

Лекція 1.
Вступ до програмування.
Типи даних та операції мови Сі

План

- 1) Організаційні моменти**
- 2) Вступ до програмування**
- 3) Рішення та проекти у Visual Studio**
- 4) Найпростіша програма на мові Сі**
- 5) Типи даних**
- 6) Введення та виведення даних**
- 7) Арифметичні та математичні операції**

1. Вступ до програмування

Комп'ютерна програма — набір інструкцій у вигляді слів, цифр, кодів, схем, символів чи у будь-якому іншому вигляді, виражених у формі, придатній для зчитування комп'ютером, які приводять його у дію для досягнення певної мети або результату.

З чого складається комп'ютерна програма?

Програма складається з коду – послідовності команд, написаних на мові програмування.

Кожна мова має свою синтаксис і семантику, тобто набір правил, за якими записуються команди та їхнє значення.

Створення програми для обчислення арифметичного виразу $y=a+b$

Delphi

```
program Suma;
var a, b, y: integer;
begin writeln('Введіть значення a: ');
      readln(a);
      writeln('Введіть значення b: ');
      readln(b);
      y := a + b;
      writeln('Сума чисел a та b дорівнює: ', y);
end.
```

Python

```
a = float(input("Введіть перше число: "))
b = float(input("Введіть друге число: "))
y = a + b print("Сума:", y)
```

C

```
#include <stdio.h>
int main() {
    int a, b, y;
    printf("Введіть значення для a: "); scanf("%d", &a);
    printf("Введіть значення для b: "); scanf("%d", &b);
    y = a + b;
    printf("Сума a і b дорівнює: %d\n", y);
    return 0;
}
```

C++

```
#include <iostream>
using namespace std;
int main() {
    double a, b, y;
    cout << "Введіть перше число: "; cin >> a;
    cout << "Введіть друге число: "; cin >> b;
    y = a + b;
    cout << "Сума: " << y << endl;
    return 0;
}
```

Парадигма програмування - це сукупність принципів, методів і понять, що визначають спосіб конструювання програм.

Структурне програмування націлено на скорочення часу розробки та спрощення підтримки програм за рахунок використання блокових операторів і підпрограм.

Парадигма програмування - це сукупність принципів, методів і понять, що визначають спосіб конструювання програм.

Структурне програмування націлено на скорочення часу розробки та спрощення підтримки програм за рахунок використання блокових операторів і підпрограм.

Основні принципи структурного програмування:

Модульність. Розбиття програми на окремі, логічно пов'язані блоки (модулі, функції, процедури), що спрощує розуміння коду та дозволяє повторно використовувати окремі частини.

Ієрархічна структура. Організація програми у вигляді ієрархії, де кожен модуль може викликати інші модулі нижчого рівня.

Обмежений набір конструкцій. Використання лише трьох основних конструкцій: послідовність; розгалуження; цикл.

Відмова від оператора GOTO!!!!



Чому саме C:

- Мова спочатку створювався для системного програмування, тому не дивно, що її активно застосовують при створенні операційних систем і різного прикладного програмного забезпечення;
- Зростання кількості пристроїв, що підтримують Інтернет речей, а ці пристрої зав'язані на управлінні машинним кодом і пам'яттю;
- Низькорівневе програмування різних пристроїв, наприклад, Arduino;
- Системний адміністратор, який знає Cі, глибше розуміє структуру ОС, може справлятися з несподіваними проблемами експлуатації і пропонувати більш оптимізовані рішення.

Cі - типізована мова програмування загального призначення, розроблена в 1969-1973 роках співробітником Bell Labs Деннісом Рітчі.

Серед переваг мови Cі потрібно відзначити основні:

- компактність та універсальність коду;
- швидкість виконання програм;
- гнучкість мови;
- висока структурованість.
- проста мовна база;
- доступ до пам'яті через використання покажчиків.

 **Visual Studio 2022** | 

The most comprehensive IDE for .NET and C++ developers on Windows for building web, cloud, desktop, mobile apps, services and games.

Preview
Get early access to latest features not yet in the main release
[Learn more →](#)

Community
Powerful IDE, free for students, open-source contributors, and individuals
[Free download](#)

Professional
Professional IDE best suited to small teams
[Free trial](#)

Enterprise
Scalable, end-to-end solution for teams of any size
[Free trial](#)

[Release notes →](#) [Compare Editions →](#) [How to install offline →](#) [License Terms →](#)

<https://visualstudio.microsoft.com/downloads/>

2. Рішення та проекти у Visual Studio

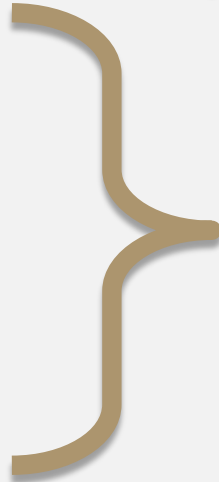
- **Проект (Project)** – програма;
- **Рішення (Solution)** – група проектів (набір програм)

Лабораторна робота №1

- Завдання №1
- Завдання №2
- Завдання №3
- Завдання №4

P_Lab1

- Task1
- Task2
- Task3
- Task4



Рішення





*Проекти
(4 штуки)*

Для того, щоб створити рішення та проект потрібно:





Visual Studio 2022

Open recent

Search recent (Alt+S) 🔍

- Today
 -  Project1.sln 31.08.2023 11:28
C:\Users\Gala\source\repos\Project1
- Older
 -  Practik_1.sln 10.02.2023 14:53
D:\СИ(Visual Studio 2017)\source\repos\Practik_1
 -  Practik.sln 25.01.2023 19:31
D:\СИ(Visual Studio 2017)\source\repos\Practik
 -  LekOP22.sln 28.12.2022 8:20
D:\СИ(Visual Studio 2017)\source\repos\LekOP22

Get started

-  **Clone a repository**
Get code from an online repository like GitHub or Azure DevOps
-  **Open a project or solution**
Open a local Visual Studio project or .sln file
-  **Open a local folder**
Navigate and edit code within any folder
-  **Create a new project**
Choose a project template with code scaffolding to get started

Create a new project

Recent project templates

Empty Project

C++

Search for templates (Alt+S)



All languages

All platforms

All project types



Blank Solution

Create an empty solution containing no projects

New

Other



Empty Project

Start from scratch with C++ for Windows. Provides no starting files.

C++

Windows

Console



Console App

Run code in a Windows terminal. Prints "Hello World" by default.

C++

Windows

Console



CMake Project

Build modern, cross-platform C++ apps that don't depend on .sln or .vcxproj files.

C++

Windows

Linux

Console



Windows Desktop Wizard

Create your own Windows app using a wizard.

C++

Windows

Desktop

Console

Library



Windows Desktop Application

A project for an application with a graphical user interface that runs on Windows.

C++

Windows

Desktop

Back

Next

Configure your new project

Empty Project C++ Windows Console

Project name
Project2

Location
C:\Users\Gala\source\repos

Solution name ⓘ
Project2

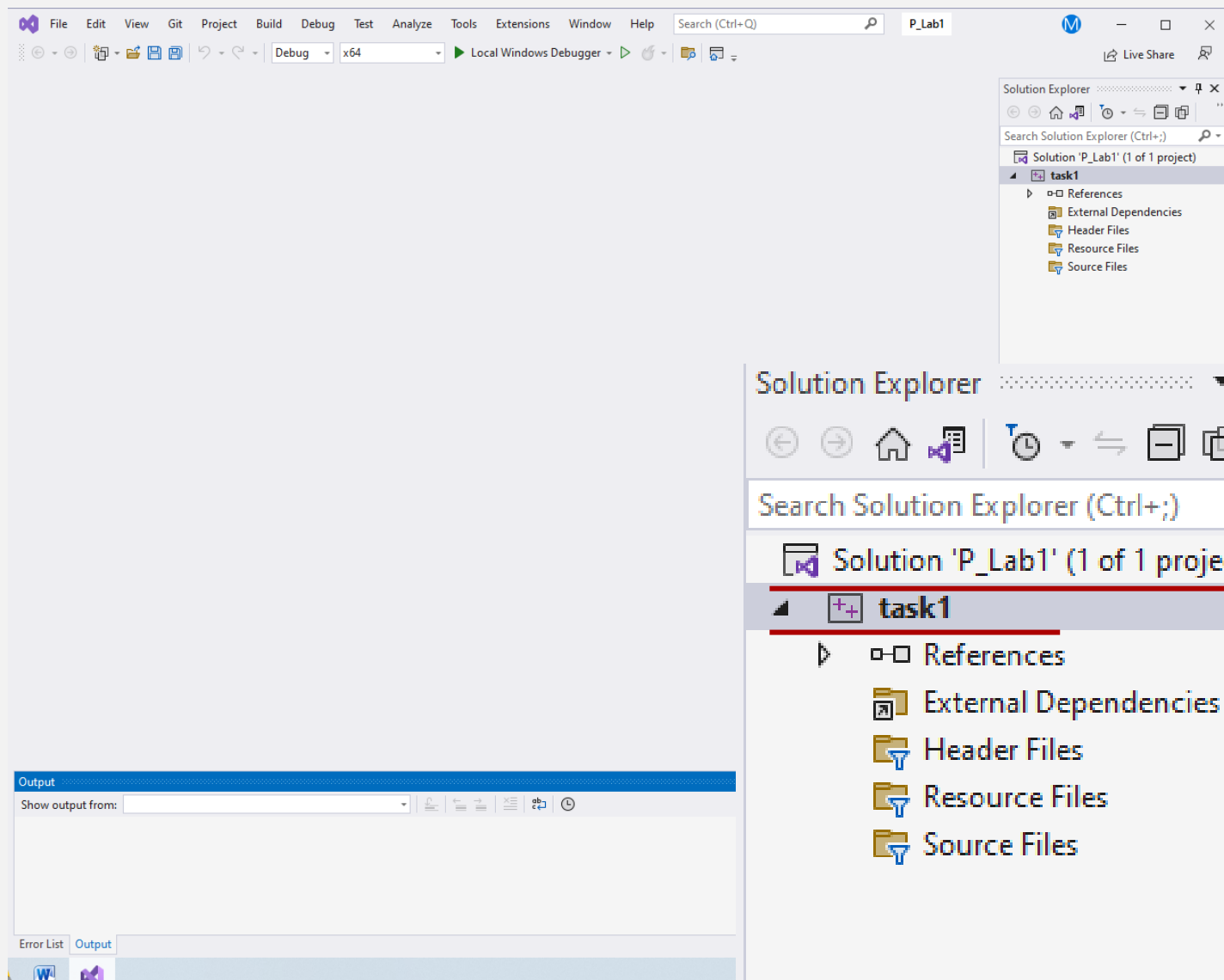
Place solution and project in the same directory

Back Create

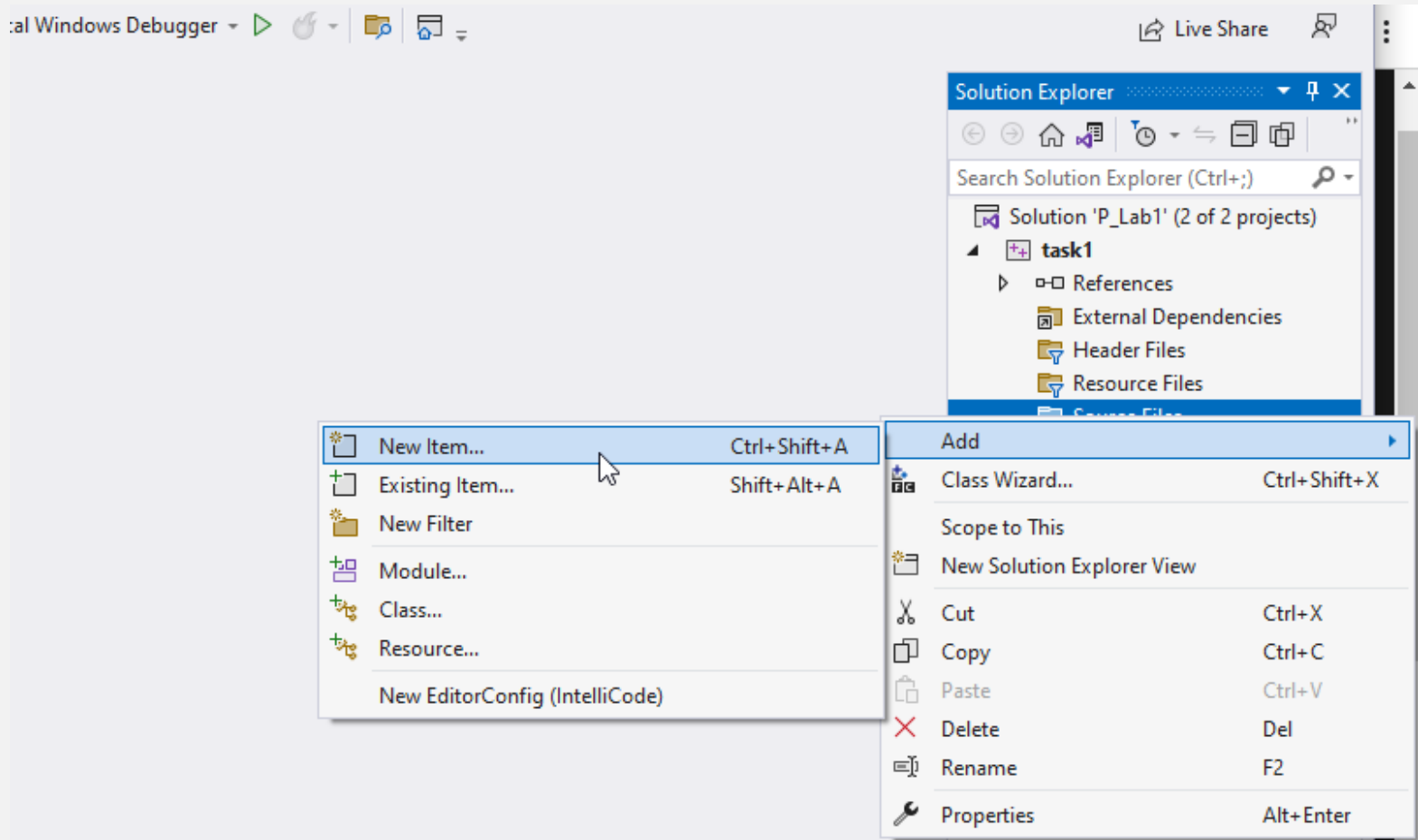
Назва проекту

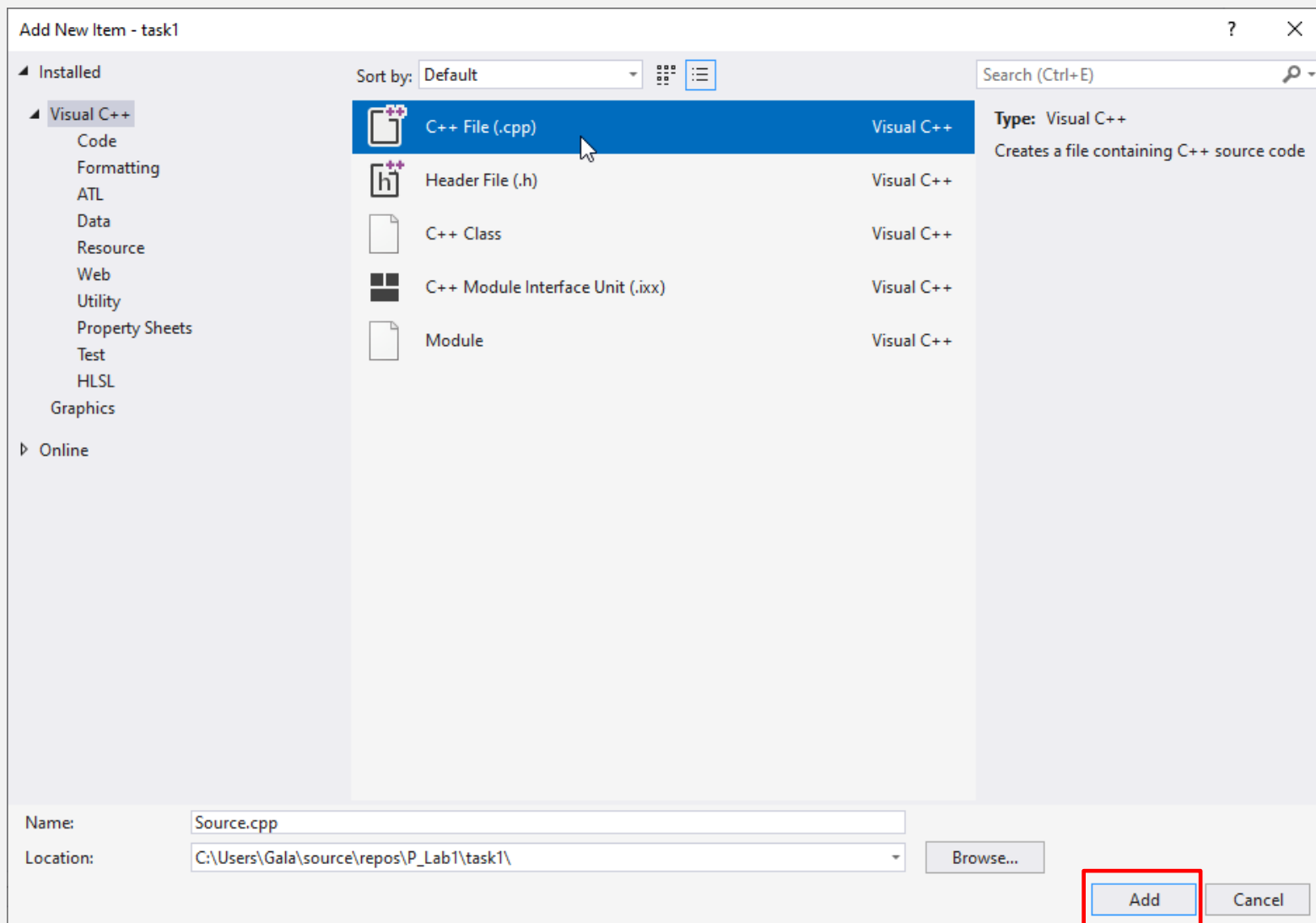
Папка, де будуть зберігатися файли

Назва рішення

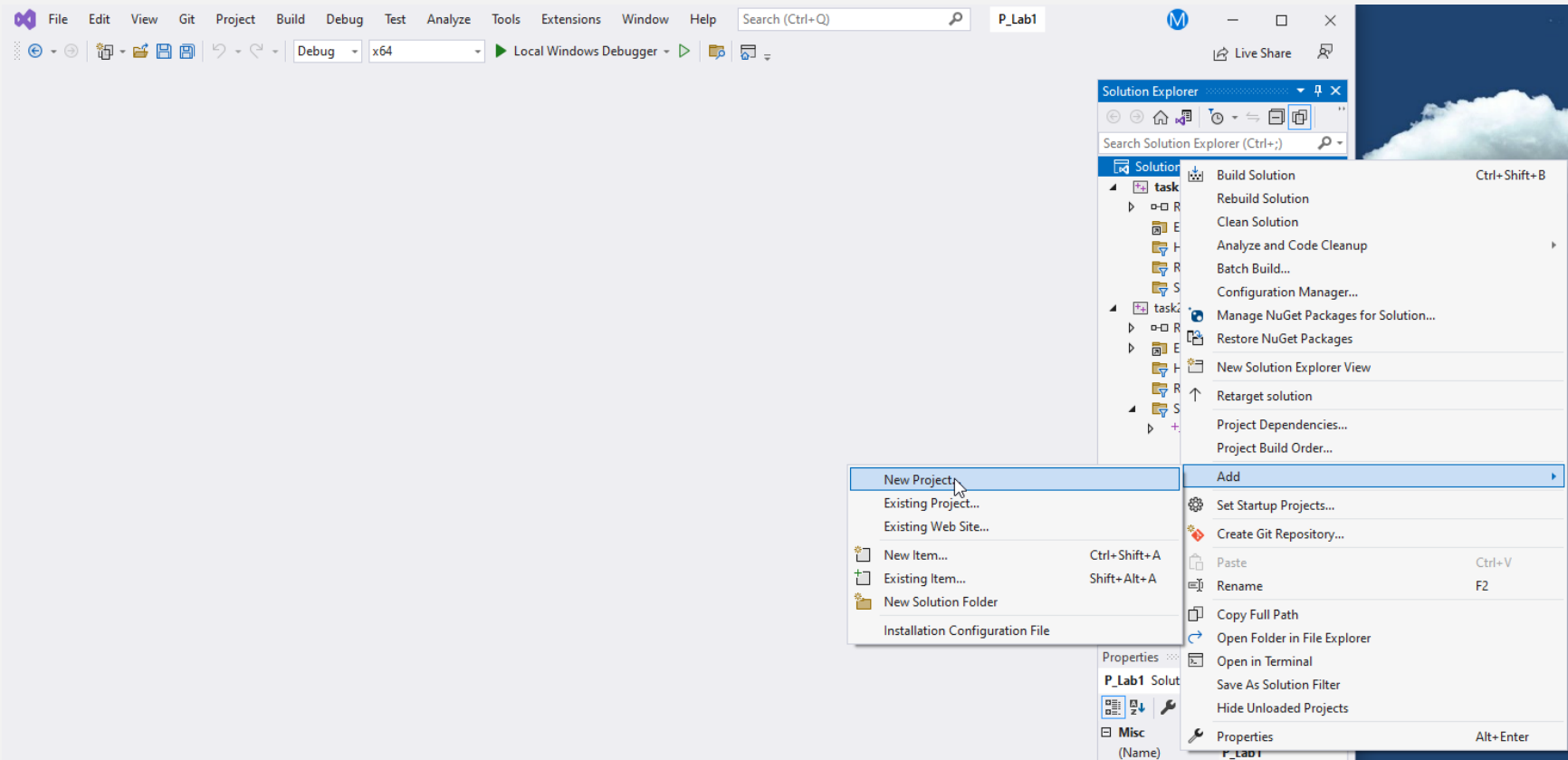


Створення робочого файлу : ПКМ → Source Files → Add → New Item





Створення нового проекту у вже існуючому рішенні: ПКМ -> ADD -> New Project



Add a new project

Recent project templates

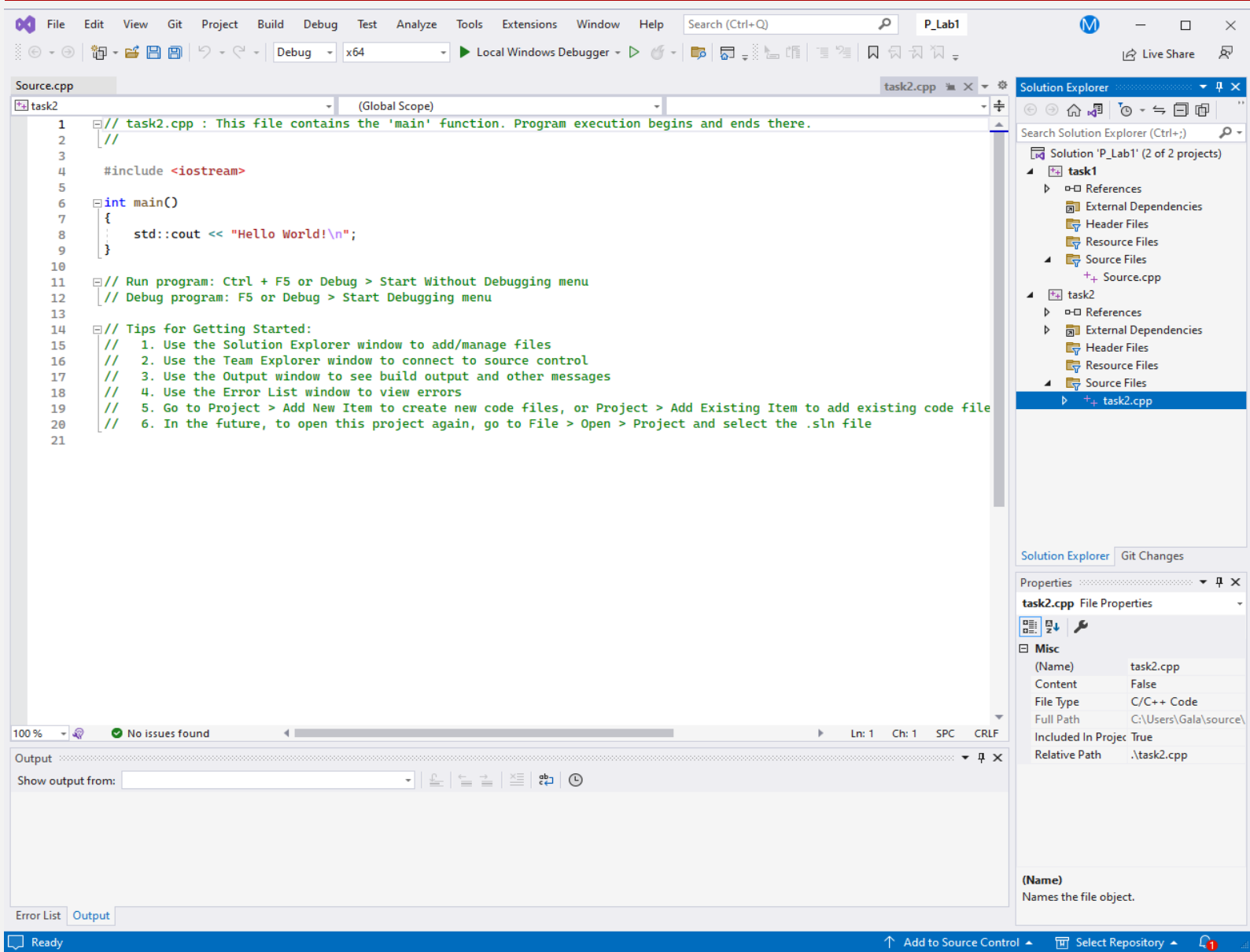
- Empty Project C++
- Console App C++

Search for templates (Alt+S)

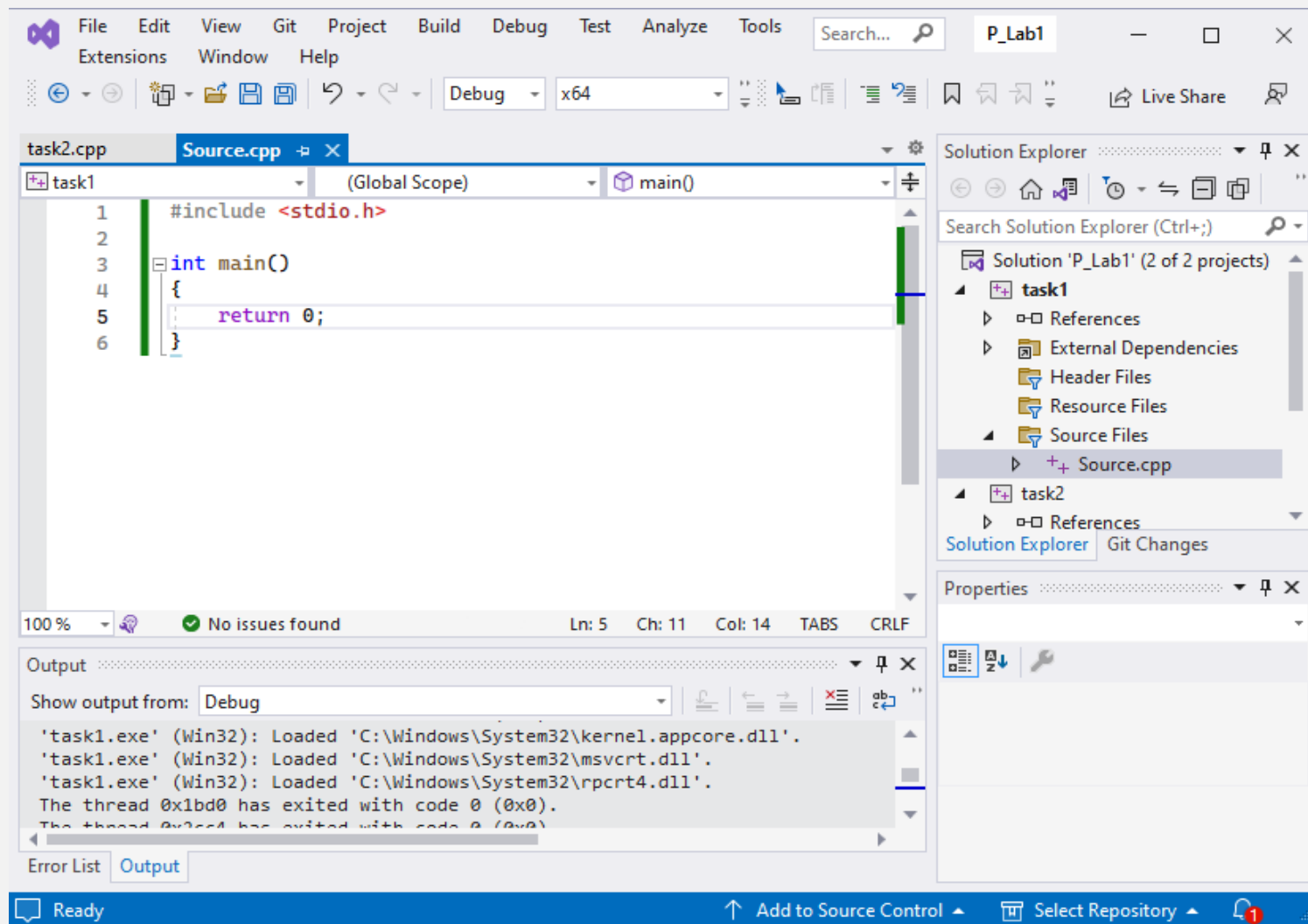
All languages All platforms All project types

- Empty Project**
Start from scratch with C++ for Windows. Provides no starting files.
C++ Windows Console
- Console App**
Run code in a Windows terminal. Prints "Hello World" by default.
C++ Windows Console
- CMake Project**
Build modern, cross-platform C++ apps that don't depend on .sln or .vcxproj files.
C++ Windows Linux Console
- Windows Desktop Wizard**
Create your own Windows app using a wizard.
C++ Windows Desktop Console Library
- Windows Desktop Application**
A project for an application with a graphical user interface that runs on Windows.
C++ Windows Desktop
- TypeScript Console Application**
A basic TypeScript Console application template which can be run with your local node installation.

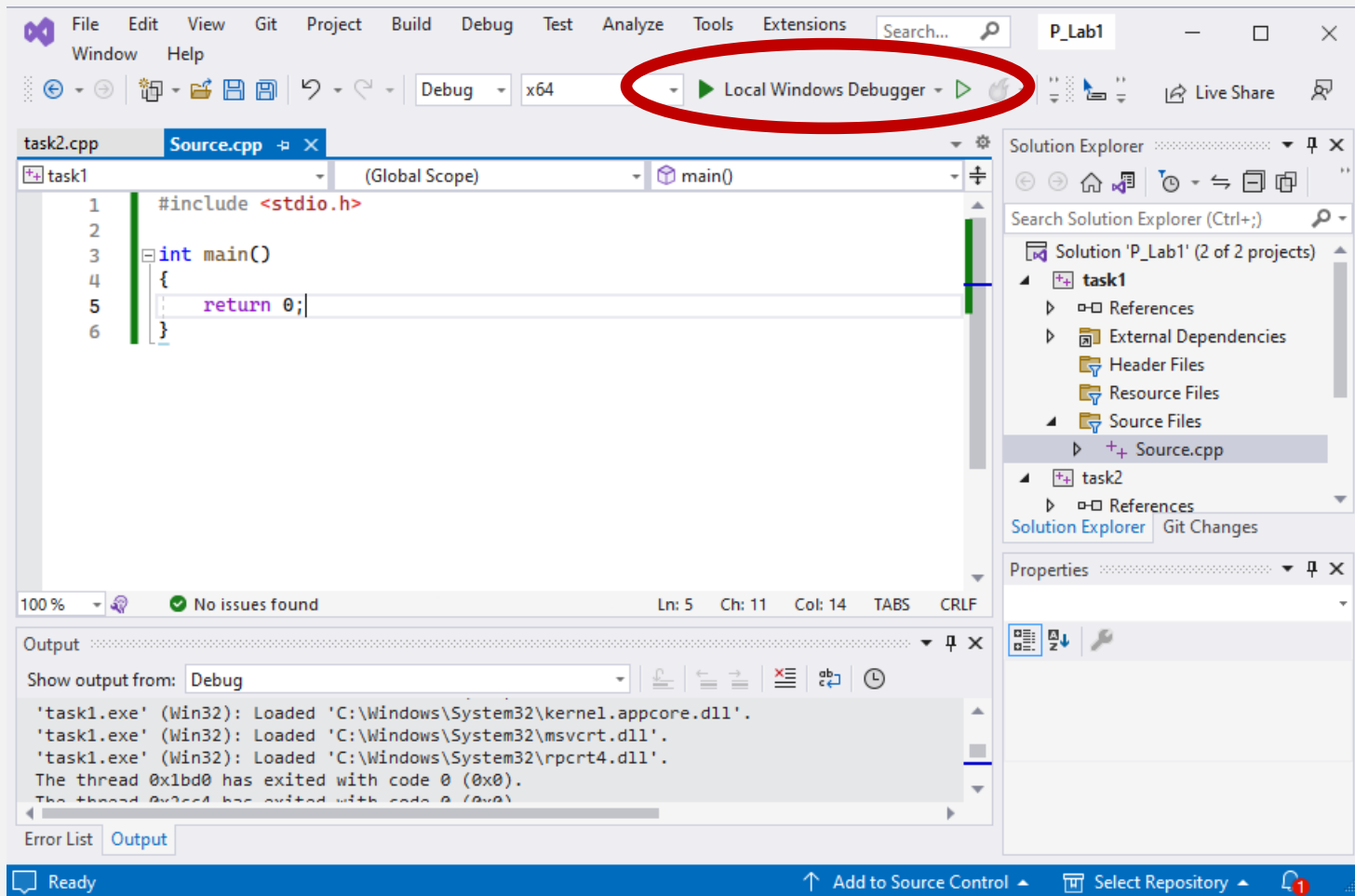
Next



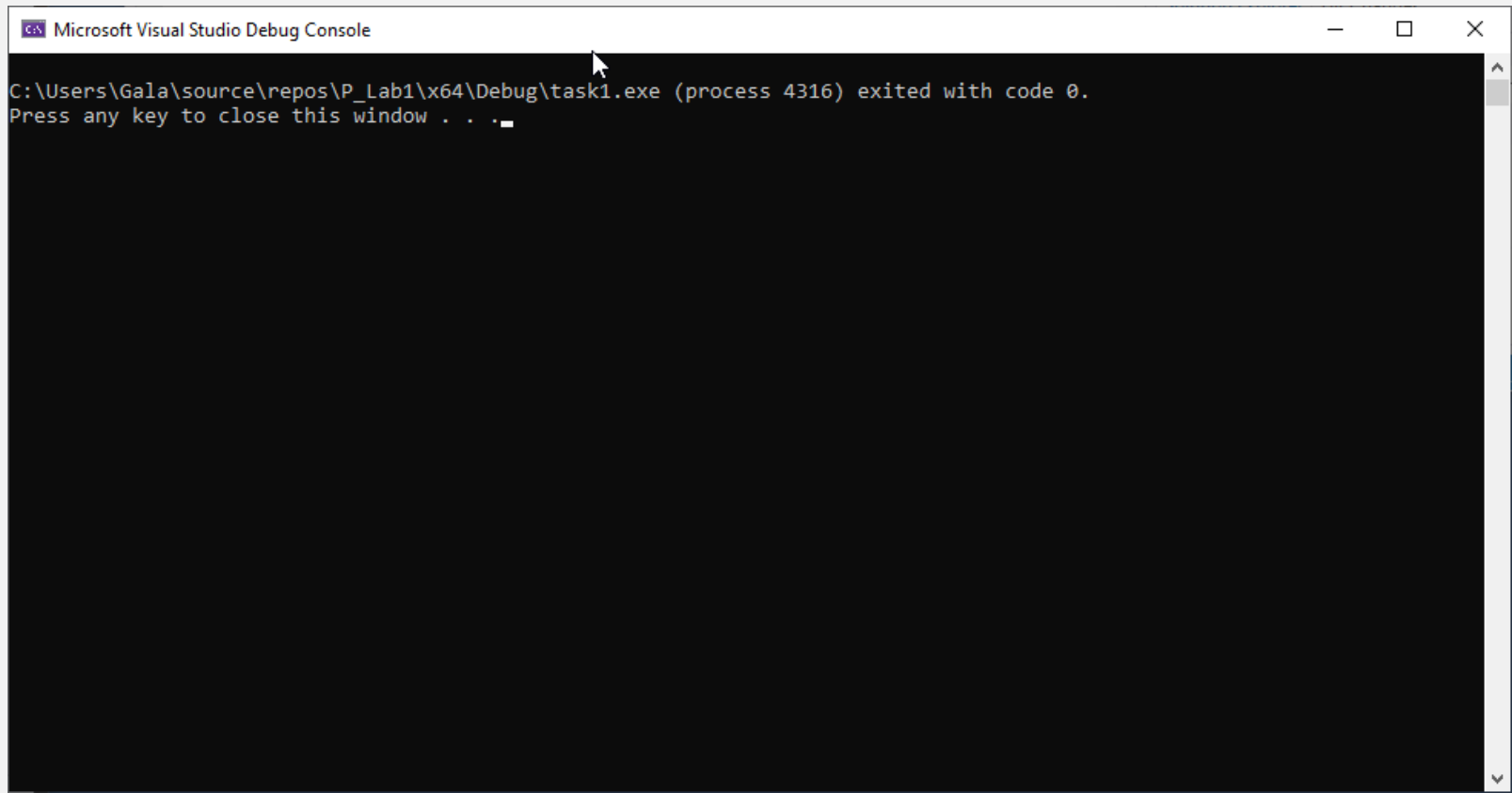
3. Найпростіша програма на мові Сі



Запуск програми виконується натисканням
клавiшi **F5** або



Щоб програма одразу не закривалася для запуску потрібно натискати комбінацію клавіш **CTRL-F5**



```
Microsoft Visual Studio Debug Console
C:\Users\Gala\source\repos\P_Lab1\x64\Debug\task1.exe (process 4316) exited with code 0.
Press any key to close this window . . .
```

4. Типи даних

Цілі типи даних

Тип	Діапазон значень	Розмір (байт)
char	-128 ... 127	1
short	-32768 ... 32767	2
int		2 або 4
long	-2 147 483 648 ... 2 147 483 647	4
long long	-9 223 372 036 854 775 807 ... 9 223 372 036 854 775 807	8
unsigned char	0 ... 255	1
unsigned short	0 ... 65535	2
unsigned int		2 або 4
unsigned long	0 ... 4 294 967 295	4
unsigned long long	0 ... 18 446 744 073 709 551 615	8



Дробові типи даних

Тип	Кількість знаків після коми	Розмір (байт)
float	6-7	4
double	15-16	8
long double	19-20	10

Тип даних – множина можливих значень та набір операцій над цими значеннями.

Важливо!!!

Точний розмір у байтах для кожного типу може змінюватись в залежності від архітектури та компілятора.

Діапазон значень для кожного типу також залежить від архітектури та компілятора.

При виборі *модифікатора* необхідно враховувати діапазон значень, так і вимоги до ефективності.

Змінна – це область пам'яті, яка має ім'я і в якій зберігається значення певного типу даних

Основні характеристики змінних:

- **Ім'я змінної (Унікальне слово або набір символів, за яким ми звертаємось до змінної)**
- **Значення змінної (Визначає, які дані можна зберігати у змінній)**
- **Тип змінної (Конкретне дане, яке зберігається у змінній на даний момент)**

Правила написання імен змінних

- Перший символ повинен бути літерою (великою або маленькою) або символом підкреслення (`_`) і не може бути цифрою.
- Наступні символи не можуть бути пробілами або спеціальними символами (наприклад, `@`, `#`, `$`, `%`, `&`).
- Більшість мов програмування розрізняють великі та маленькі літери. Тобто змінні `myVariable` та `MyVariable` – це різні змінні.
- Не можна використовувати як імена змінних ключові слова мови програмування.
- Ім'я змінної має бути максимально зрозумілим та відображати призначення змінної. Це полегшує розуміння коду.

Стили написання імен змінних

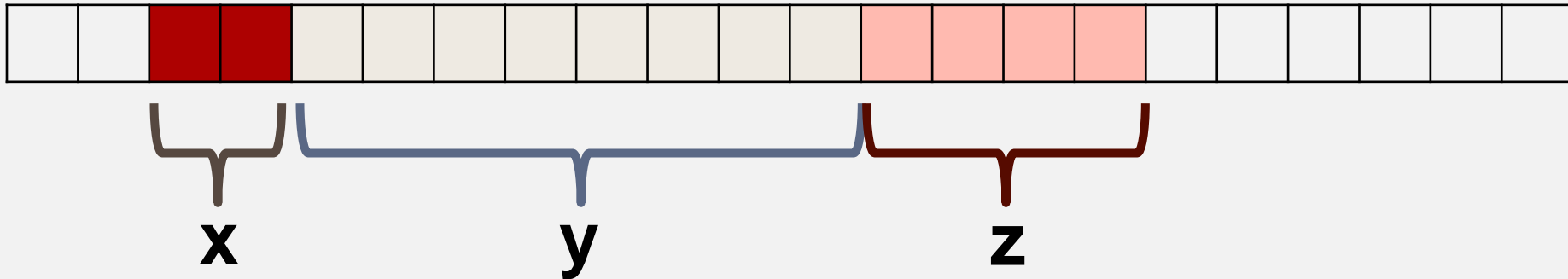
- **camelCase.** Перше слово пишеться з маленької літери, а кожне наступне – з великої (наприклад, ,). `firstName lastName`
- **snake_case.** Всі слова пишуться маленькими літерами і розділяються символом підкреслення (наприклад, ,). `first_name last_name`
- **PascalCase.** кожне слово пишеться з великої літери (наприклад, ,). `FirstName LastName`

Для того, щоб створити змінну, її потрібно **оголосити**, вказавши тип даних:

тип даних Ім'я змінної ;

Приклад:

short x;
double y;
int z;

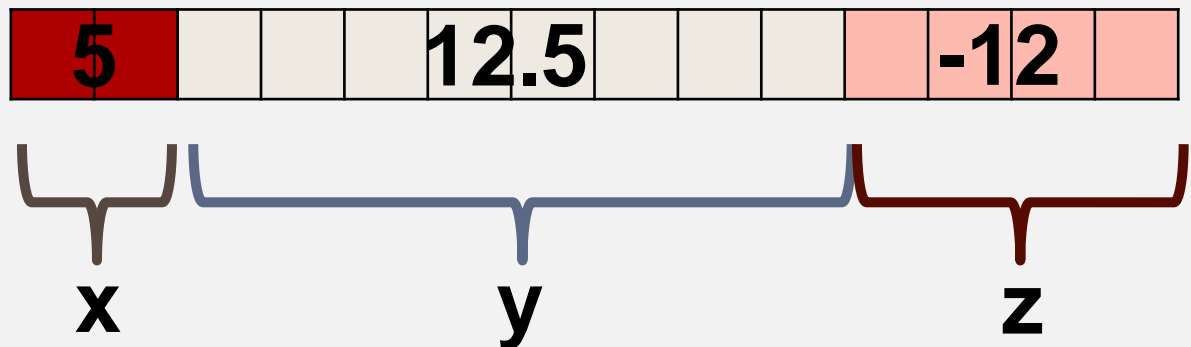


Для запису конкретного значення у змінну потрібно використовувати операцію **присвоювання значення змінній**:

ім'я змінної = **значення** ;

Приклад:

```
short x;  
double y;  
int z;  
x = 5;  
y = 12.5;  
z = -12;
```



В одному рядку можна оголошувати кілька змінних та одразу присвоювати їм початкові значення.

Приклад 1:

```
short x;  
double y;  
int z;  
short a;  
short b;  
x = 5;  
y = 12.5;  
z = -12;
```

Приклад 2:

```
short x = 5, a, b;  
double y = 12.5;  
int z = -12;
```

5. Введення та виведення даних

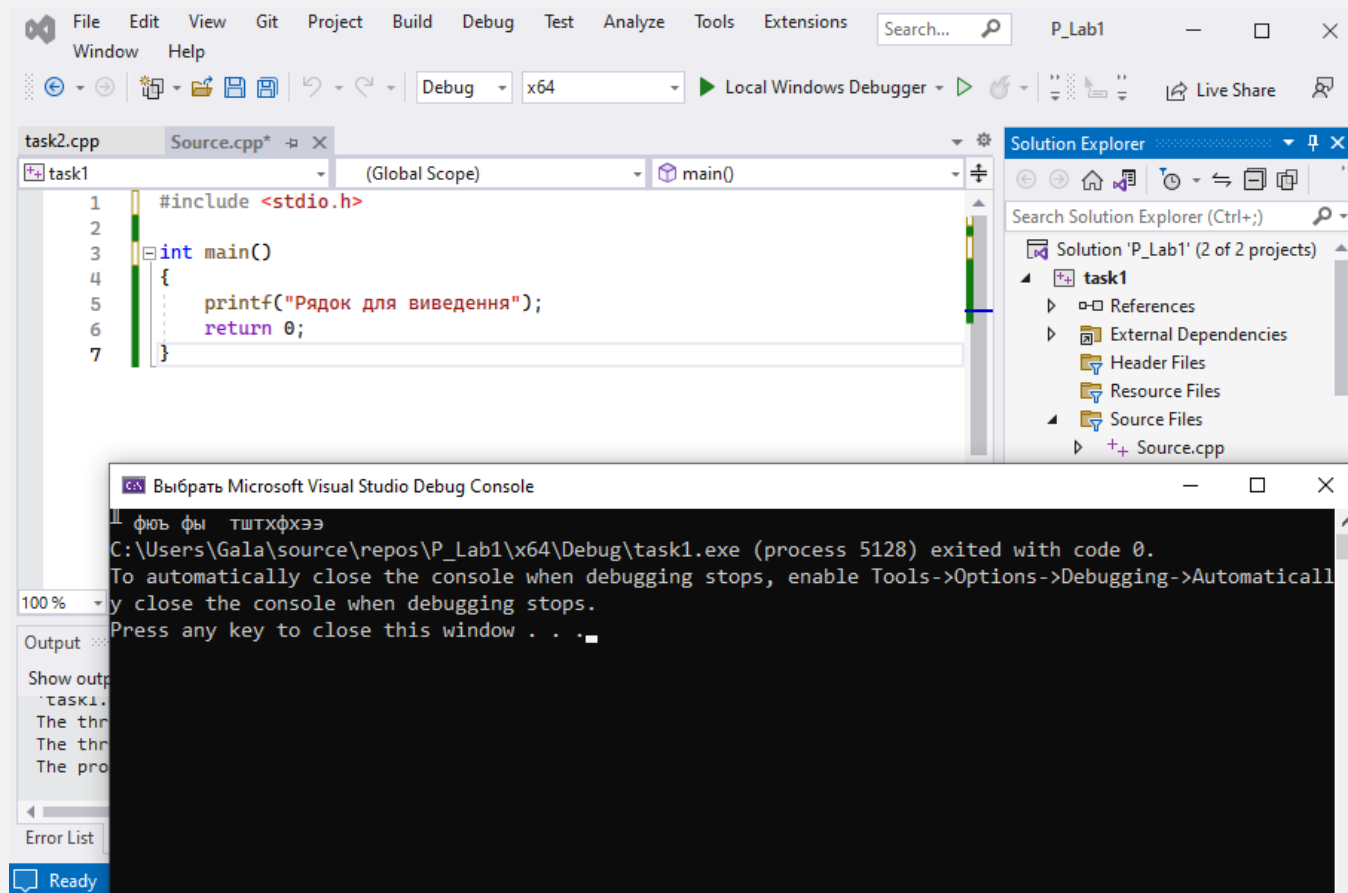
Для виконання введення та виведення інформації потрібна бібліотека **stdio.h**

```
#include <stdio.h>
```

Щоб вивести інформацію на екран використовується функція **printf()**:

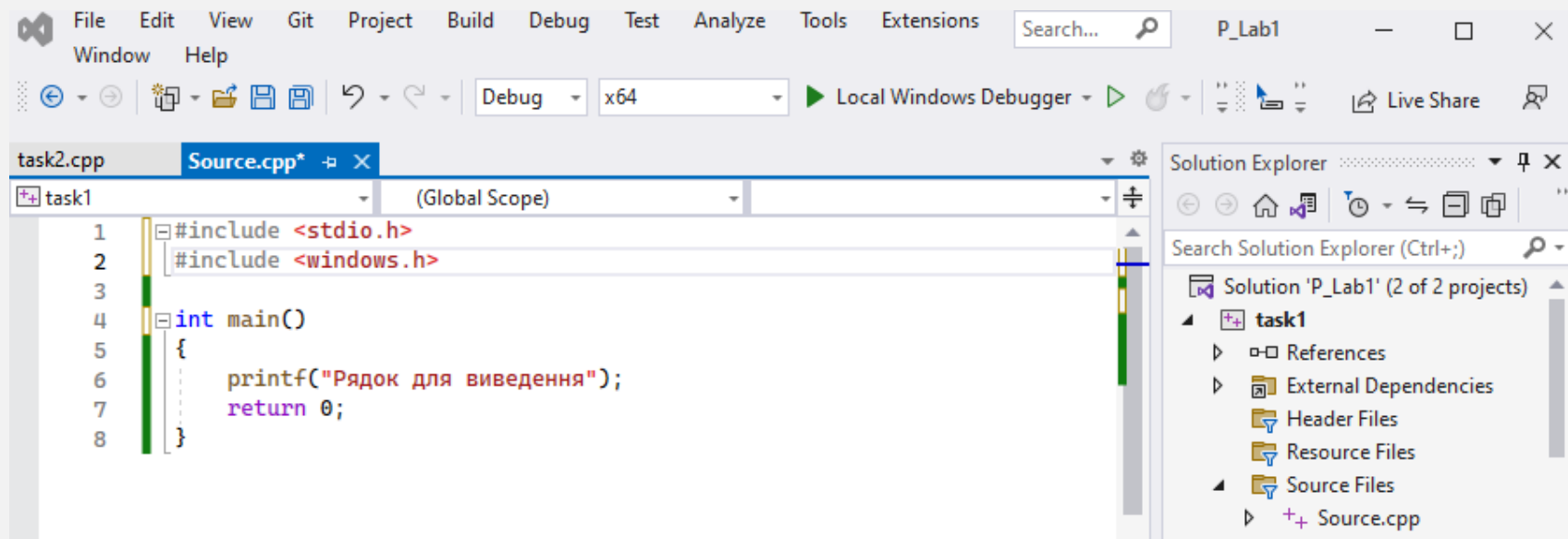
```
printf("Рядок для виведення");
```

Але виникне проблема при виведенні українського тексту



Щоб увімкнути правильне виведення кириличного тексту, потрібно:

1) додати підключення бібліотеки **windows.h**

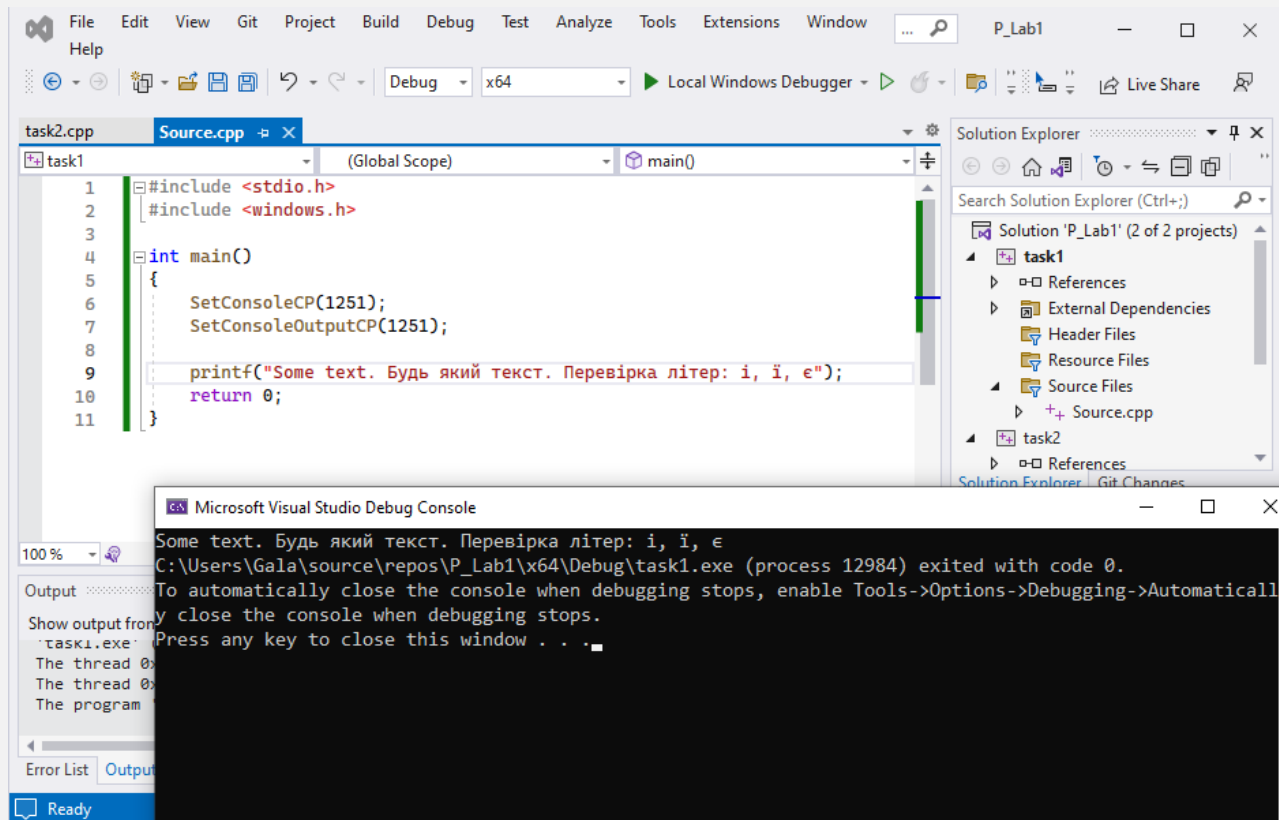


The screenshot shows the Visual Studio IDE with a C++ source file named 'Source.cpp' open. The code in the editor is as follows:

```
1 #include <stdio.h>
2 #include <windows.h>
3
4 int main()
5 {
6     printf("Рядок для виведення");
7     return 0;
8 }
```

The Solution Explorer on the right shows the project structure for 'P_Lab1', with 'task1' expanded to show 'Source.cpp'.

2) на початку функції main дописати рядки:
SetConsoleCP(1251);
SetConsoleOutputCP(1251);



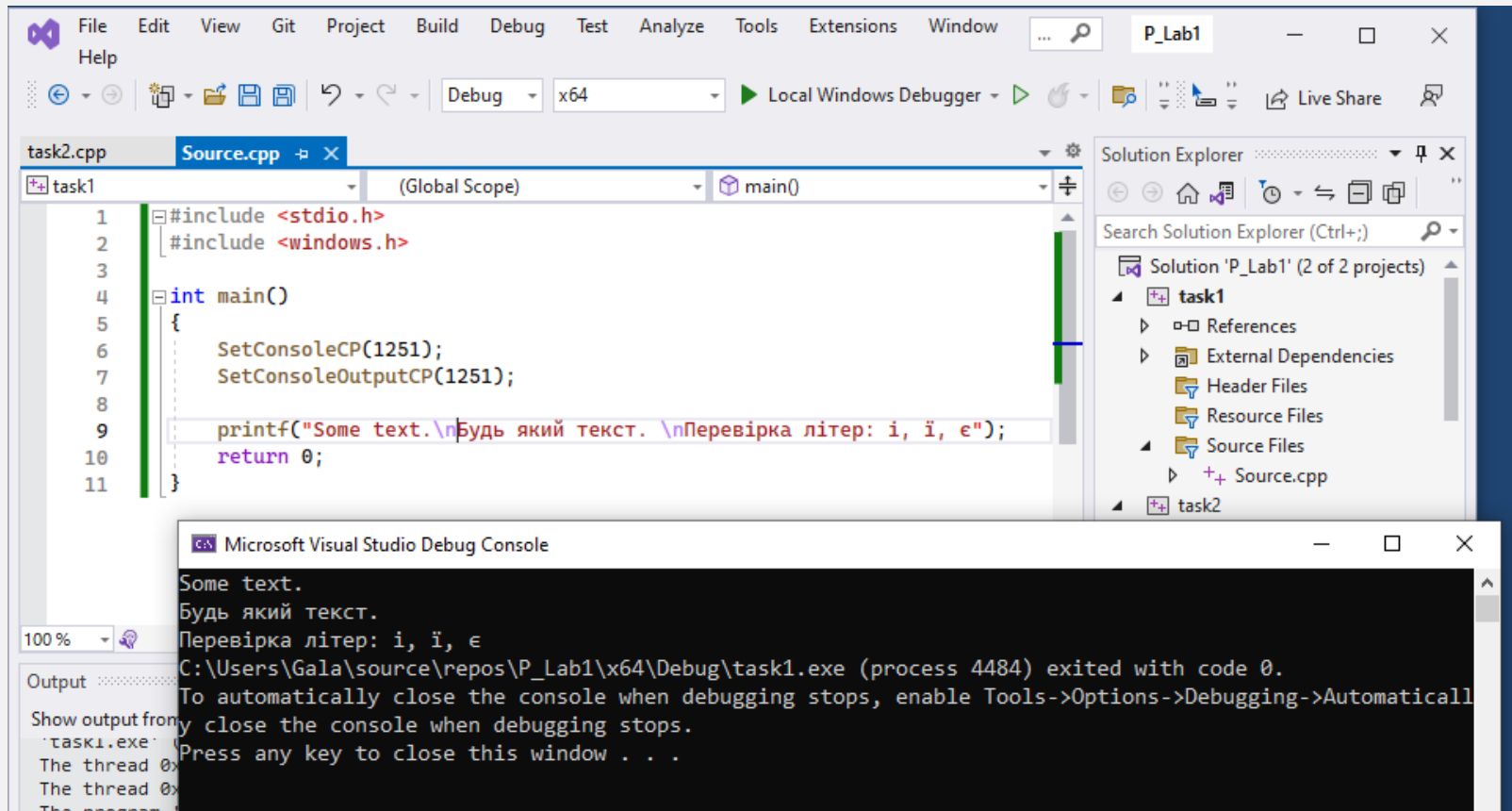
The screenshot displays the Visual Studio IDE with a C++ project named 'P_Lab1'. The main source file, 'task2.cpp', contains the following code:

```
1 #include <stdio.h>
2 #include <windows.h>
3
4 int main()
5 {
6     SetConsoleCP(1251);
7     SetConsoleOutputCP(1251);
8
9     printf("Some text. Будь який текст. Перевірка літер: i, і, є");
10    return 0;
11 }
```

The Solution Explorer on the right shows the project structure, including 'task1' and 'task2'. Below the code editor, the 'Microsoft Visual Studio Debug Console' window shows the output of the program:

```
Some text. Будь який текст. Перевірка літер: i, і, є
C:\Users\Gala\source\repos\P_Lab1\x64\Debug\task1.exe (process 12984) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

У рядок можна вставляти переходи на наступний рядок за допомогою `\n`:



The screenshot displays the Visual Studio IDE with a C++ source file named `Source.cpp`. The code defines a `main` function that sets console parameters and prints a multi-line message using `printf`. The message is: `Some text.\nБудь який текст. \nПеревірка літер: i, і, є`. The output window shows the execution result, confirming that the line breaks were processed correctly.

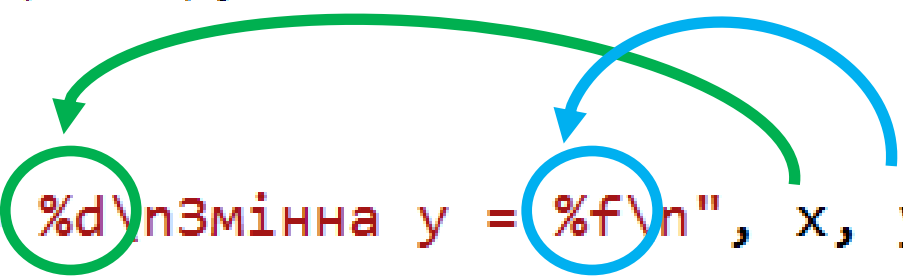
```
1 #include <stdio.h>
2 #include <windows.h>
3
4 int main()
5 {
6     SetConsoleCP(1251);
7     SetConsoleOutputCP(1251);
8
9     printf("Some text.\nБудь який текст. \nПеревірка літер: i, і, є");
10    return 0;
11 }
```

Microsoft Visual Studio Debug Console

```
Some text.
Будь який текст.
Перевірка літер: i, і, є
C:\Users\Gala\source\repos\P_Lab1\x64\Debug\task1.exe (process 4484) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Для того, щоб вивести на екран значення змінних у рядках в місцях, де потрібно підставити значення змінної розміщують **специфікатор форматування:**

```
int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    int x = 12;
    float y = -15.5;
    printf("Змінна x = %d\nЗмінна y = %f\n", x, y);
    return 0;
}
```



```
C:\Windows\system32\cmd.exe
Змінна x = 12
Змінна y = -15.500000
```


Специфікатор форматування – це рядок вигляду **%специфікатор**, де **специфікатор** вибирається в залежності від **типу даних**:

Типи	Специфікатор форматування для printf
char, unsigned char, short, unsigned short, int, unsigned int long, unsigned long	%d або %i
long long, unsigned long long	%lld
float, double, long double	%f або %g

Щоб зчитати інформацію з клавіатури використовується функція `scanf()`:

```
scanf ("%специфікатор", &змінна);
```

Специфікатор має відповідати типу даних змінної.

Для функції `scanf()` використовуються інші специфікатори форматування, ніж для функції `printf()`.

Типи	Специфікатор форматування для scanf()	Специфікатор форматування для printf()
char	%hhd	%d або %i
unsigned char	%hhu	%d або %i
short	%hd	%d або %i
unsigned short	%hu	%d або %i
int	%d	%d або %i
unsigned int	%u	%d або %i
long	%Ld	%d або %i
unsigned long	%Lu	%d або %i
long long	%lld	%lld
unsigned long long	%llu	%lld

Типи	Специфікатор форматування для scanf()	Специфікатор форматування для printf()
float	%f	%f або %g
double	%lf	%f або %g
long double	%Lf	%f або %g

6. Арифметичні та математичні операції

Над **цілими** змінними і значеннями можна виконувати математичні операції

Операція	Позначення
Додавання	+
Віднімання	-
Множення	*
Ділення	/
Залишок від ділення	%

Над **дробовими** змінними і значеннями можна виконувати математичні операції

Операція	Позначення
Додавання	+
Віднімання	-
Множення	*
Ділення	/

!!! Операцію **%** застосовувати для дробових значень не можна

Приклад. Написати програму для підрахунку значення виразу

$$y = \frac{12 + x}{5 + x} + \frac{x^2}{7 - z/3}$$

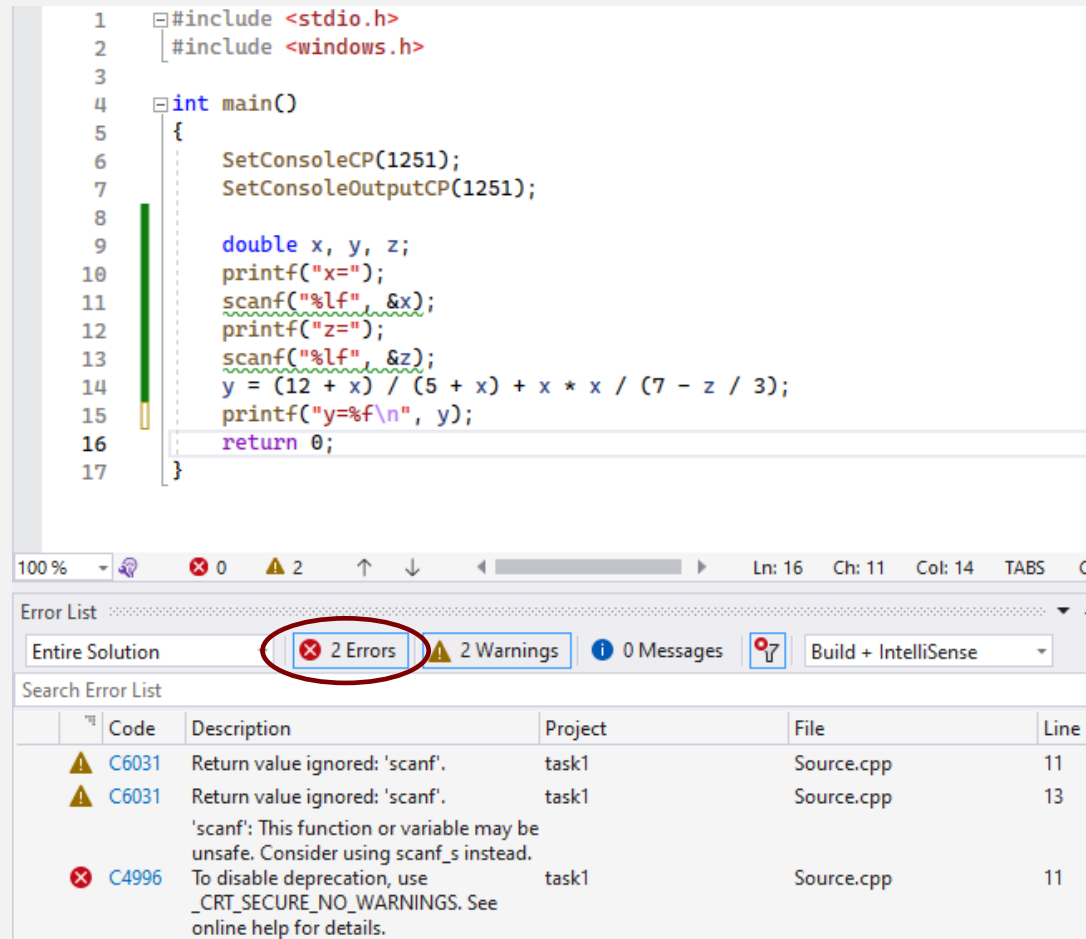
1) визначаємо вхідні дані. Це змінні **x та **z**, які потрібно зчитати з клавіатури**

2) визначаємо вихідні дані. Це змінна **y, в яку буде записуватись результат, і, значення яке буде виводитися на екран**

3) записуємо програмний код:

```
1  #include <stdio.h>
2  #include <windows.h>
3
4  int main()
5  {
6      SetConsoleCP(1251);
7      SetConsoleOutputCP(1251);
8
9      double x, y, z;
10     printf("x=");
11     scanf("%lf", &x);
12     printf("z=");
13     scanf("%lf", &z);
14     y = (12 + x) / (5 + x) + x * x / (7 - z / 3);
15     printf("y=%f\n", y);
16     return 0;
17 }
```


При спробі запуску програми на виконання отримуємо **помилку**:



```
1 #include <stdio.h>
2 #include <windows.h>
3
4 int main()
5 {
6     SetConsoleCP(1251);
7     SetConsoleOutputCP(1251);
8
9     double x, y, z;
10    printf("x=");
11    scanf("%lf", &x);
12    printf("z=");
13    scanf("%lf", &z);
14    y = (12 + x) / (5 + x) + x * x / (7 - z / 3);
15    printf("y=%f\n", y);
16    return 0;
17 }
```

100% 0 2 Ln: 16 Ch: 11 Col: 14 TABS C

Error List

Entire Solution 2 Errors 2 Warnings 0 Messages Build + IntelliSense

Search Error List

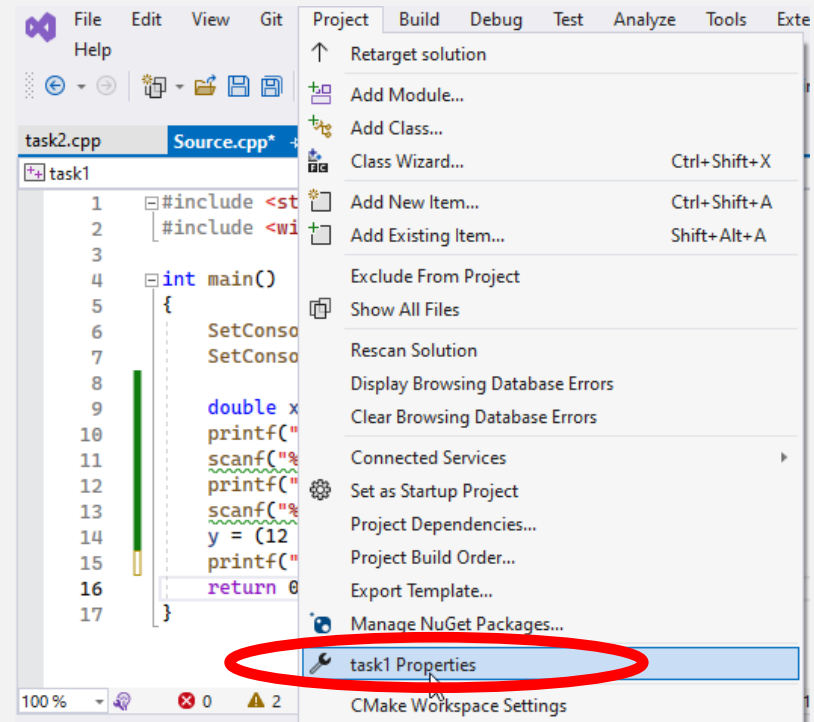
	Code	Description	Project	File	Line
⚠	C6031	Return value ignored: 'scanf'.	task1	Source.cpp	11
⚠	C6031	Return value ignored: 'scanf'.	task1	Source.cpp	13
⊗	C4996	'scanf': This function or variable may be unsafe. Consider using scanf_s instead. To disable deprecation, use _CRT_SECURE_NO_WARNINGS. See online help for details.	task1	Source.cpp	11

Для вирішення проблеми можна використовувати два способи:

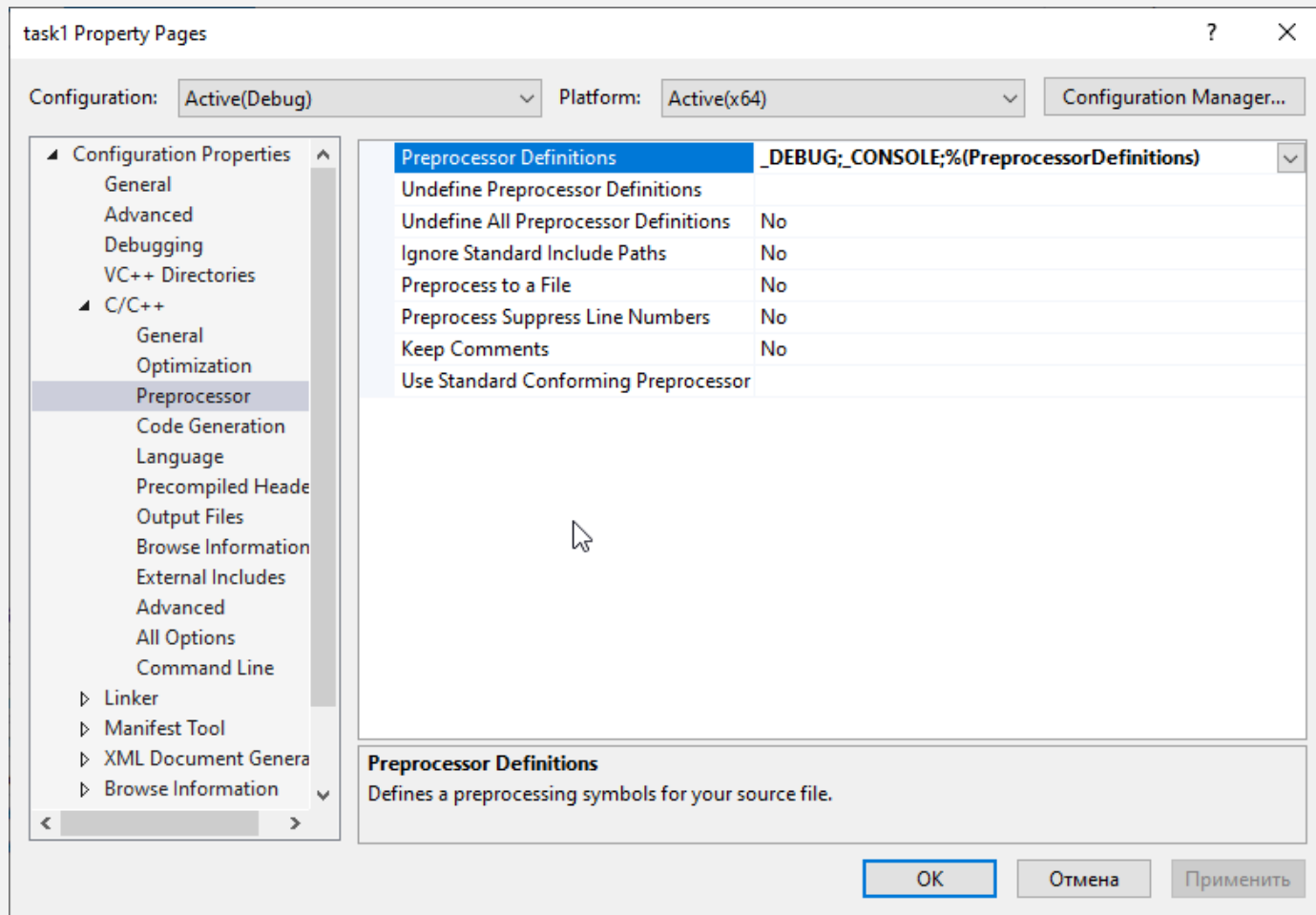
1) замість *scanf* використовувати *scanf_s*

2) змінити налаштування проекту:

а) У верхньому меню вибираємо "Проект",
далі - «Властивості...».

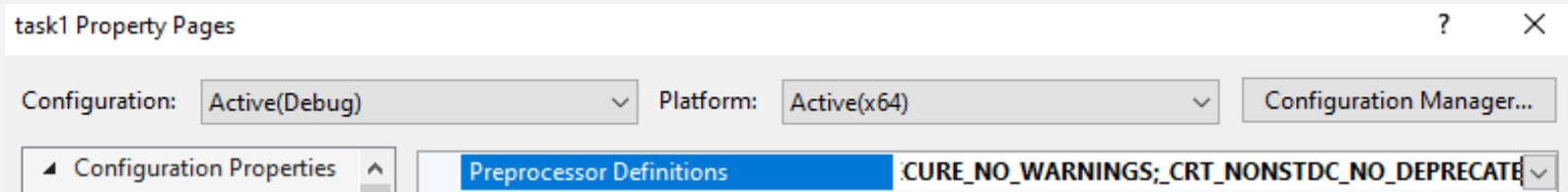


б) Далі у лівій панелі навігації вибрати: "Свойства конфигурации", "C/C++", "Препроцессор"



в) У рядок введення «Визначення препроцесору»
дописати в кінець:

`:_CRT_SECURE_NO_WARNINGS;_CRT_NONSTDC_NO_DEPRECATED`



```
task1 (Global Scope) main()
1 #include <stdio.h>
2 #include <windows.h>
3
4 int main()
5 {
6     SetConsoleCP(1251);
7     SetConsoleOutputCP(1251);
8
9     double x, y, z;
10    printf("x=");
11    scanf_s("%lf", &x);
12    printf("z=");
13    scanf_s("%lf", &z);
14    y = (12 + x) / (5 + x) + x * x / (7 - z / 5);
15    printf("y=%f\n", y);
16    return 0;
17 }
```

100% No issues found Ln: 17 Ch: 2 TABS CR

Error List

Entire Solution 0 Errors Warnings 0 Messages Build + IntelliSense

Code	Description	Project	File	Line
------	-------------	---------	------	------

*Помилки
немає*

Програма виконуватиметься без помилок

```
Microsoft Visual Studio Debug Console
x=12
z=12
y=49.411765

C:\Users\Gala\source\repos\P_Lab1\x64\Debug\task1.exe (process 6
To automatically close the console when debugging stops, enable
y close the console when debugging stops.
Press any key to close this window . . .
```

Мова **Ci** має математичну бібліотеку, яка містить такі функції:

Математична функція	Функція бібліотеки	Опис
$ x $	<code>fabs(x)</code>	модуль дробового числа
$ x $	<code>abs(x)</code>	модуль цілого числа
$\cos(x)$	<code>cos(x)</code>	косинус x
$\sin(x)$	<code>sin(x)</code>	синус x
$\text{tg}(x)$	<code>tan(x)</code>	тангенс x
e^x	<code>exp(x)</code>	експонента числа x
\sqrt{x}	<code>sqrt(x)</code>	квадратний корінь числа x

Мова **Ci** має математичну бібліотеку, яка містить такі функції:

Математична функція	Функція бібліотеки	Опис
x^y	<code>pow(x, y)</code>	Піднесення числа x до степеня y
<code>arccos(x)</code>	<code>acos(x)</code>	арккосинус x
<code>arctg(x)</code>	<code>atan(x)</code>	арктангенс x
<code>ln(x)</code>	<code>log(x)</code>	натуральний логарифм x
<code>lg10(x)</code>	<code>log10(x)</code>	десятковий логарифм x

Для виконання перерахованих функцій потрібно підключити бібліотеку **math.h**.

```
#include <math.h>
```

Тепер можна записувати вираз за допомогою функцій математичної бібліотеки:

$$s = \frac{x^{2y} + e^{y-1}}{1 + x|y - \operatorname{tg}z|} + \sqrt[3]{x} - \ln(z)$$

$$s = \frac{x^2y + e^{y-1}}{1 + x|y - \operatorname{tg}z|} + \sqrt[3]{x} - \ln(z)$$

```

1  #include <stdio.h>
2  #include <windows.h>
3  #include <math.h>
4
5  int main()
6  {
7      SetConsoleCP(1251);
8      SetConsoleOutputCP(1251);
9
10     double x, y, z, rez;
11     double a, b, c;
12     printf("x=");
13     scanf_s("%lf", &x);
14     printf("y=");
15     scanf_s("%lf", &y);
16     printf("z=");
17     scanf_s("%lf", &z);
18     a = pow(x, 2 * y) + exp(y - 1);
19     b = 1 + x * fabs(y - tan(z));
20     c = pow(x, 1 / 3.0) - log(z);
21     rez = a / b + c;
22     printf("rez=%f\n", rez);
23     return 0;
24 }

```

a b c

Ця та інша презентації доступні на сайті:

<http://learn.ztu.edu.ua/course/view.php?id=1650>