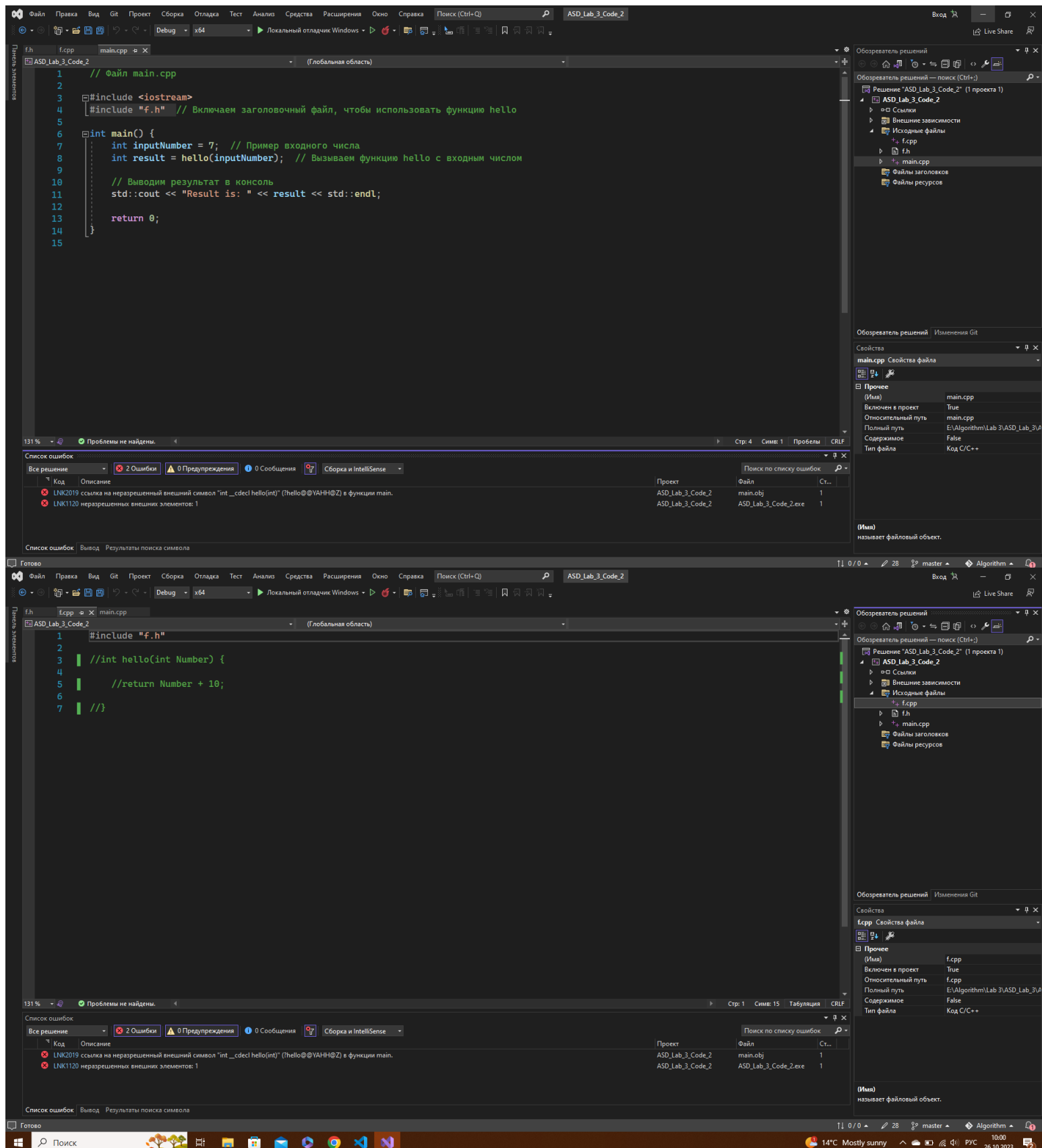


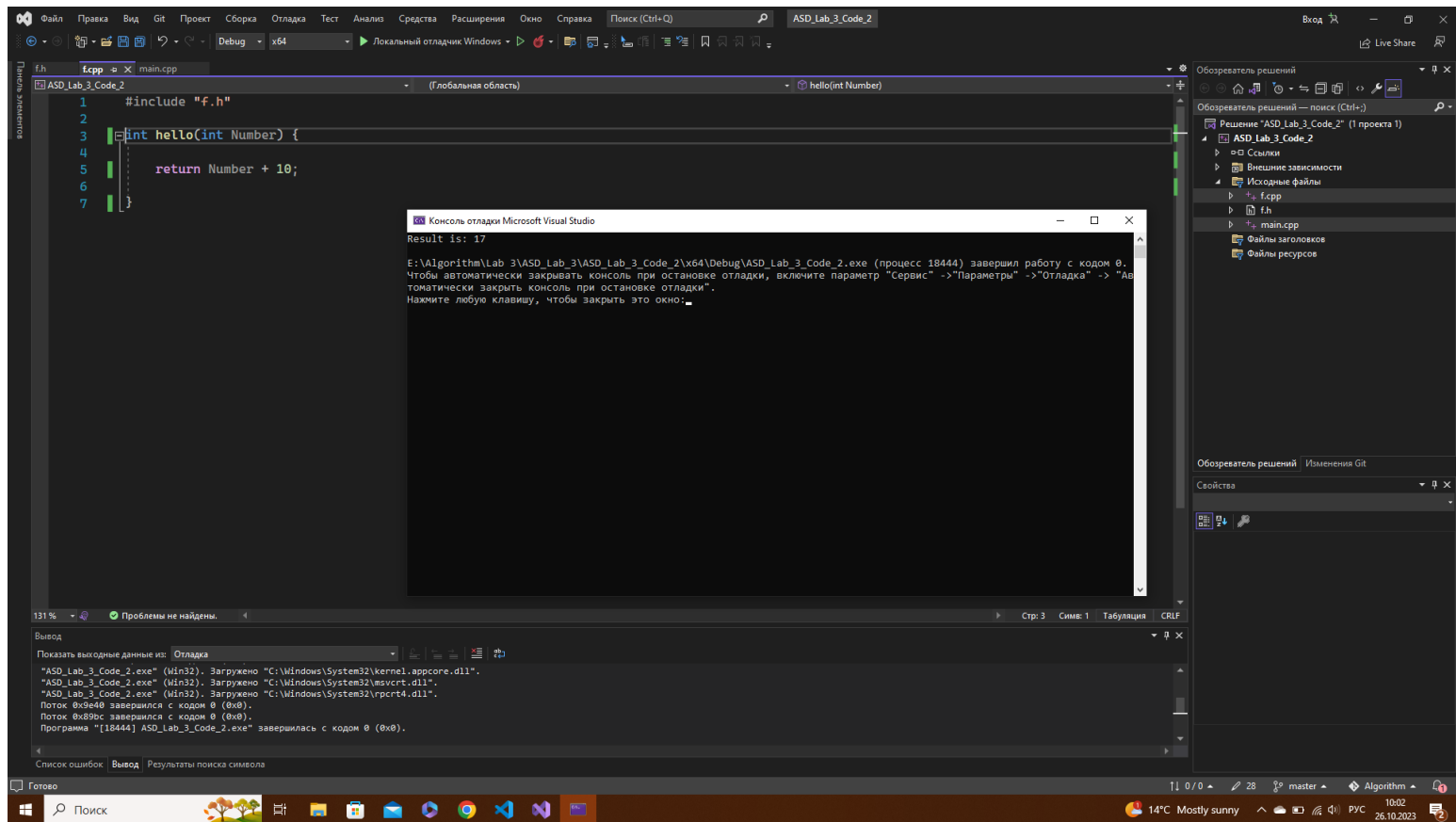
- Пример с 2 файлами:

1. Объявите в новом файле `main.cpp` функцию которая принимает `int` и возвращает `int` с именем `hello`.
2. Вызовите функцию в функции `main` и выведите результат в консоль.
3. Объявите в новом файле `f.cpp` функцию которая принимает `int` и возвращает `int` с именем `hello`.
4. Попробуйте скомпилировать (оба файла разом). Объясните ошибки.



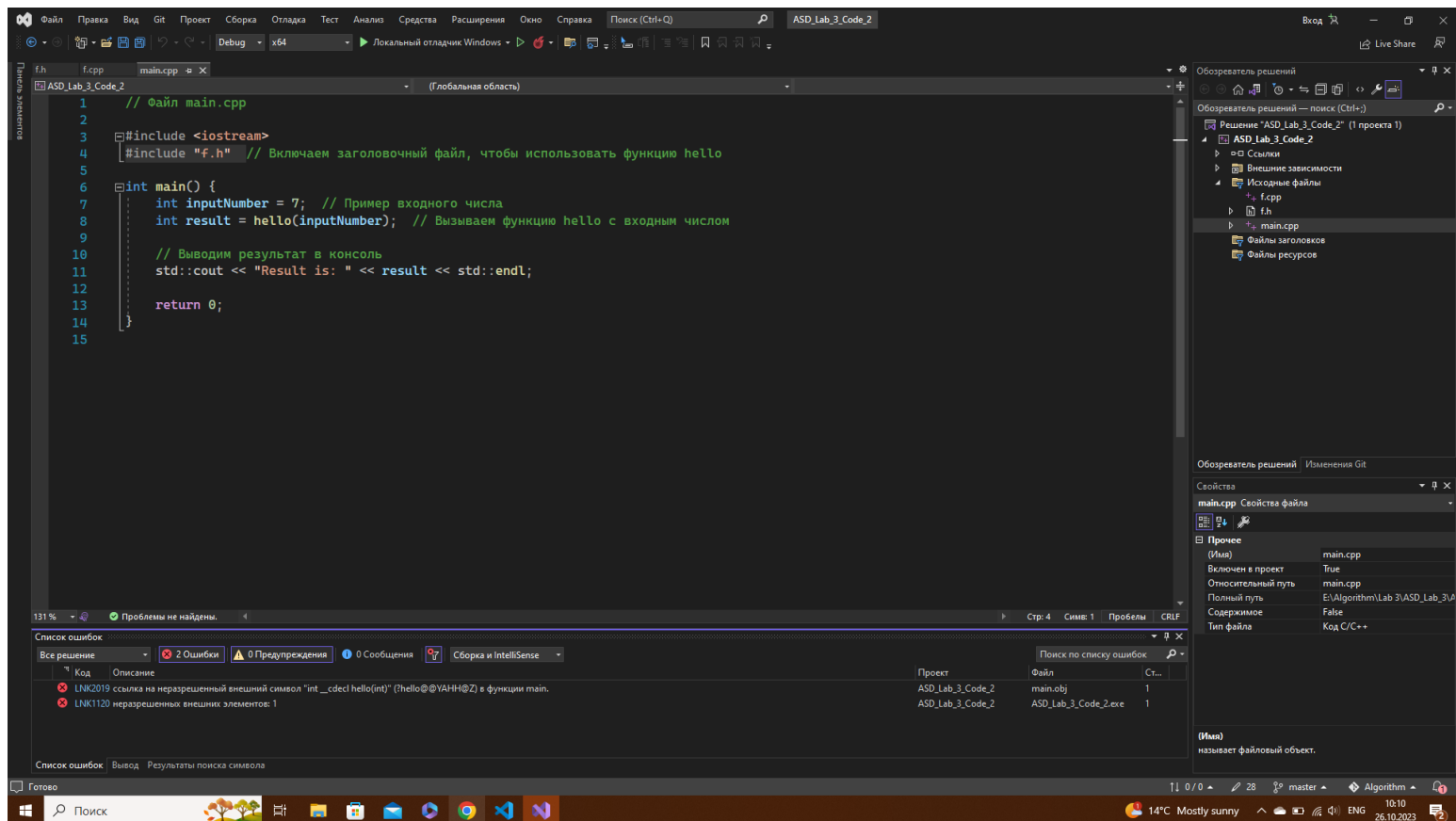
Ошибка в данном случае заключается в том, что компилятор обнаружит, что функция `hello` объявлена, но не определена.

5. Определите функцию `hello` в файле `f.cpp` и попробуйте скомпилировать. (Должно пойти).



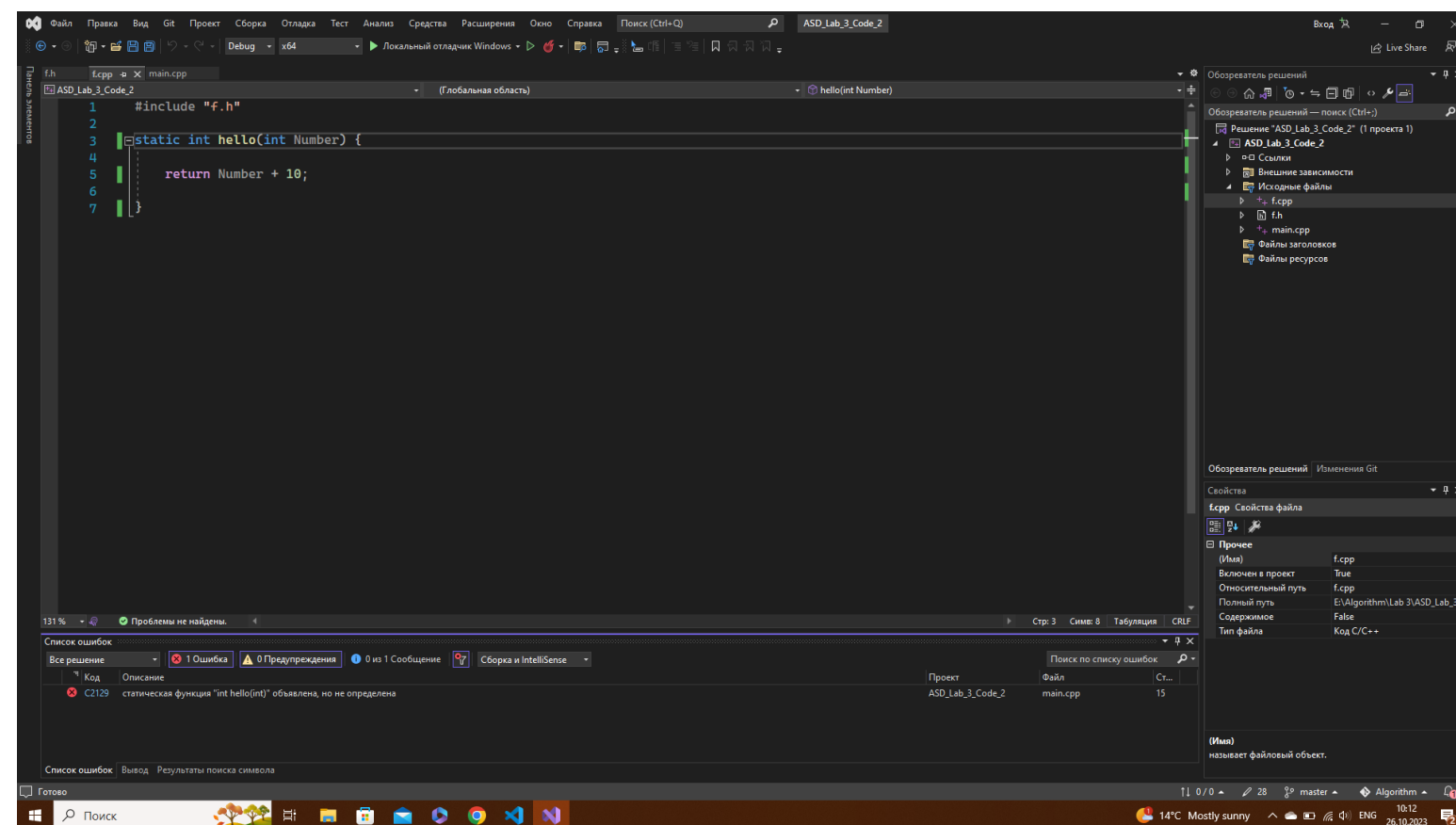
В этот раз код заработал без каких-либо ошибок.

6. Попробуйте скомпилировать только файл main.cpp. Объясните ошибки.



В данном случае ошибка заключается в том, что функция `hello` теперь определена в файле `f.cpp`, но не видна в `main.cpp`. На английском ошибка будет звучать так: `undefined reference to 'hello(int)'`.

7. Добавьте к определению функции модификатор `static`.

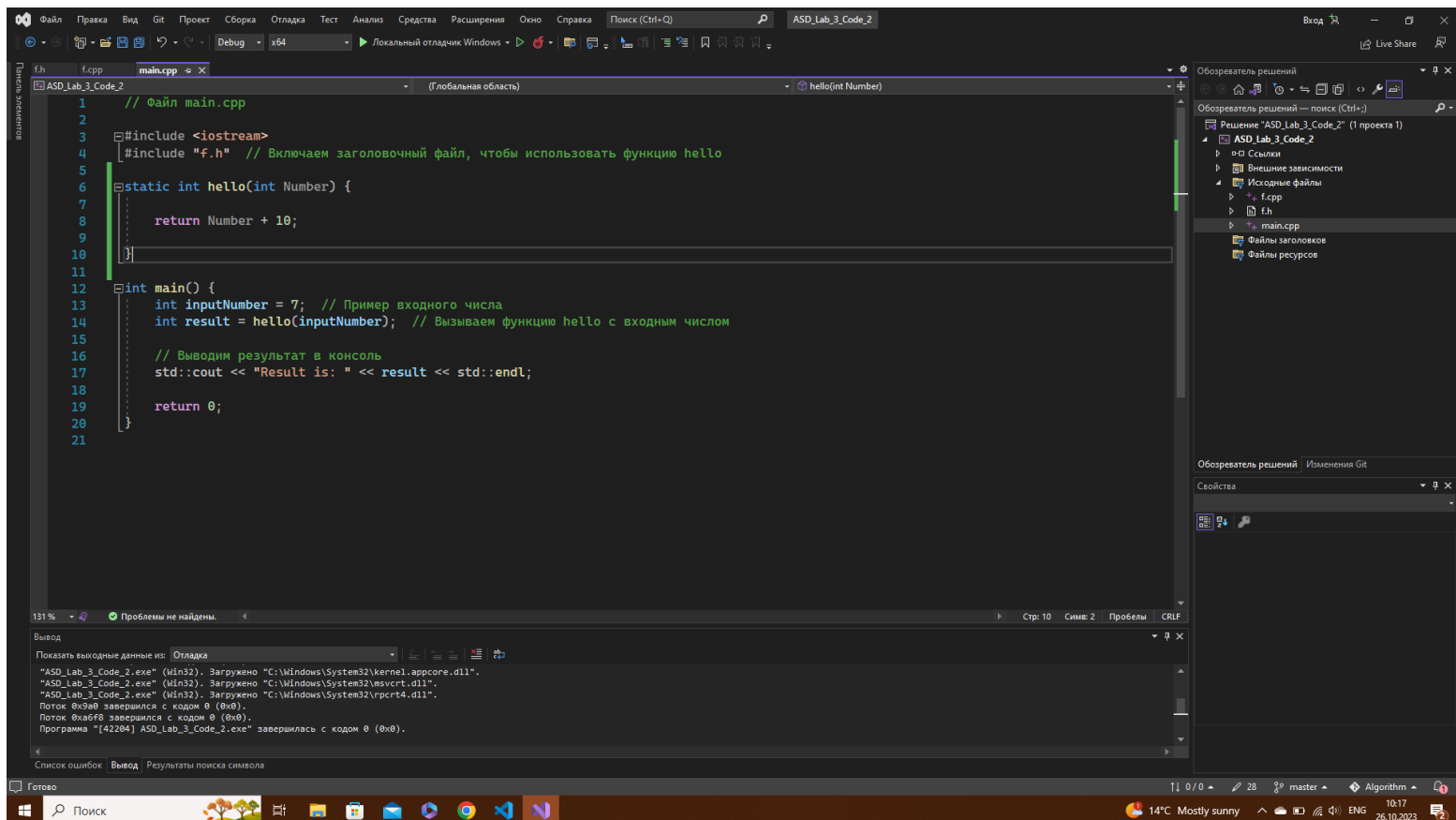


8. Попробуйте скомпилировать. Объясните ошибки.

При попытке компиляции мы получаем ошибку, практически аналогичную такой, как в 4 пункте. Здесь объявлена статическая функция, т.е. `static int hello(int)`, но она не определена.

9. Перекопируйте определение функции в `main.cpp` (включая `static`).

10. Попробуйте скомпилировать. Объясните почему работает.



Сейчас программа работает, потому что функция `hello` определена локально в `"main.cpp"` с модификатором `static`, и компилятор не будет искать её в других файлах.

11. Объясните сколько копий тела функции `hello` скомпилируется в финальный executable (можете предположить, что компилятор не уберет неиспользованные копии).

Когда функция `hello` определена с модификатором `static`, в каждом исходном файле, где она используется, создается своя собственная копия этой функции. Поэтому, если вы использовали функцию `hello` в нескольких файлах, каждый файл будет содержать свою собственную копию этой функции в итоговом исполняемом файле. Это может привести к увеличению размера исполняемого файла, если функция использовалась в нескольких местах.