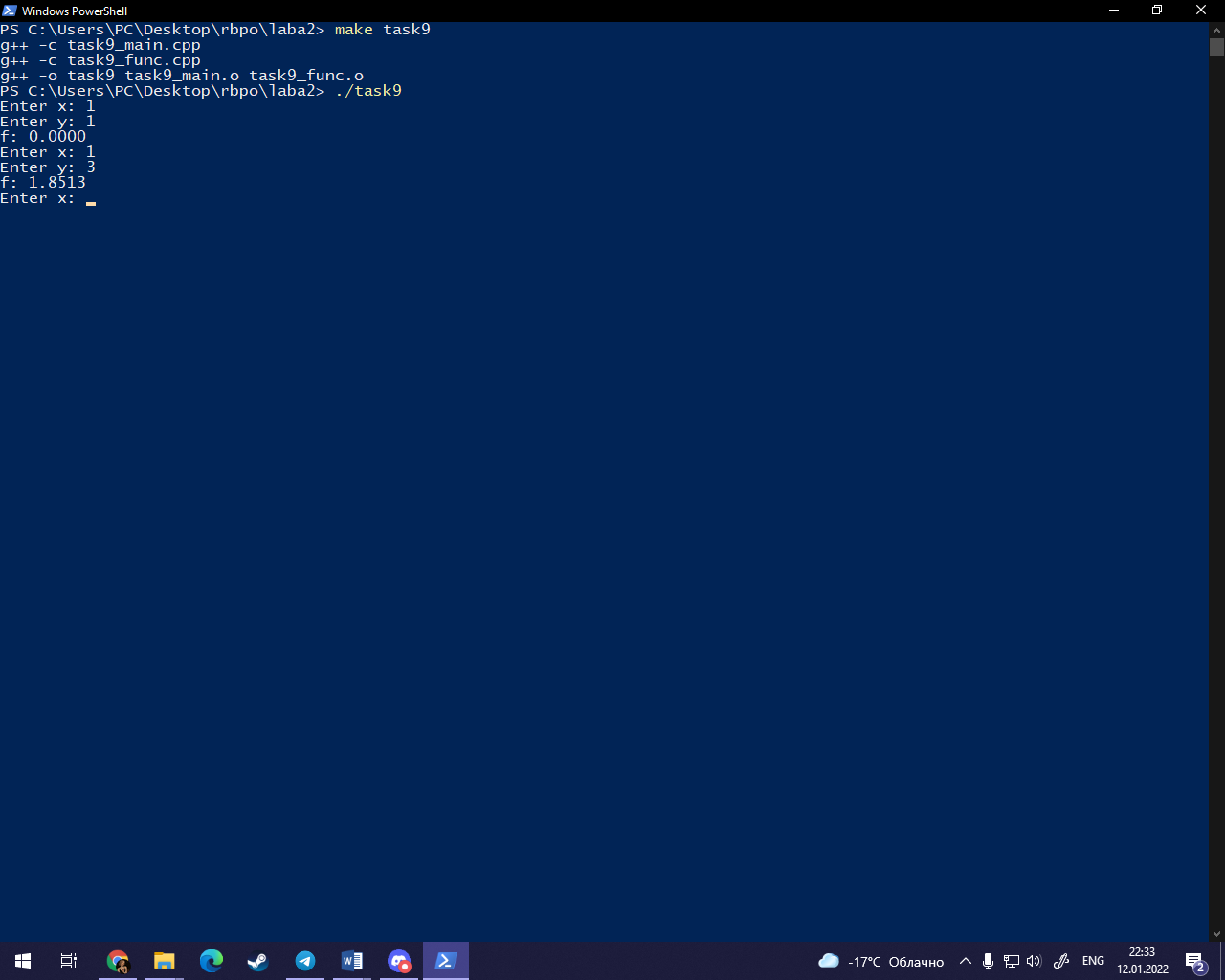




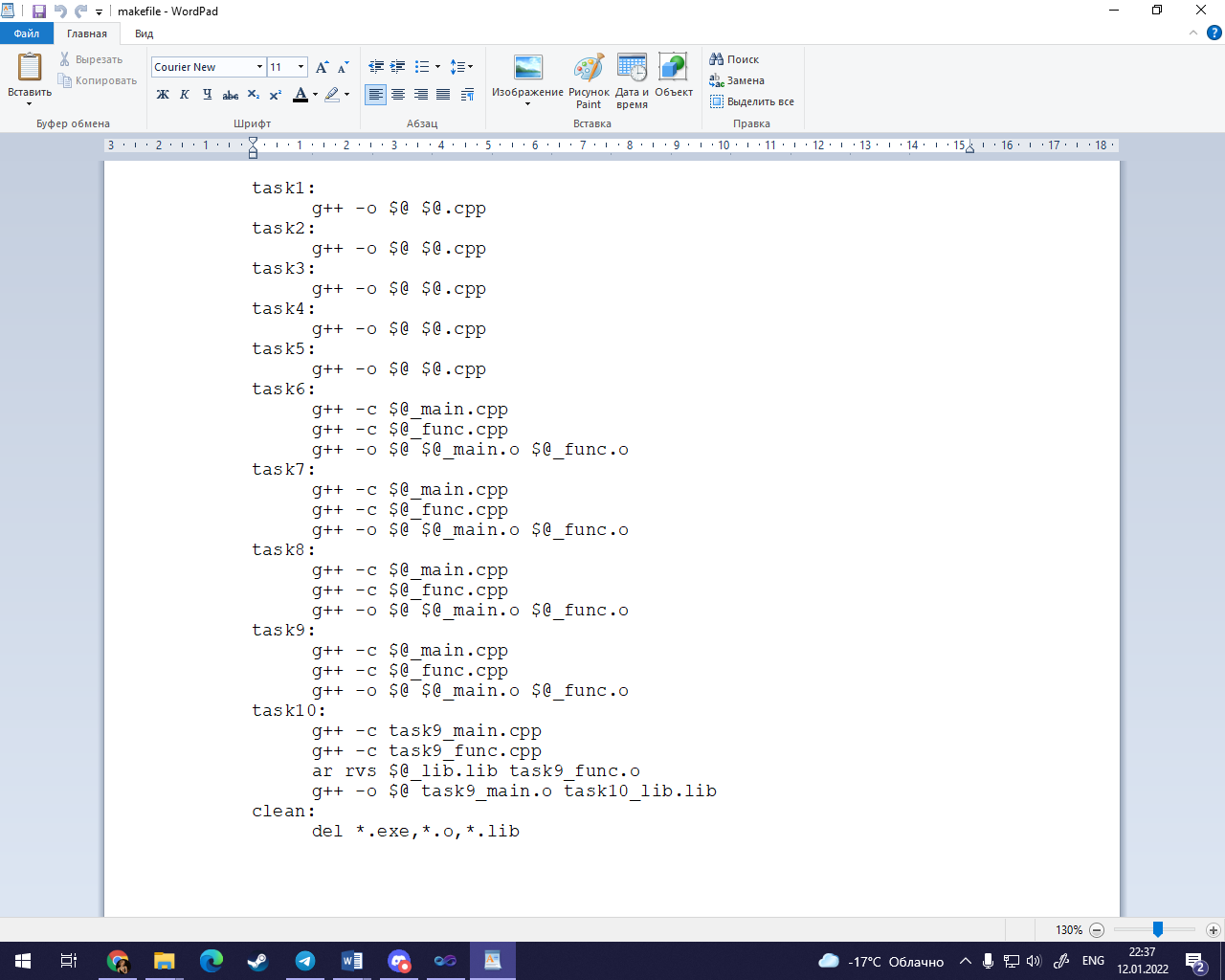
func:



Результат:



Задание 10:



Результат:

