

## Лабораторна робота №1

### Робота з IDE Microsoft Visual Studio.

#### Програмування алгоритмів лінійної структури, компіляція та відлагодження програми

**Мета роботи:** Вивчення основних прийомів роботи з інтегрованим середовищем розробки *Microsoft Visual Studio*. Написання найпростішої програми на мові C, отримання навичок компіляції та відлагодження програми.

#### Короткі теоретичні відомості

*Microsoft Visual Studio* – лінійка продуктів компанії Майкрософт, що включають інтегроване середовище розробки програмного забезпечення та ряд інших інструментальних засобів.

#### Інсталяція IDE Microsoft Visual Studio

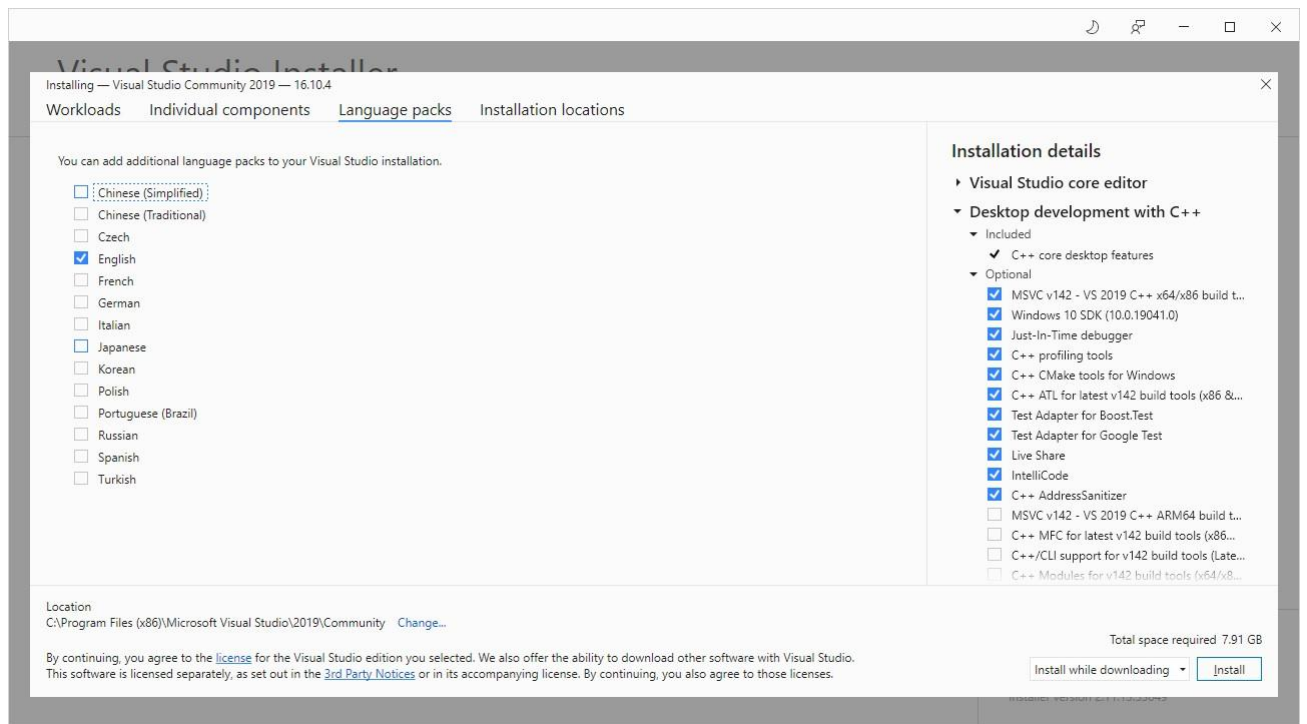
Для установки *Microsoft Visual Studio* з належними засобами потрібно близько 8 ГБ дискового простору. Для скачування інсталяційного файлу із *IDE Microsoft Visual Studio* потрібно перейти за посиланням:

<https://visualstudio.microsoft.com>

У вікні що відкрилось потрібно обрати першу опцію *Download Visual Studio* (не *Visual Studio Code*!):

*IDE Microsoft Visual Studio* – це повнофункціональне та безкоштовне для студентів, ентузіастів інтегроване середовище розробки для створення сучасних додатків для *Android*, *iOS* та *Windows*, а також веб-додатків та хмарних служб.

Для розробки програм на C++ при інсталяції *IDE* слід виконати такі налаштування:



Після завершення установки цього програмного забезпечення потрібно буде перезавантажити комп'ютер.

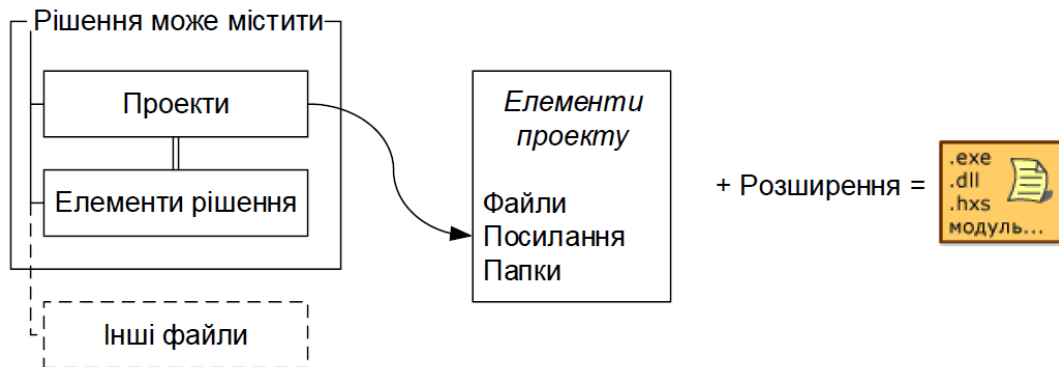
## Можливості та засоби розробки додатків Microsoft Visual Studio

У сімействі продуктів *Visual Studio* використовується загальне інтегроване середовище розробки, що складається з декількох елементів: панелі інструментів, різних закріплених вікон або вікон що автоматично приховуються в лівій, нижній або правій областях, а також області редакторів. Набір доступних вікон інструментів, меню і панелей інструментів залежить від типу проекту або файлу, в якому виконується розробка.

## Рішення та проекти як контейнери

У *Visual Studio* реалізовані контейнери: рішення та проекти, які роблять можливим використання в інтегрованому середовищі розробки (*IDE*) всього діапазону засобів, конструкторів, шаблонів і параметрів. Також, *Visual Studio* надає папки рішень для того, щоб структурувати пов'язані проекти по групах і потім виконувати дії над цими групами проектів.

Проект – це група файлів і налаштувань, з яких збирається остаточна програма або вихідні файли. Проект включає набір файлів вихідних текстів та метаданих, наприклад посилання на компоненти та інструкції побудови. Як правило, при побудові проектів створюється один або кілька файлів із кодом програми. Рішення включає один або декілька проектів, а також файли і метадані, необхідні для опису рішення в цілому:



Щоб допомогти користувачам організувати і виконувати стандартні завдання із застосуванням елементів, що розробляються, проекти *Visual Studio* використовуються як контейнери в межах рішення. Це дозволяє логічно управляти, виконувати побудову і налагоджувати елементи, що утворюють програму. На виході, як правило, ми отримуємо програму що виконується (*EXE*), файл бібліотеки динамічного компонування (*DLL*) або модуль.

Проект може бути простим або складним залежно від конкретних вимог. Простий проект може містити файли коду програми та файл проекту. Більш складні проекти можуть включати ці ж елементи і, крім того, сценарії баз даних, збережені процедури та посилання на існуючі *XML* (веб-служби).

## Шаблони проектів

*Visual Studio* надає шаблони для проектів найбільш поширених типів.

Використання проектів і їх шаблонів дозволяє користувачеві зосередитися на реалізації окремої функції, в той час як типовий проект обраного типу буде згенерований автоматично.

## Файли проектів

Кожен шаблон створює файл проекту, в якому містяться метадані поточного проекту. Файл проекту містить налаштування проекту, а також, можливо, список файлів проекту та їх розташування.

При додаванні файлу в проект інформація про його фізичне місце розташування заноситься в файл проекту. При видаленні такого зв'язку, ця інформація видаляється з файлу проекту. Шаблон проекту визначає, які команди доступні для кожного його елемента.

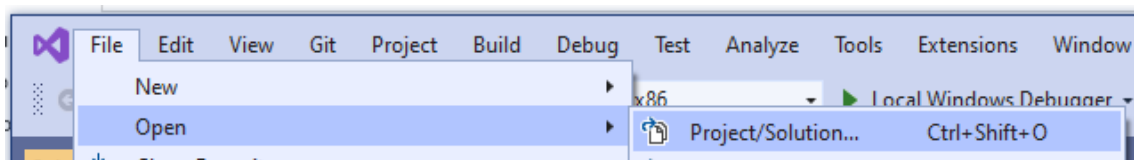
Майстер створення програм надає інтерфейс для створення проекту за шаблоном та створення шаблонів для файлів коду програми. Майстер налаштовує структуру програми, основні меню та панелі інструментів, забезпечує включення деяких файлів із заголовками.

## Головна сторінка, відкриття та створення проектів

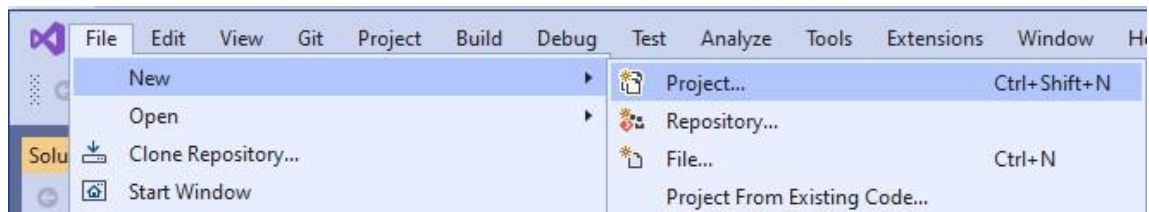
Після запуску *Visual Studio* відкривається стартова сторінка, яка дозволяє отримати легкий доступ до наявних проектів або створити новий проект.

В області, яка розташована ліворуч, можна обрати та відкрити проекти, з якими працювали нещодавно (це область нижче слів *Open recently*). Праворуч, нижче слів *Get started*, розташовані кнопки, які дозволяють відкрити чи створити проекти та рішення.

Для відкриття вже існуючого проекту потрібно або натиснути кнопку *Open a project or solution* стартової сторінки, або обрати опції головного меню *File: Open: Project/Solution...*:



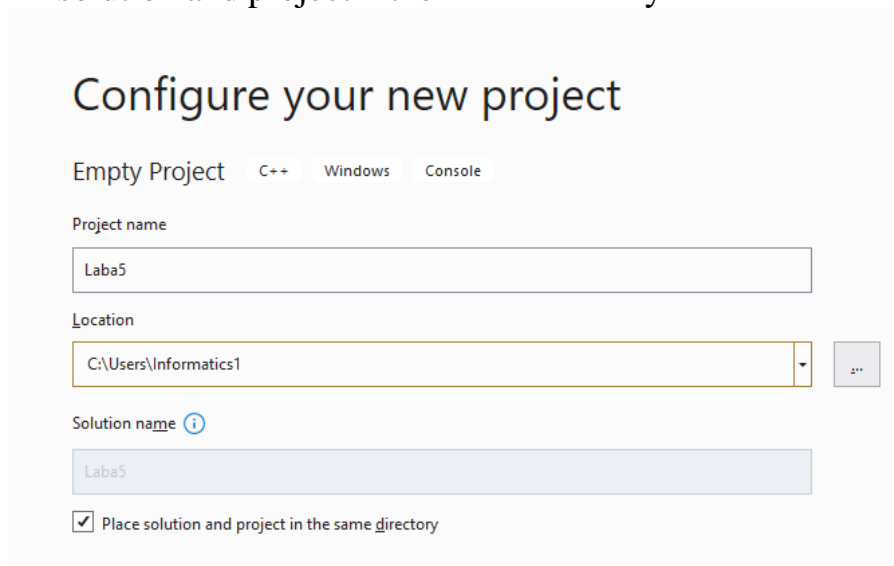
Для створення нового проекту використовується майстер додатків. Для його відкриття потрібно або натиснути кнопку *Create a new project* стартової сторінки, або обрати опції головного меню *File: New: Project...*:



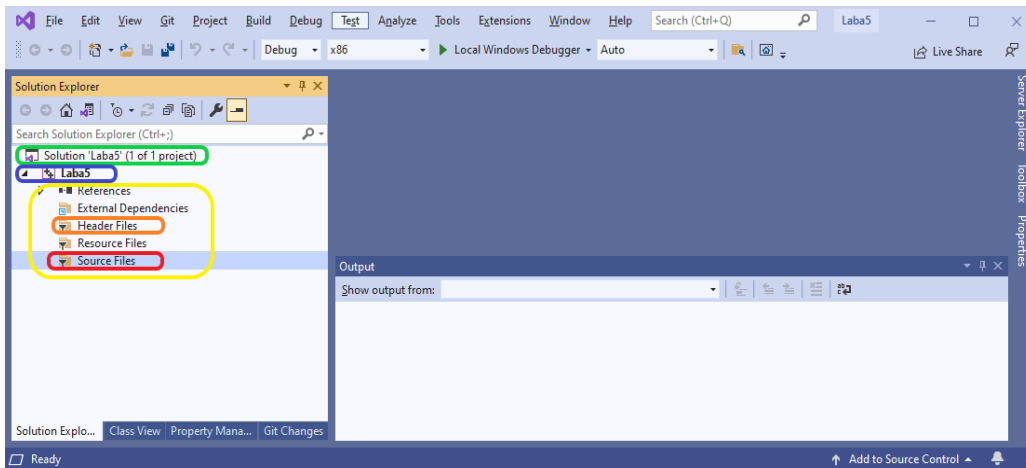
В діалоговому вікні можна задати ім'я, місце зберігання і шаблон нового проекту.

Щоб створити порожній проект, використовуйте шаблон *Empty Project*:

У вікні налаштування параметрів проекту (*Configure your new project*) потрібновказати ім'я проекту (поле *Project name*) та місце його зберігання (поле *Location*). Найчастіше ми будемо створювати рушення, які складаються з одного проекту, тому для спрощення файлової структури можна вибрати опцію "Place solution and project in the same directory":

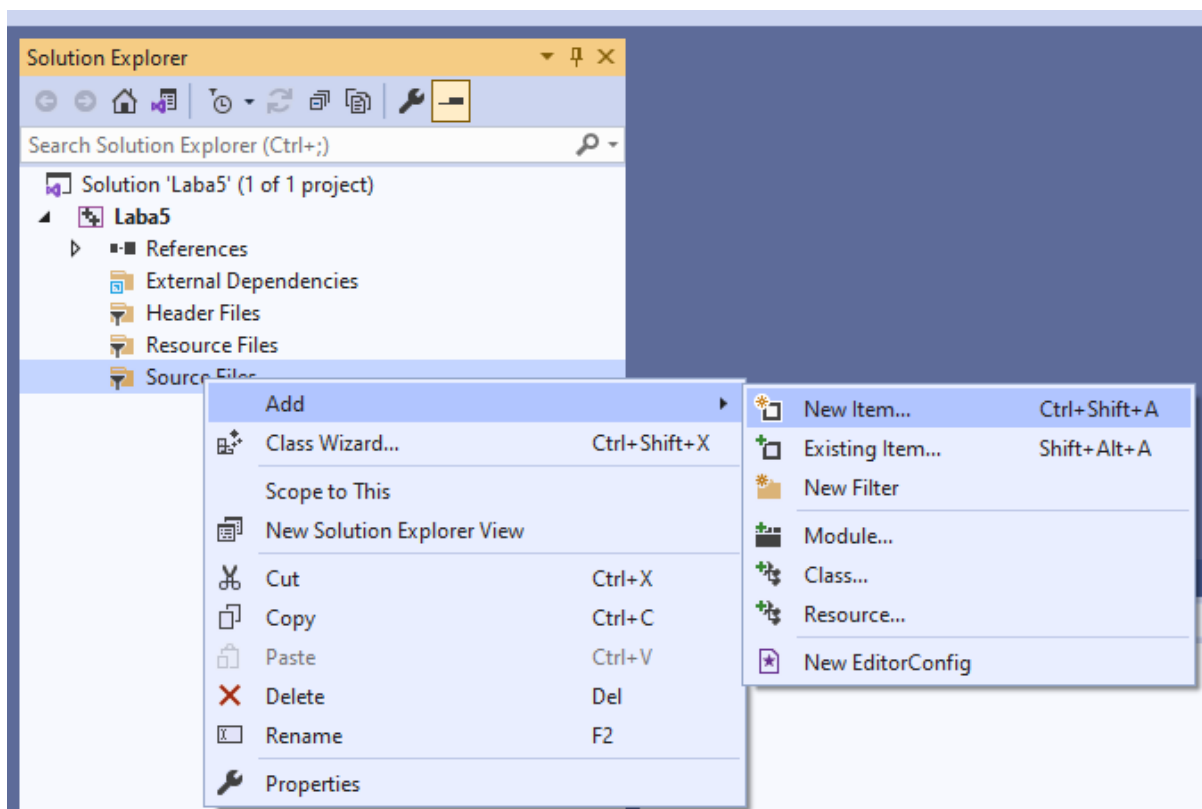


Після цього відкриється робоче вікно *Visual Studio*. В області ліворуч розташований Провідник рішень (*Solution Explorer*), який відображає поточне рішення, яке може містити один чи більше проектів. На рисунку нижче наведено приклад рішення «*Laba5*» (наведено зеленим). Це рішення містить один проект із ім'ям «*Laba5*» (наведено синім). В проект можуть входити різні типи файли (наведено жовтим) – файли з кодом програми (*Source Files*) із розширенням *\*.c*, файли із заголовками (*Header Files*) із розширенням *\*.h*, та інші. В прикладі на рисунку нижче проект ще не містить жодного файлу.



## Додавання файлів до проекту

Після створення проекту до нього можна додавати нові чи вже існуючі файли. Для цього потрібно обрати підкаталог проекту відповідного типу та, натиснувши правою кнопкою миші, визвати контекстне меню. В цьому контекстному меню потрібно обрати опції *Add: New Item...* для додавання нового файлу чи *Add: Existing Item...* для додавання існуючого файлу. На прикладі нижче показано додавання до проекту «Laba5» нового файлу, в якому буде написано код програми:



Якщо додавати вже існуючий файл, то у провіднику, який відкрився, потрібно просто вказати шлях до потрібного файлу.

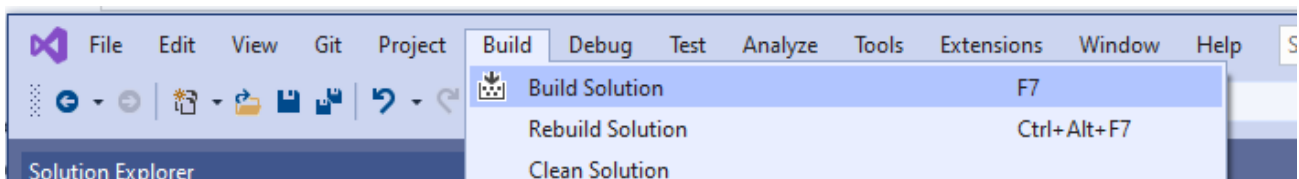
Якщо файл потрібно створити (опції *Add: New Item...* ), відкривається вікно *Add New Item*, в якому потрібно вказати тип файлу, його ім'я та місце розташування. За замовчуванням файл буде збережений в поточному каталозі проекту. На рисунку нижче показано додавання файлу, в якому буде написана програма мовою *C++* із ім'ям *Laba5\_code.c*. Цей файл буде частиною проекту *Laba5*.

Після натискання кнопки *Add*, в області ліворуч, над вікном *Output*, у текстовому редакторі відкривається створений файл.

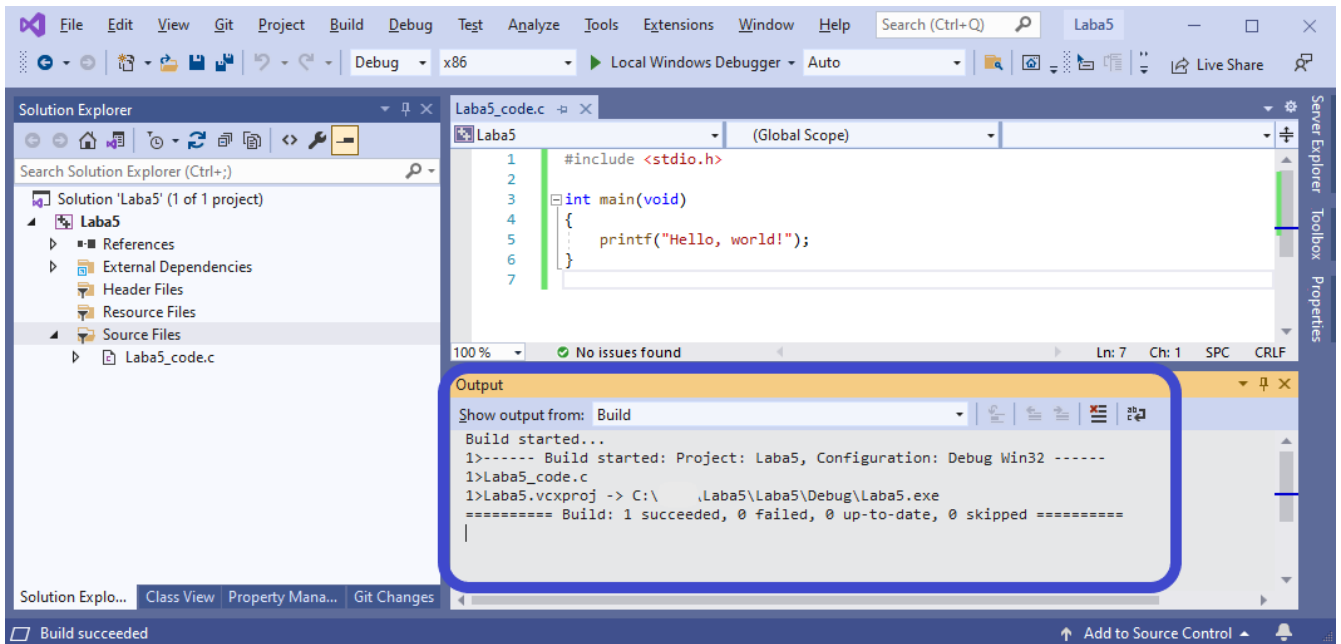
### Засоби побудови

У середовищі *Visual Studio* передбачений потужний набір засобів побудови та налагодження. За допомогою конфігурацій побудови можна вибирати компоненти для побудови, виключати компоненти, які не потрібно включати в побудову, а також визначати, як будуть побудовані вибрані проекти і для якої платформи.

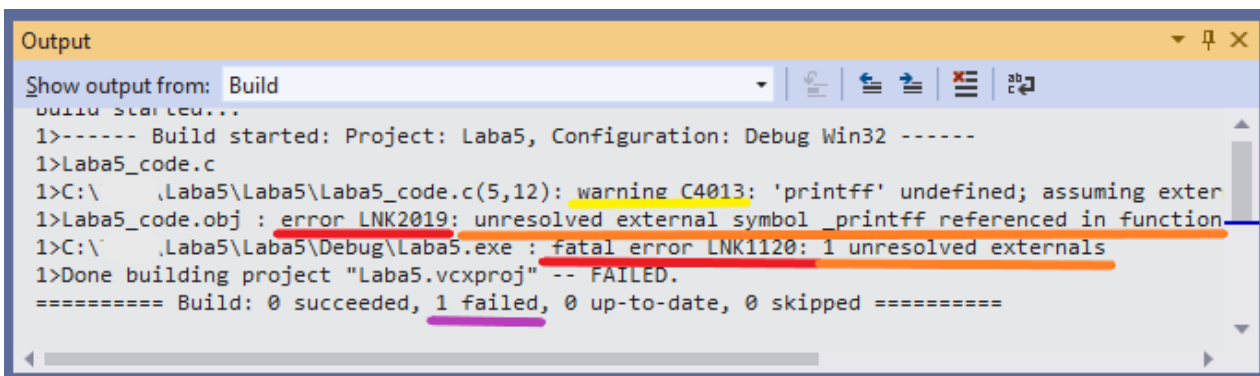
Для побудови (збірки) програми потрібно вибрати опції меню *Build:Build Solution*:



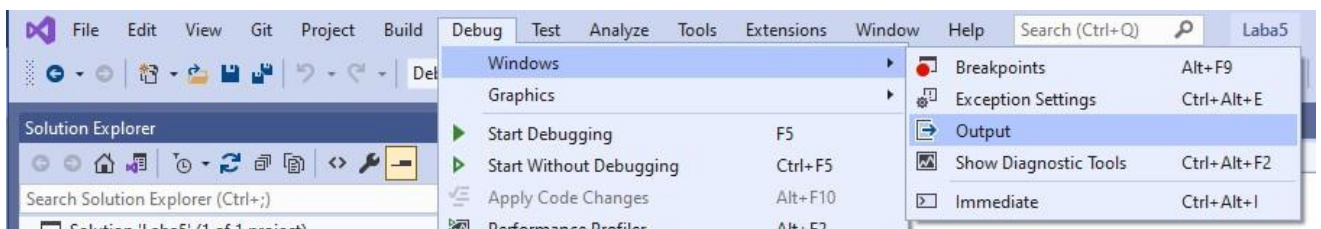
У вікні *Output* повинна відображатися інформація про хід збирання, а також про виявлені помилки компіляції. Приклад інформації про результати збирання проекту можна побачити на рисунку нижче (наведено синім):



Якщо при побудові файлу що виконується були виявлені помилки, то, зазвичай, файл що виконується побудовано не буде і про це буде повідомлено у вікні *Output* (наведено фіолетовим). Список помилок (наведено червоним) та попереджень (наведено жовтим) із детальним описом (наведено помаранчевим) також можна побачити в вікні *Output*:



Якщо вікно *Output* було випадкова закрито, то знову відкрити його можна обравши опції меню *Debug: Windows: Output*:

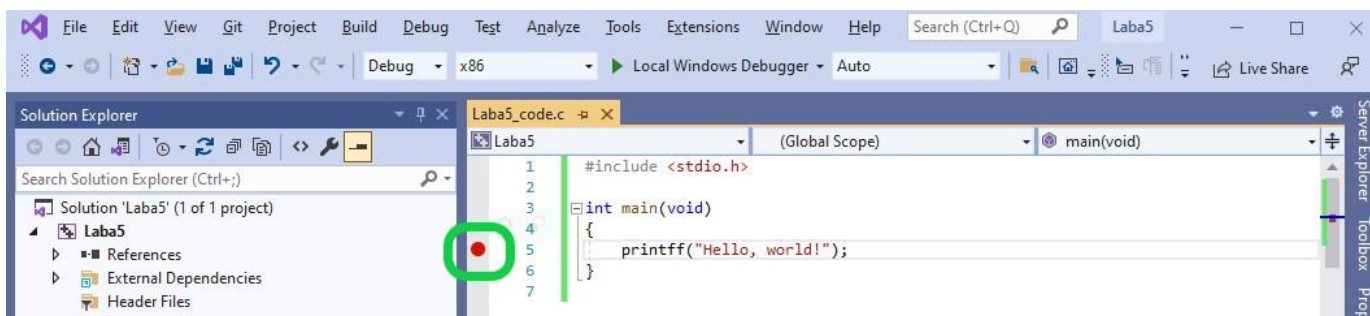


## Засоби налагодження

Якщо побудова виконуваного файлу пройшла успішно, тобто усі синтаксичні помилки було виправлено, то можна починати процес налагодження.

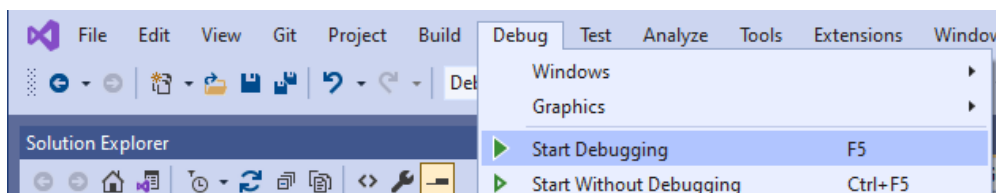
Процес налагодження, який виконується за допомогою відладчика, дозволяє виявити та усунути таких проблем, як логічні та семантичні помилки, які проявляються вже під час виконання. У режимі зупинки (*Break Point*) можна переглядати локальні змінні та інші дані, використовуючи такі засоби, як Вікна змінних (*Watch*) і Вікно пам'яті (*Memory*).

Для того, щоб зупинити виконання програми до певного рядка, в цьому рядку потрібно встановити *Break Point*. Його можна встановити клікнувши лівою кнопкою миші на сірій області, яка розташована ліворуч від тексту програми, навпроти обраного рядка (на рисунку нижче *Break Point* – це червоний кружечок):



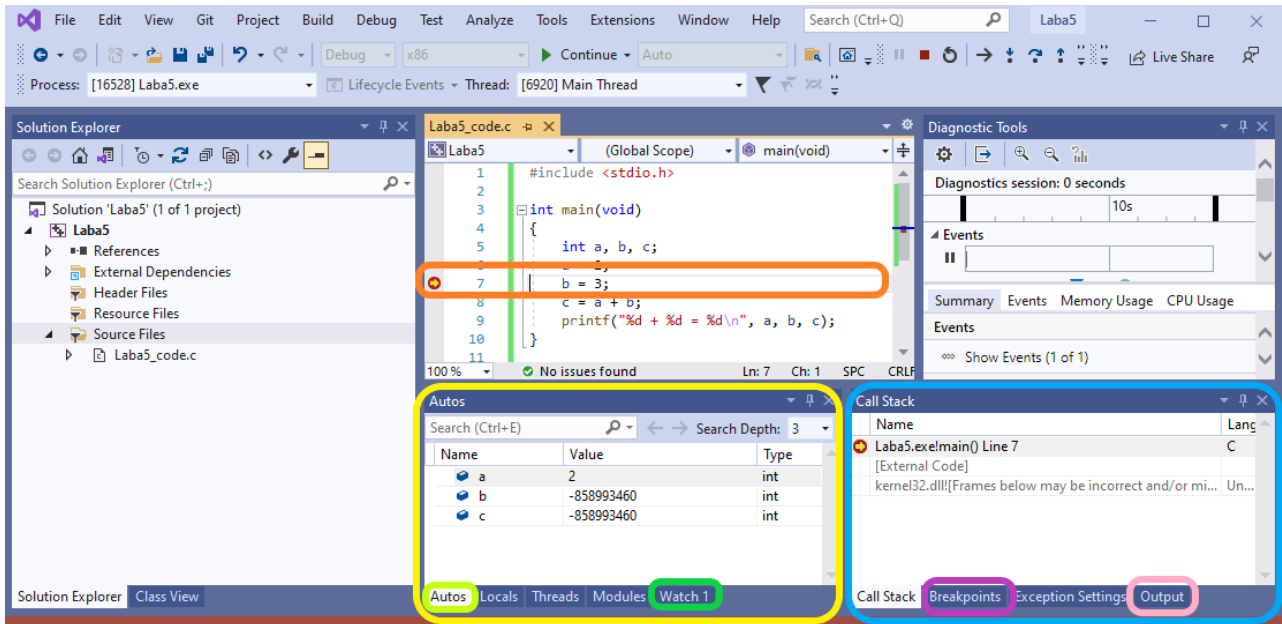
Для запуску програми в режимі налагодження потрібно або обрати опції меню

*Debug: Start Debugging*, або натиснувши гарячу клавішу **F5**:



Якщо заздалегідь були поставлені *Break Point*, то після запуску програми на виконання в режимі налагодження, програма буде виконана тільки до першого *Break Point*, після чого виконання програми буде призупинено. Про місце зупинки програми інформує жовта стрілка на сірому полі ліворуч від тексту

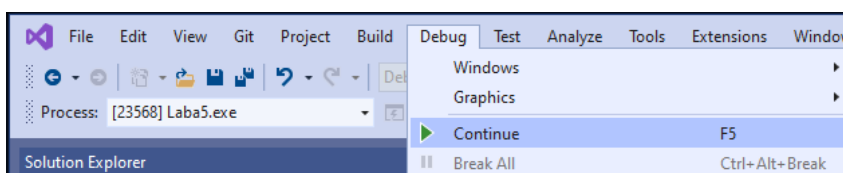
програми (на рисунку нижче наведено помаранчевим). Інформацію про значення, які зберігаються у змінних, можна отримати або за допомогою вікна *Autos* (на рисунку нижче наведено салатом), або вікна *Watch* (на рисунку нижче наведено зеленим):



До вікна *Autos* автоматично додаються змінні, видимі у межах поточного блоку, а до вікна *Watch* змінні потрібно додавати вручну. Для цього потрібно або безпосередньо набрати назву змінної в полі *Name* вікна *Watch* (там де написано підказка *Add item to watch*), або клікнути правою кнопкою миші і в контекстному меню обрати опцію *Add Watch*:

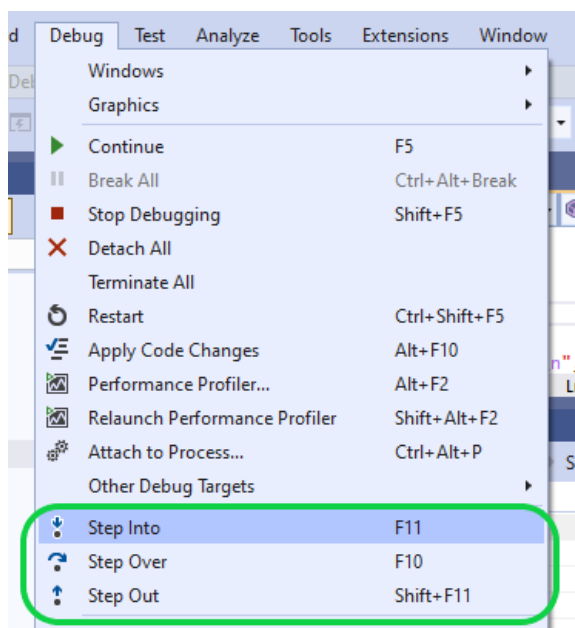


Продовжити виконання програми до наступної *Break Point* або до кінця програми можна або обравши опції меню *Debug: Continue*, або скориставшись гарячою клавішею *F5*:



Середовище *IDE Visual Studio* також пропонує покрокове виконання

програми, коли за один крок буде виконано інструкції певного рядка. Покрокове виконання інструкцій можна виконувати із заходом всередину функцій, що викликаються або беззаходу всередину функцій:



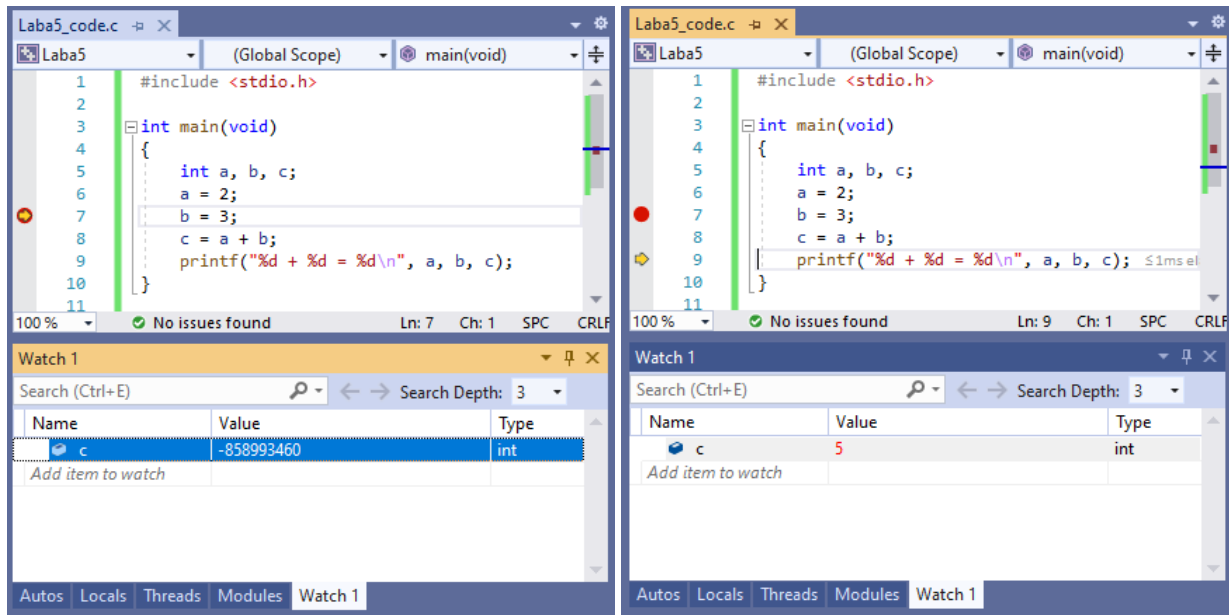
Якщо обрати опції меню *Debug:Step Over* або натиснути **F10**, то наступний рядок буде виконано без заходу всередину функції.

Якщо обрати опції меню *Debug:Step Into* або натисніть **F11**, то наступний рядок буде виконано із заходом всередину функцій яка написана в цьому рядку.

Якщо обрати опцію меню *Debug:Step Out*, то буде виконано вихід із поточної функції в функцію, що її викликала.

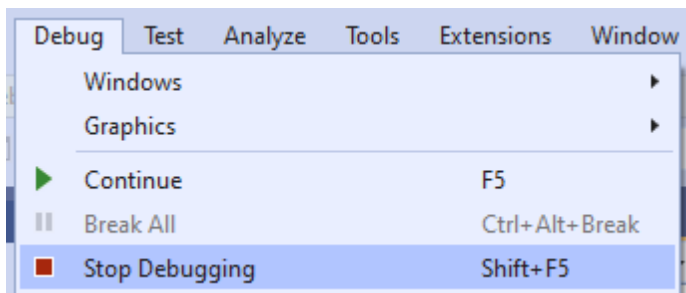
Значення змінних у вікні перегляду *Watch* оновлюються по мірі виконання програми.

На рисунку нижче наведено приклад вмісту змінної *c* до (рисунок нижче ліворуч) та після (рисунок нижче праворуч) виконання її модифікації (рядок 8):



## Дострокове завершення роботи програми в налагоджувальному режимі

Для дострокового завершення роботи в програмі, запущеної в налагоджувальному режимі в потрібно обрати опції меню *Debug:Stop Debugging* (абонатисніть **Shift + F5**):



## Деякі стандартні математичні функції

Стандартні математичні функції оголошені в заголовочному файлі *math.h*.

**abs** – абсолютне значення цілого числа -  $|x|$ : *int abs(int x)*;

**fabs** – абсолютне значення числа з плаваючою точкою -  $|x|$ : *double fabs(double x)*;

**sqrt** – обчислення квадратного кореня: *double sqrt(double x)*;

**pow** – піднесення до степеня: *double pow(double x, double y)*;

**cos** – косинус –  $\cos(x)$  (тут і далі  $x$  задається в радіанах): *double cos(double x)*;

**sin** – синус –  $\sin(x)$ : *double sin(double x)*;

**tan** – тангенс –  $\tan(x)$ : *double tan(double x)*;

**acos** – арккосинус –  $\arccos(x)$ : *double acos(double x)*;

**asin** – арксинус –  $\arcsin(x)$ : *double asin(double x)*;

**atan** – арктангенс –  $\arctg(x)$ : *double atan(double x)*;

**atan2** – арктангенс –  $\arctg(y/x)$ : *double atan2(double y, double x)*;

**exp** – експонента: *double exp(double x)*;

**log** – натуральний логарифм –  $\ln(x)$ : *double log(double x)*;  
**log10** – десятковий логарифм –  $\log_{10}(x)$ : *double log10(double x)*;

Для роботи з даними типу *float* більшість перерахованих функцій має еквіваленти, такі як *logf*, *powf* і т.д. (див. довідку).

### *Деякі математичні константи*

Стандартні математичні функції знаходяться в заголовному файлі *math.h*. Для використання констант **перед** підключенням файлу заголовків *math.h* слід додати наступний рядок

```
#define _USE_MATH_DEFINES
```

Перелік констант, які визначені в файлі *math.h*, наведено в таблиці нижче:

| Символ     | Вираз          | Значення                |
|------------|----------------|-------------------------|
| M_E        | e (експонента) | 2.71828182845904523536  |
| M_LOG2E    | $\log_2(e)$    | 1.44269504088896340736  |
| M_LOG10E   | $\log_{10}(e)$ | 0.434294481903251827651 |
| M_LN2      | $\ln(2)$       | 0.693147180559945309417 |
| M_LN10     | $\ln(10)$      | 2.30258509299404568402  |
| M_PI       | Pi             | 3.14159265358979323846  |
| M_PI_2     | $\pi/2$        | 1.57079632679489661923  |
| M_PI_4     | $\pi/4$        | 0.785398163397448309616 |
| M_1_PI     | $1/\pi$        | 0.318309886183790671538 |
| M_2_PI     | $2/\pi$        | 0.636619772367581343076 |
| M_2_SQRTPI | $2/\sqrt{\pi}$ | 1.12837916709551257390  |
| M_SQRT2    | $\sqrt{2}$     | 1.41421356237309504880  |
| M_SQRT1_2  | $1/\sqrt{2}$   | 0.707106781186547524401 |

### **Робоче завдання**

1. Скласти алгоритм розв'язання задачі згідно варіанту. Намалювати блок-

схему алгоритму.

2. Створити новий порожній проект.

3. Написати програму, що реалізує виконання складеного алгоритму. Вхідні дані(задані символами у кожному варіанті) задати константами, а результат обчислень вивести на екран (використовувати функцію *printf()*).

4. Скомпілювати проект.

5. Провести покрокове налагодження:

a. Встановити *Break Point* на певному рядочку всередині програми.

b. Додати до віконця *Watch* частину змінних, які використовуються в програмі.

c. Запустити на виконання програму у режимі налагодження.

d. Подивитися значення змінних за допомогою віконця *Watch*.

e. Виконати декілька рядків коду без заходу всередину функції.

f. Виконати рядок коду, який містить математичну функцію, із заходом в середину функції.

g. Повернутись із математичної функції в функцію *main()* та переконатися за допомогою віконця *Watch* що деякі змінні змінили своє значення.

h. Виконати програми до кінця та продемонструвати результат обчислення задачі згідно варіанту.

| Варіант | Завдання  |
|---------|---|
| 1.      | Обчислити довжину кола, площу кола і об'єм кулі одного і того ж заданого радіусу $r$ .  |
| 2.      | Обчислити периметр прямокутного трикутника за довжинами двох його катетів $a$ та $b$ .  |
| 3.      | За координатами трьох вершин $((x_1, y_1), (x_2, y_2)$ та $(x_3, y_3))$ трикутника знайти його площу.   |
| 4.      | Обчислити площу кільця, заданого внутрішнім радіусом $r$ та зовнішнім радіусом $R$ ( $R > r$ ).   |
| 5.      | Обчислити площу прямокутного трикутника за довжинами його катета $a$ і гіпотенузи $c$ .   |
| 6.      | Змішали $v_1$ літрів води з температурою $t_1$ градусів Цельсія з $v_2$ літрами води з температурою $t_2$ градусів Цельсія. Обчислити температуру суміші що утворилася. |

|     |  |
|-----|--|
| 7.  | У кубічний акваріум зі стороною $a$ метрів, наповнений до країв, опустили кулю діаметром $b$ метрів. Обчислити, який відсоток від початкового об'єму води виллється. |
| 8.  | Обчислити периметр рівнобедреної трапеції з основами $a$ і $b$ і висотою $h$ .   |
| 9.  | Обчислити площу рівностороннього трикутника зі стороною $a$ .  |
| 10. | У кубі зі стороною $a$ висвердлили наскрізний циліндричний отвір діаметром $b$ . На скільки відсотків зменшився об'єм куба?  |
| 11. | Обчислити периметр прямокутного трикутника за довжинами катетів $a$ та $b$ .   |
| 12. | Обчислити площу рівнобедреної трапеції з основами $a$ та $b$ і висотою $h$ .   |
| 13. | Обчислити площу кола, описаного навколо рівностороннього трикутника зі стороною $a$ .  |
| 14. | За координатами трьох вершин $((x_1, y_1), (x_2, y_2)$ та $(x_3, y_3))$ трикутника знайти його периметр.   |
| 15. | Обчислити довжину кола, вписаного в рівносторонній трикутник зі стороною $a$ .   |
| 16. | Є рівносторонній трикутник зі стороною $a$ . Знайти, яке співвідношення між площею описаного навколо нього і вписаного в нього кіл.                                  |
| 17. | Обчисліть периметр прямокутного трикутника за довжинами його катета $a$ і гіпотенузи $c$ .   |
| 18. | Є рівносторонній трикутник зі стороною $a$ . Знайти співвідношення довжин кіл, описаного та вписаного в нього.   |
| 19. | Обчислити довжину кола, описаного навколо рівностороннього трикутника зі стороною $a$ .  |
| 20. | Обчислити об'єм кулі та площу її поверхні за заданим радіусом $r$ .  |
| 21. | Обчислити площу кола, вписаного в рівносторонній трикутник зі стороною $a$ .   |

### Контрольні питання

1. Які основні компоненти зазвичай включає в себе сучасне середовище розробки?
2. Які засоби відлагодження в середовищі розробки *Visual Studio* вам відомі?
3. Які додаткові можливості дає запуск програми в режимі налагодження?
4. Які стандартні математичні функції мови C++ вам відомі?

5. Як встановити та зняти Break Point у Visual Studio?
6. Які відмінності між командами Step Into (F11), Step Over (F10) та Step Out при налагодженні?
7. Як у вікні Watch можна додати змінну для відстеження?
8. Які математичні константи доступні у бібліотеці <math.h>? Наведіть приклади.
9. Яка різниця між функціями sqrt() і pow()?