



Лекція 1.

Вступ до програмування.

Типи даних та операції мови Сі

План

- 1) Вступ до програмування**
- 2) Рішення та проекти у Visual Studio**
- 3) Найпростіша програма на мові Сі**
- 4) Типи даних**
- 5) Введення та виведення даних**
- 6) 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.

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

Серед переваг мови Сі потрібно відзначити основні:

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

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

OPLab1

- Task1
- Task2
- Task3
- Task4

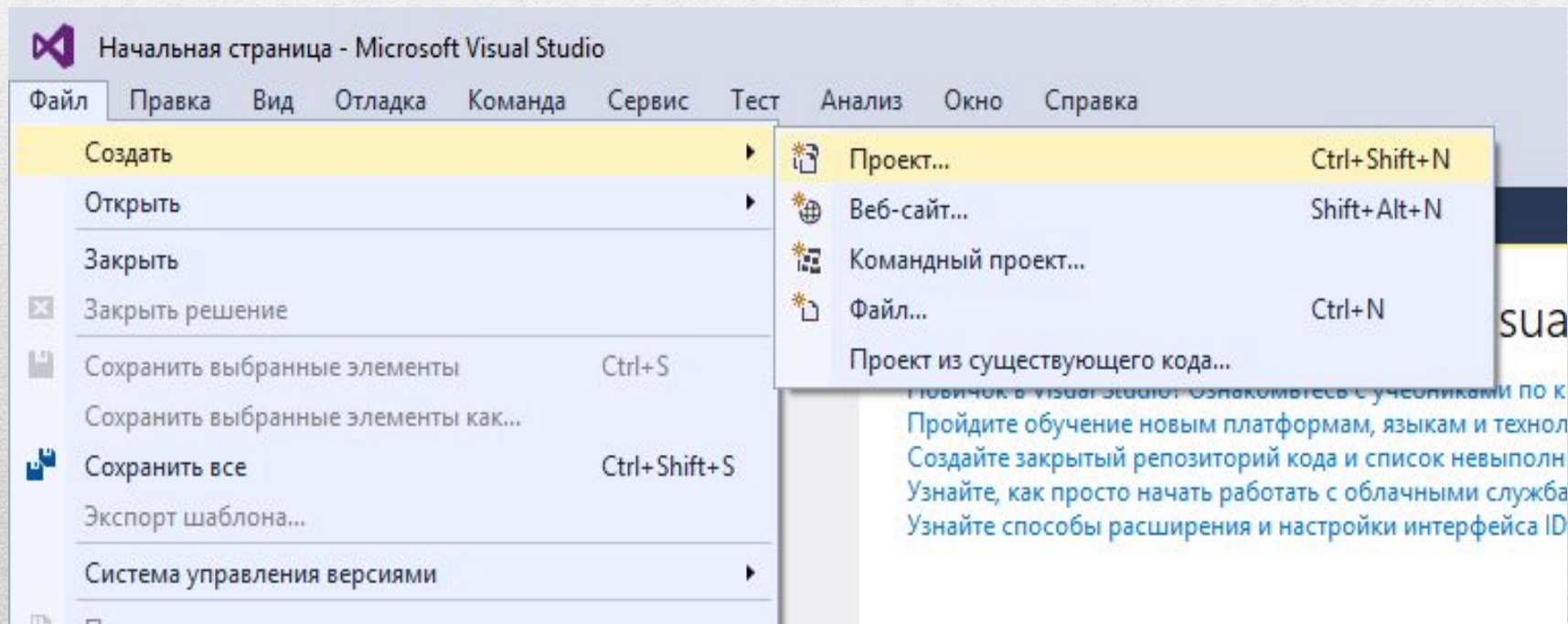


Рішення

*Проекти
(4 штуки)*

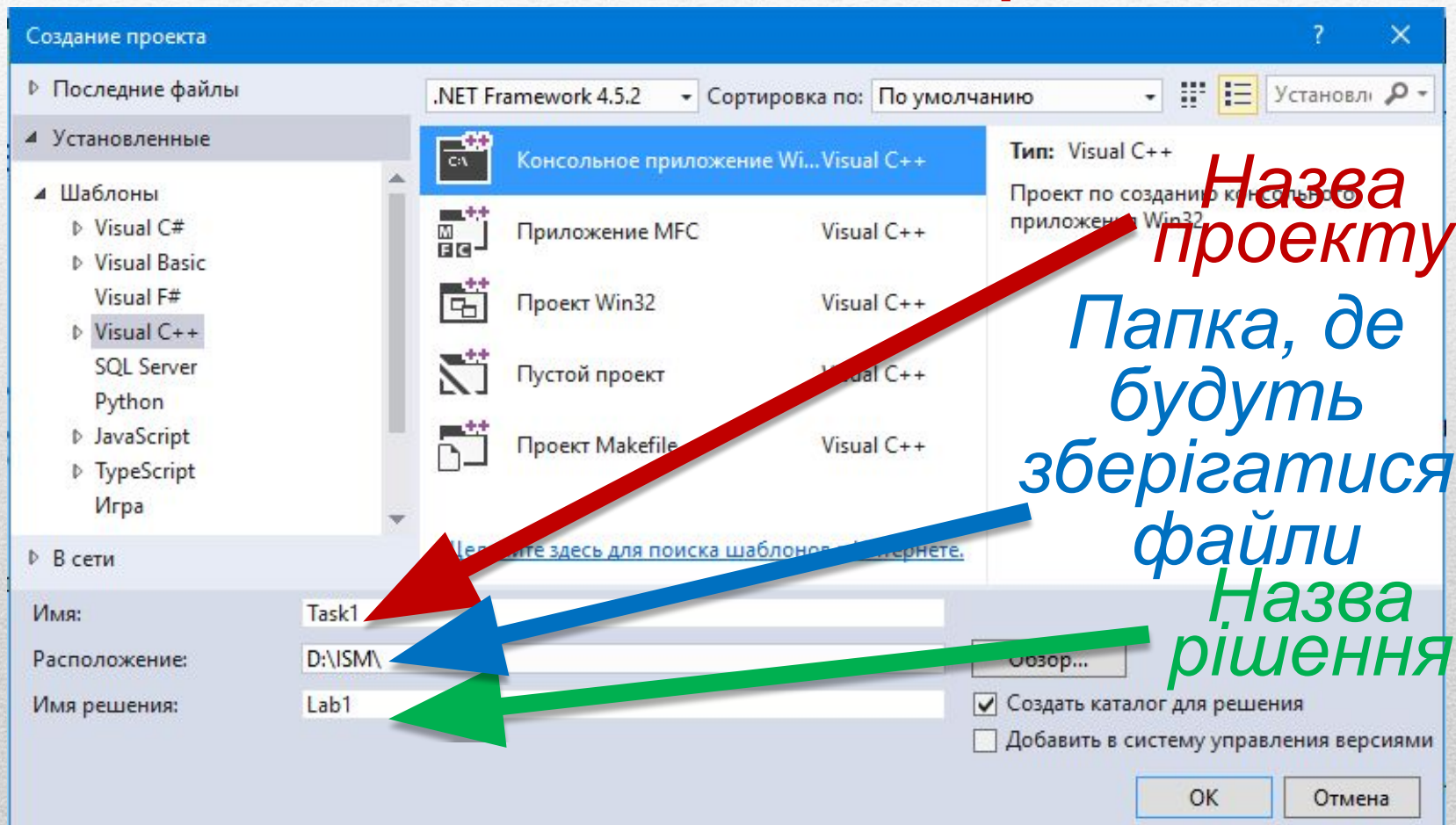
Для того, щоб створити рішення та проект потрібно:

1) У верхньому меню вибрати
Файл → Создать → Проект

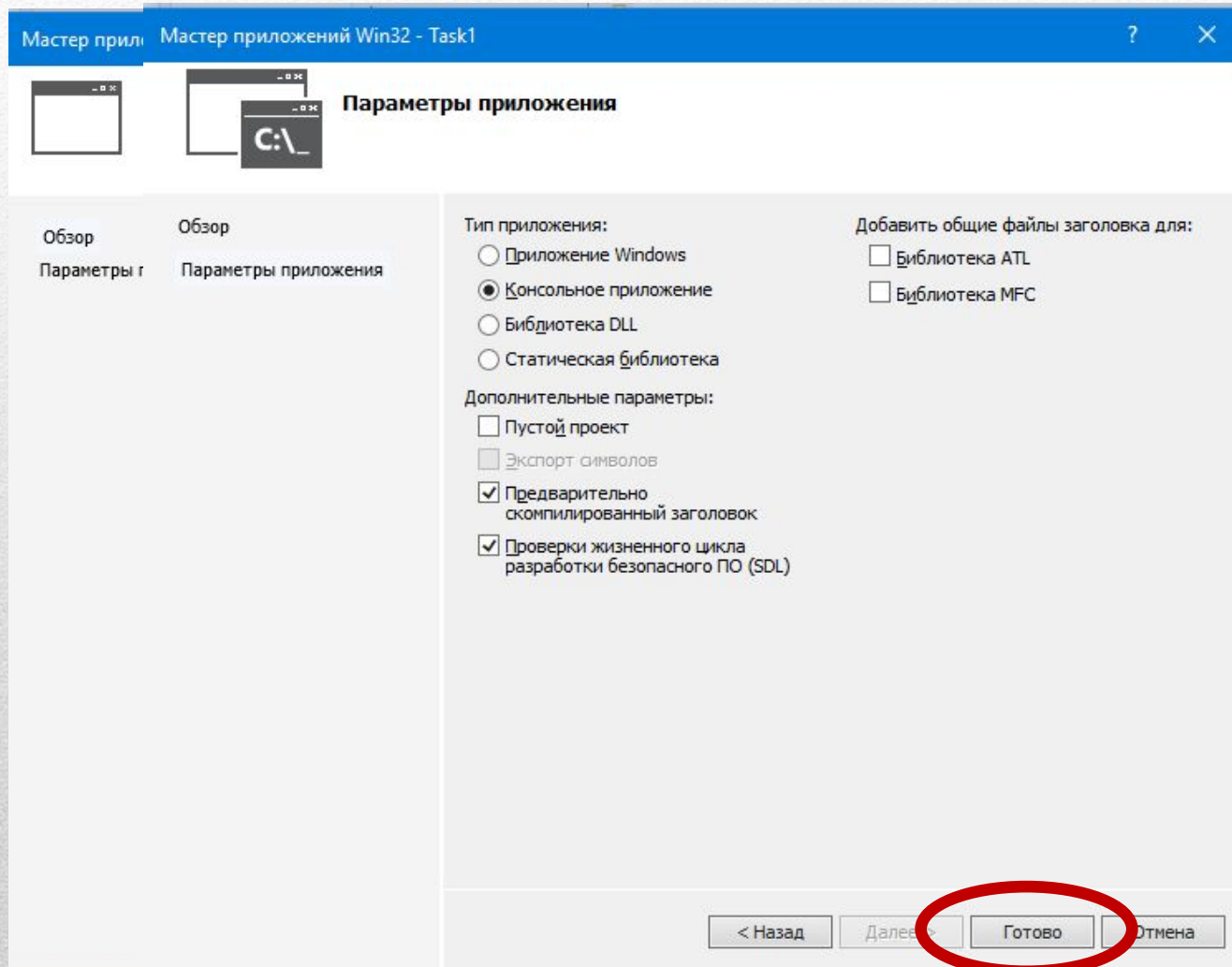


Вибрати «**Visual C++**», «**Консольное приложение**», ввести назву проекту та рішення

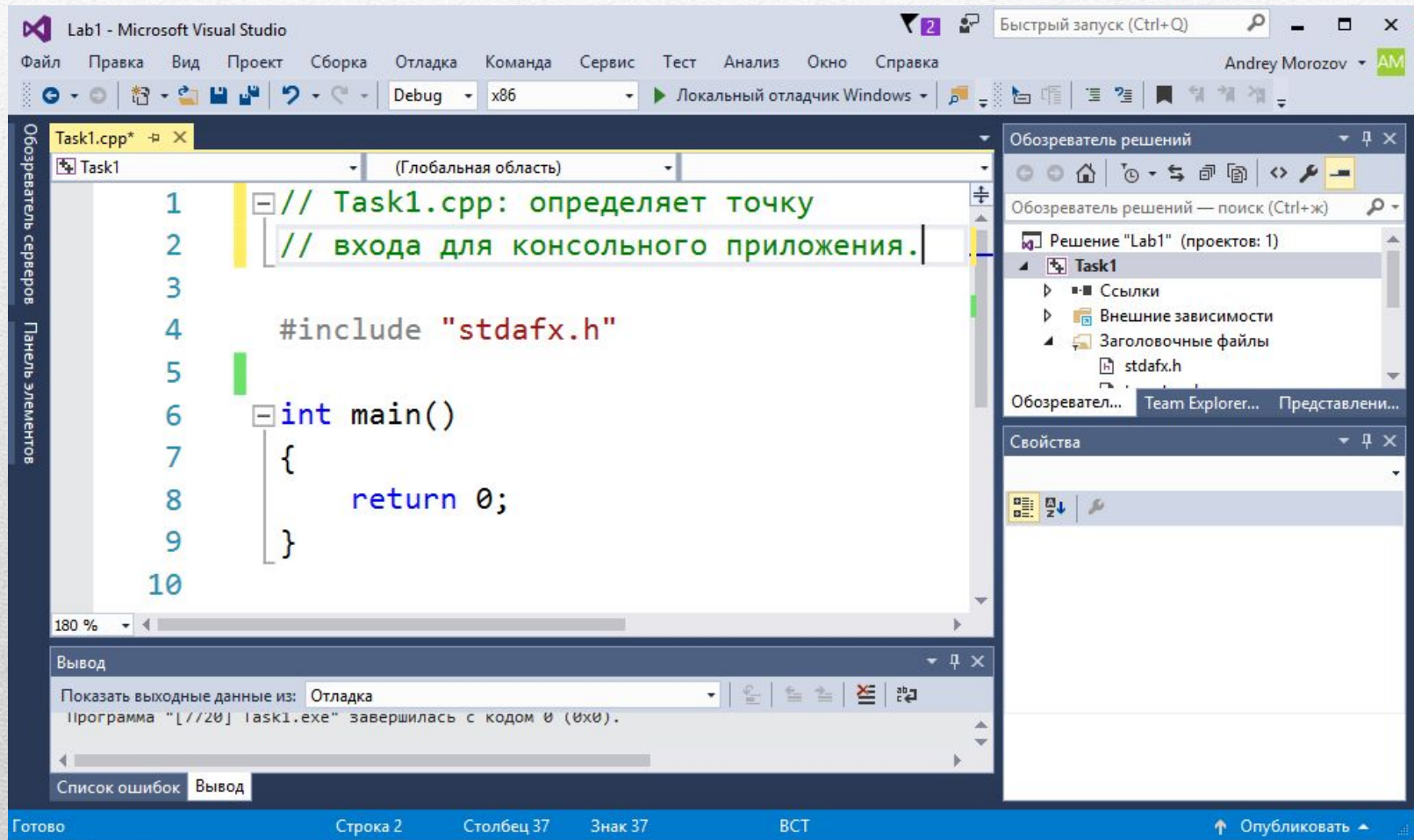
Файл → Создать → Проект



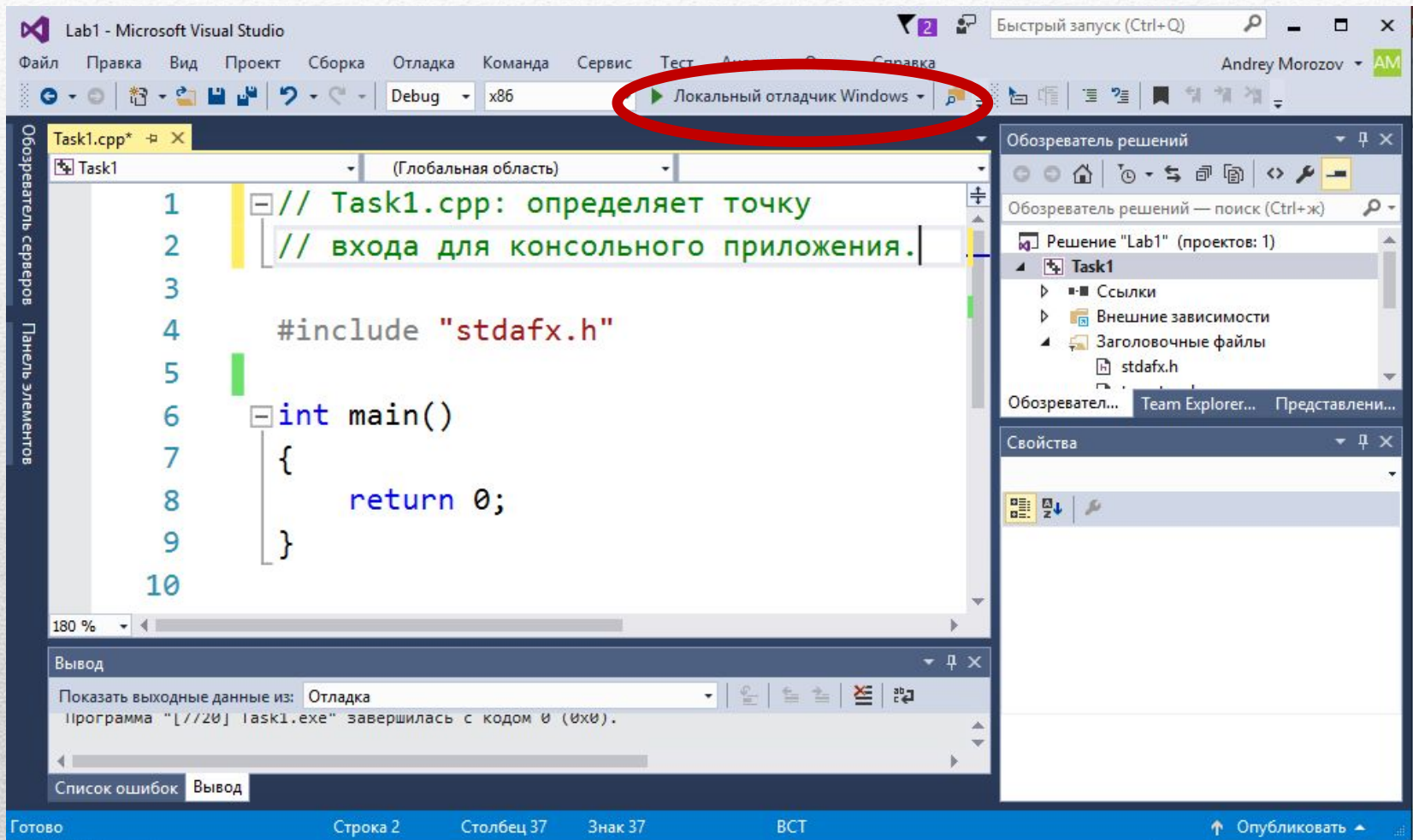
Виконати показані кроки та звірити параметри налаштування



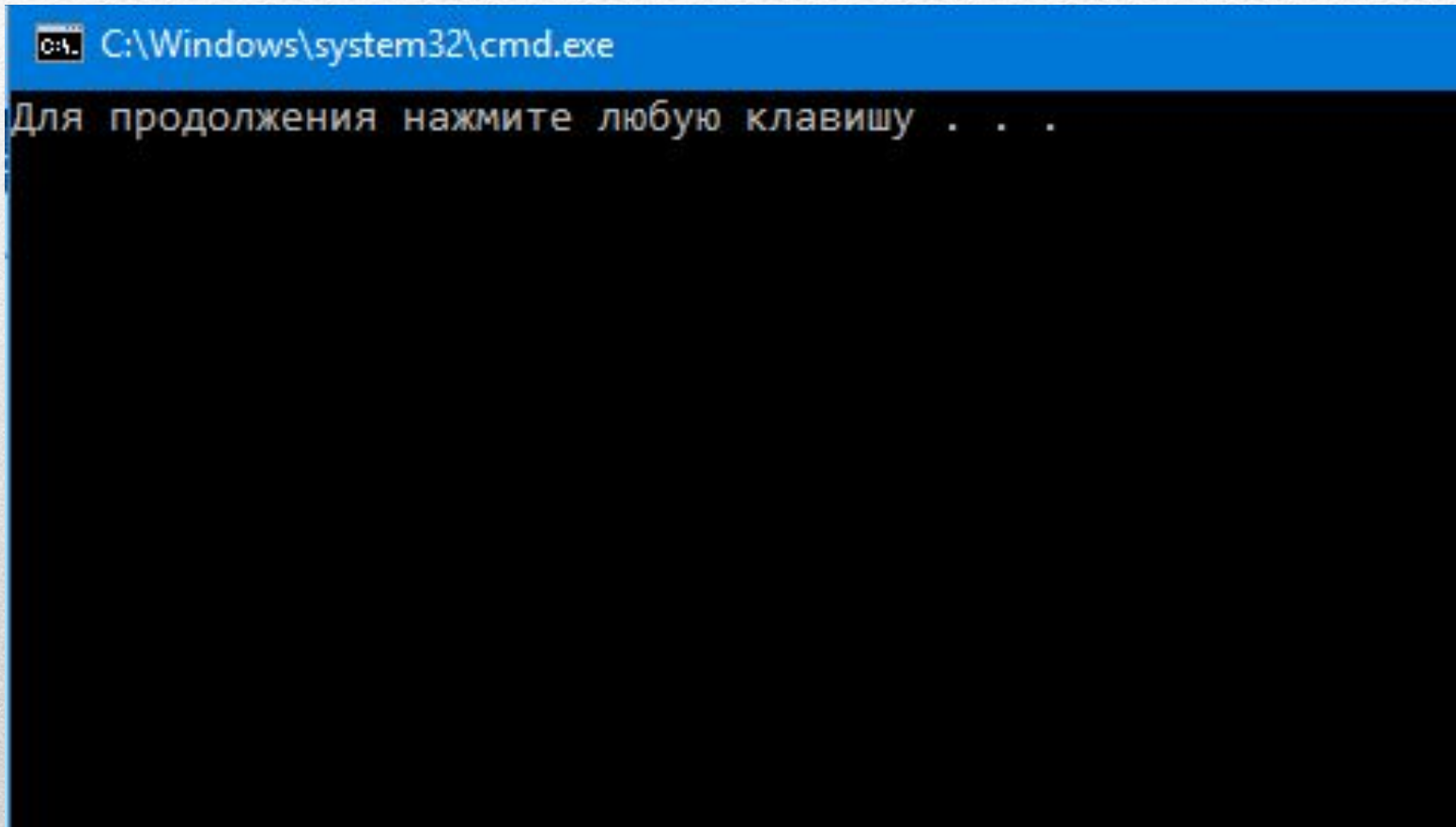
3. Найпростіша програма на мові Сі



Запуск програми виконується натисканням
клавіші **F5** або



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



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

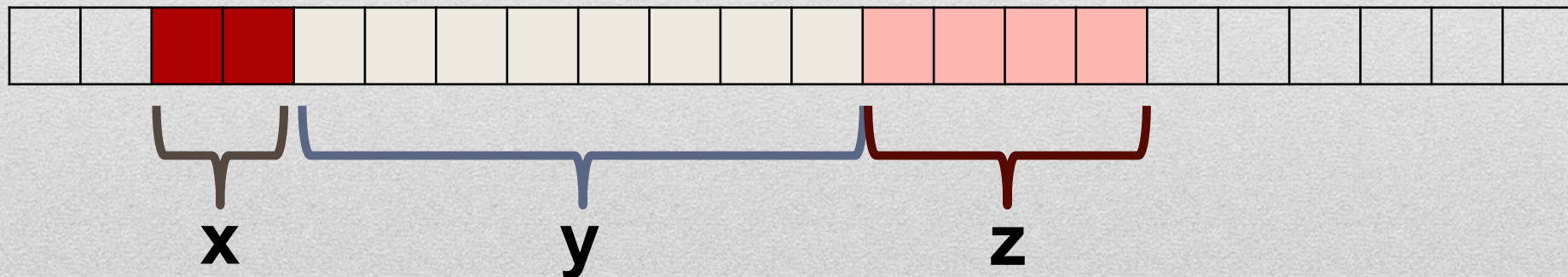
Змінна – це область пам'яті, яка має ім'я і в якій зберігається значення певного типу даних

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

| | | |
|-----------|--------------|---|
| тип даних | ім'я змінної | : |
|-----------|--------------|---|

Приклад:

short x;
double y;
int z;

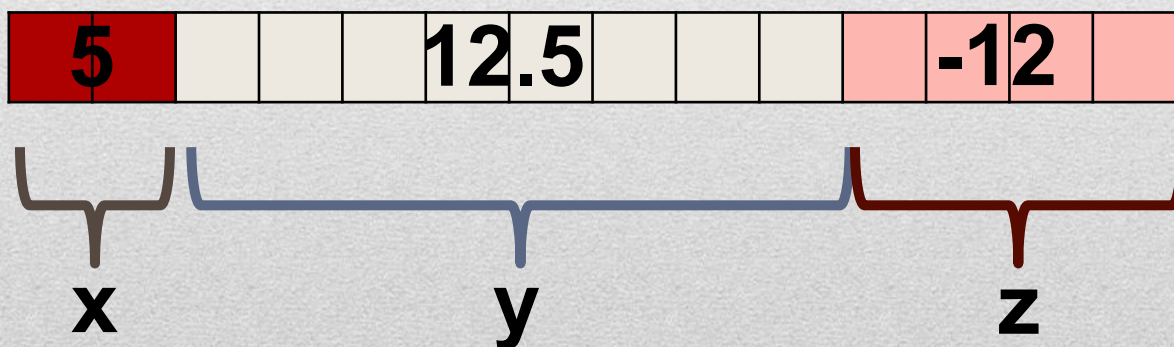


Для запису конкретного значення у змінну
потрібно використовувати операцію
присвоювання значення змінній:

Приклад:

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

ім'я змінної = значення ;



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:

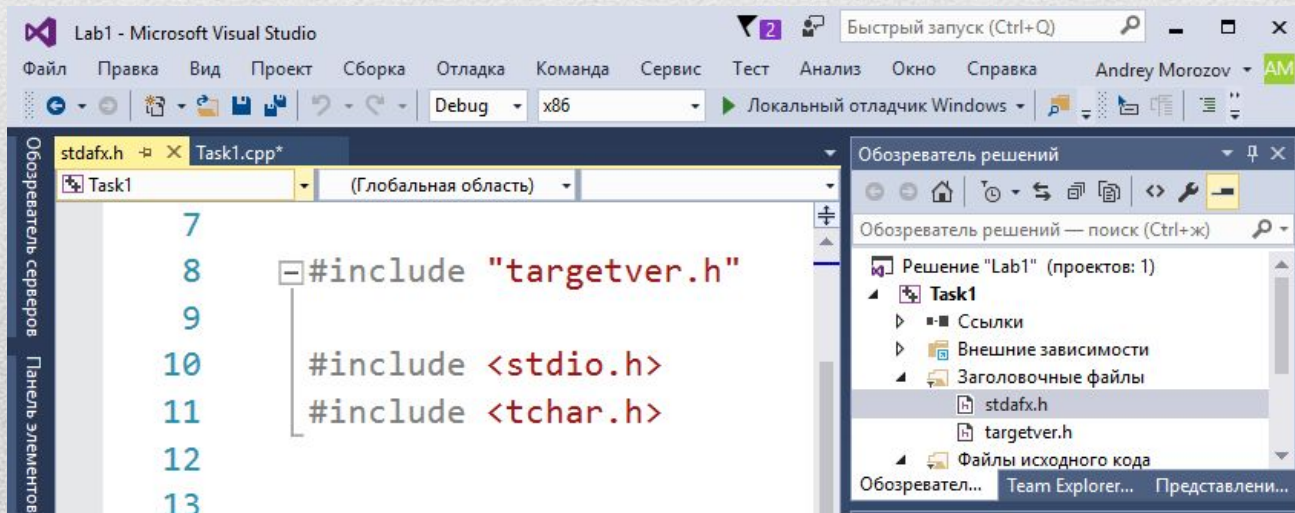
```
short x = 5, a, b;  
double y = 12.5;  
int z = -12;
```


5. Введення та виведення даних

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

Зазвичай, вона вже автоматично підключена у файлі **stdafx.h**:

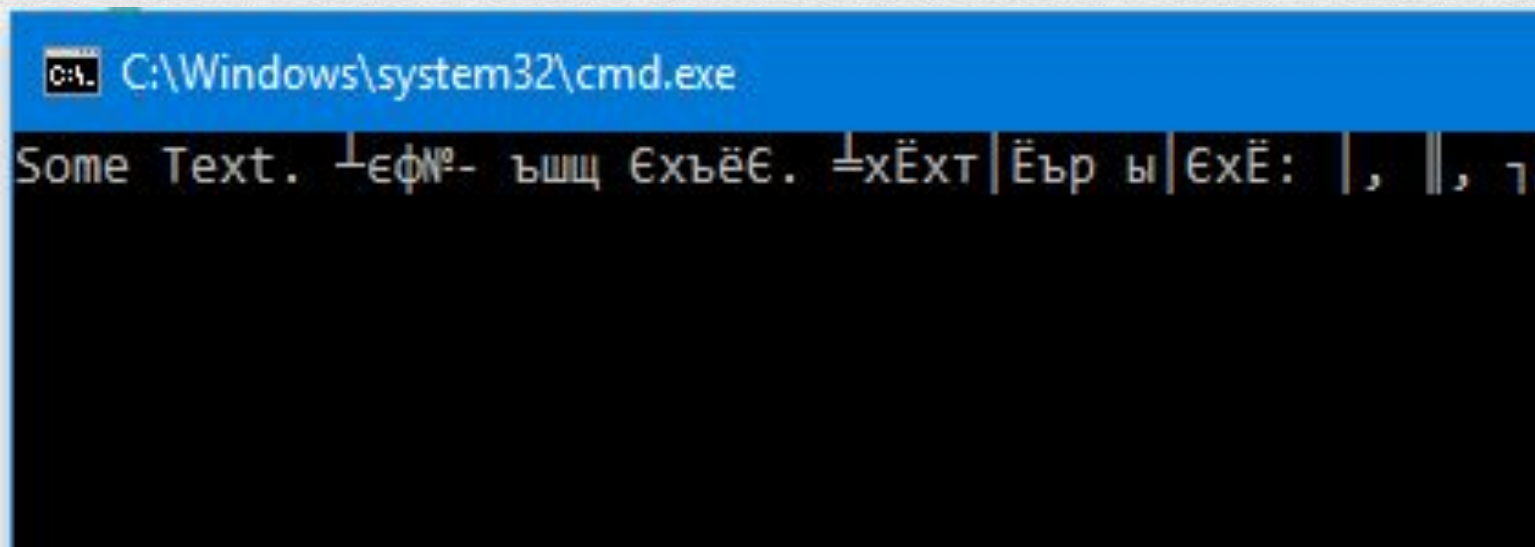
Якщо не підключена, тоді у файл **stdafx.h** потрібно додати рядок: **#include <stdio.h>**



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

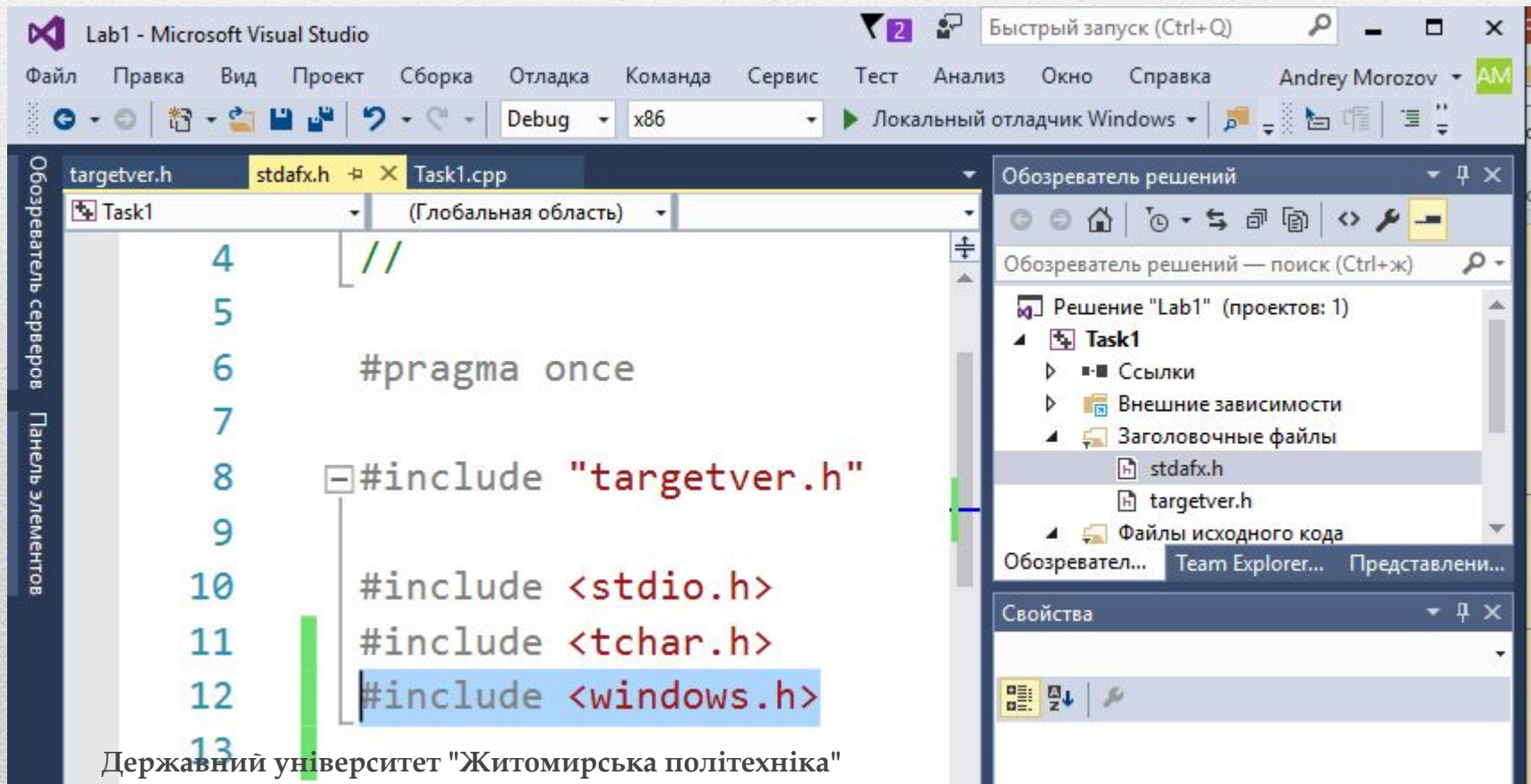
```
printf("Рядок для виведення");
```

Але виникне проблема при вивденні російського або українського тексту



Щоб увімкнути правильне виведення кириличного тексту, потрібно:

- 1) у файл `stdafx.h` додати підключення бібліотеки `windows.h`



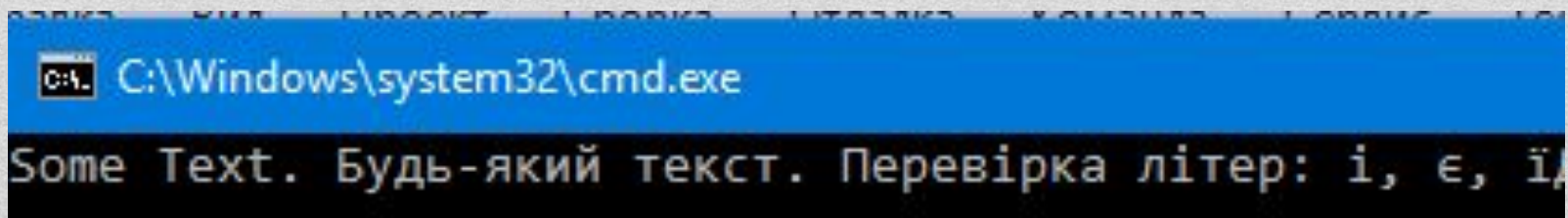
2) на початку функції main дописати рядки:

`SetConsoleCP(1251);`

`SetConsoleOutputCP(1251);`

```
#include "stdafx.h"

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    printf("Some Text. Будь-який текст. Перевірка літер: і, є, ї");
    return 0;
}
```



The screenshot shows a Windows command prompt window with the title bar "C:\Windows\system32\cmd.exe". The command prompt displays the output of a program: "Some Text. Будь-який текст. Перевірка літер: і, є, ї". The text is displayed in a monospaced font, and the window has a blue title bar and a black background.

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

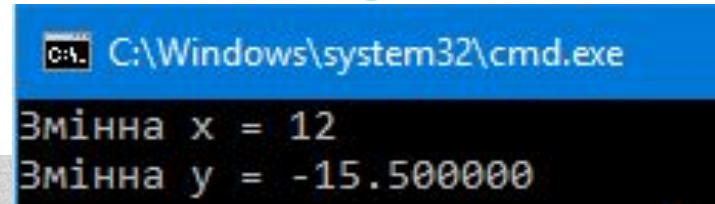
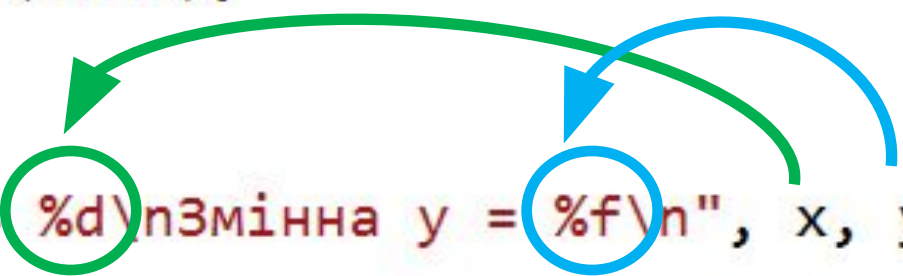
```
int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    printf("Рядок 1\nРядок 2\nРядок 3\n");
    return 0;
}
```

A screenshot of a Windows command prompt window. The title bar is blue and contains the text "C:\Windows\system32\cmd.exe". The command prompt area has a black background with white text. It displays the output of the C program: "Рядок 1", "Рядок 2", and "Рядок 3", each on a new line, demonstrating the effect of the '\n' escape sequence.

```
C:\Windows\system32\cmd.exe
Рядок 1
Рядок 2
Рядок 3
```

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 прочитати 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 прочитати з клавіатури**

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

3) записуємо програмний код:

```
int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    double x, y, z;
    printf("Введіть x = ");
    scanf("%lf", &x);
    printf("Введіть z = ");
    scanf("%lf", &z);
    y = (12 + x) / (5 + x) + x*x / (7 - z / 3);
    printf("Результат: y = %f\n", y);
    return 0;
}
```


При спробі запуску програми на виконання у Visual Studio 2015, отримуємо **помилку**:

```
10 double x, y, z;
11 printf("Введіть x = ");
12 scanf("%lf", &x);
13 printf("Введіть z = ");
14 scanf("%lf", &z);
15 y = (12 + x) / (5 + x) + x*x / (7 - z / 3);
16 printf("Результат: y = %f", y);
17 return 0;
```

164 %

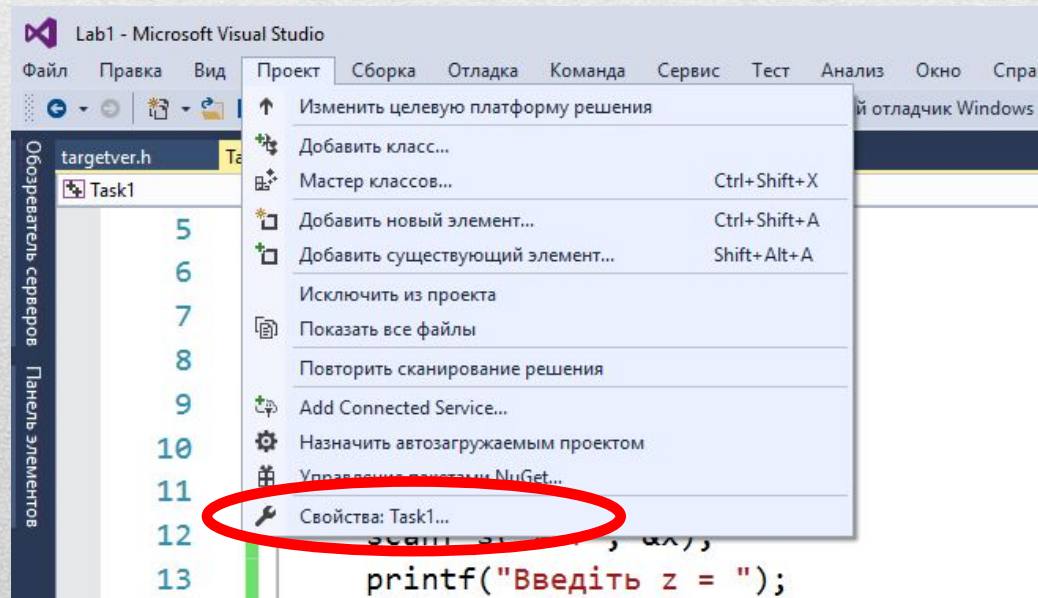
Список ошибок

Все решение 2 Ошибки 0 Предупреждения 0 Сообщения Сборка и IntelliSense

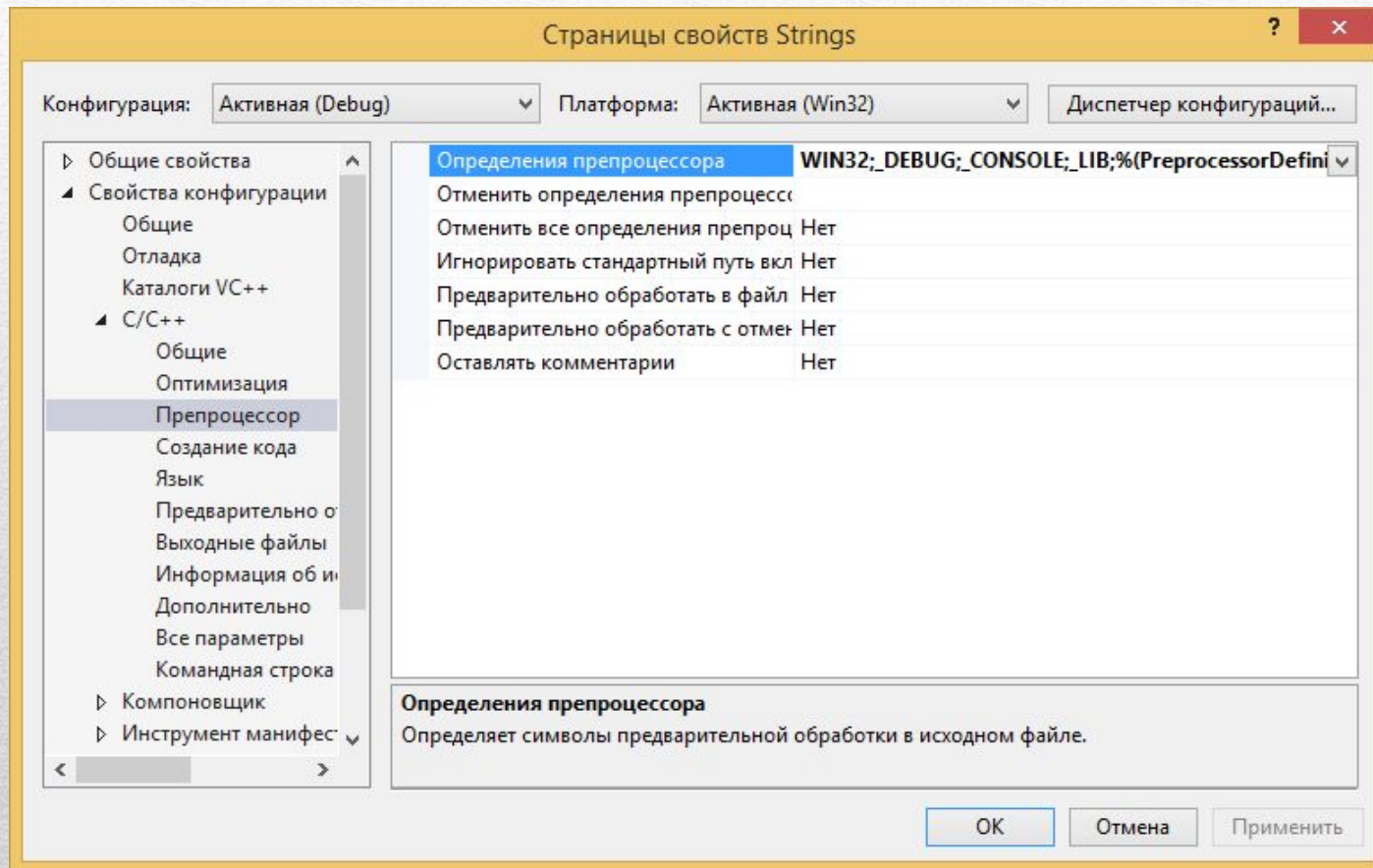
| | Код | Описание | Проект | Файл | Ст... | Состояние подае |
|---|-------|---|--------|-----------|-------|-----------------|
| ✖ | C4996 | 'scanf': This function or variable may be unsafe. Consider using scanf_s instead. To disable deprecation, use _CRT_SECURE_NO_WARNINGS. See online help for details. | Task1 | task1.cpp | 12 | |
| ✖ | 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 | task1.cpp | 14 | |

Для вирішення проблеми можна використовувати два способи:

- 1) замість **scanf** використовувати **scanf_s**
- 2) змінити налаштування проекту:
 - а) У верхньому меню вибираємо "Проект", далі - "Свойства ...".

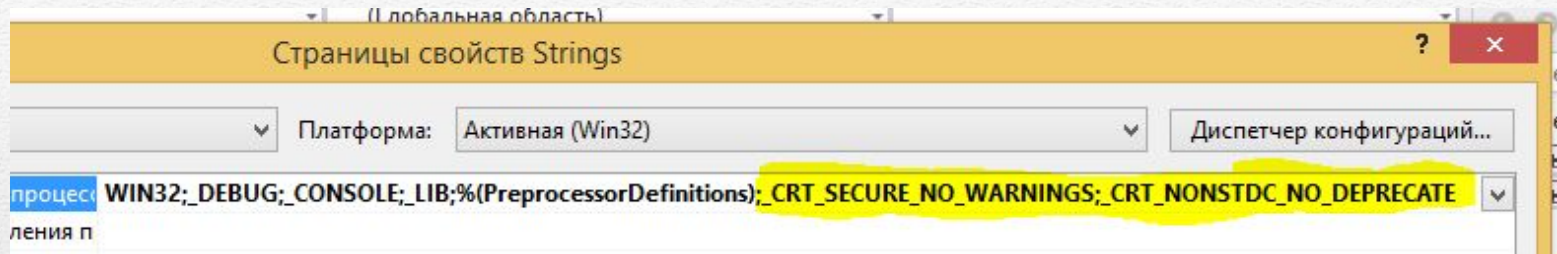


б) Далі у лівій панелі навігації вибрати: "Свойства конфигурации", "C/C++", "Препроцессор"

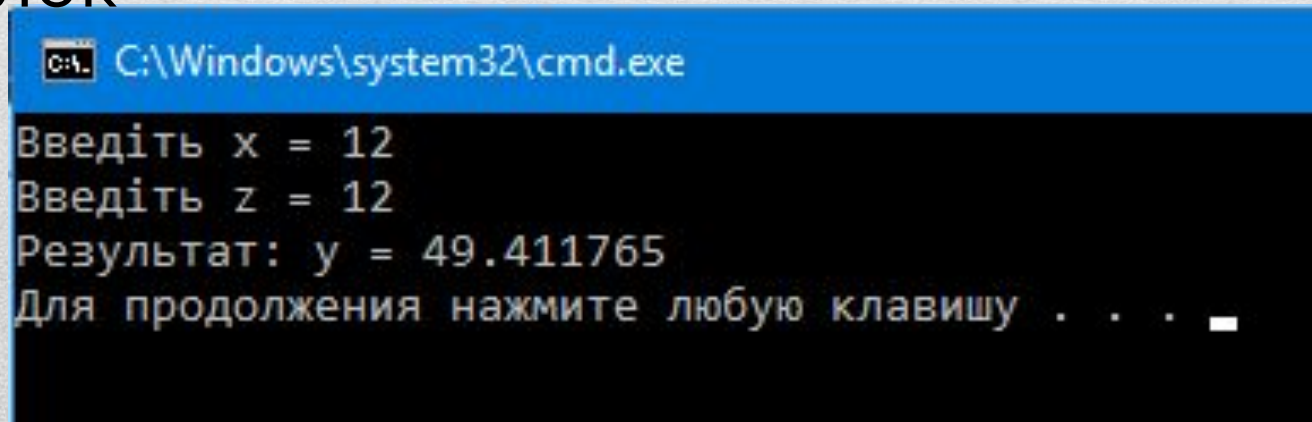


в) У рядок введення "Определения препроцессора" дописати у кінець:

`:_CRT_SECURE_NO_WARNINGS;_CRT_NONSTDC_NO_DEPRECATED`



Тепер програма виконуватиметься без помилок



```
C:\Windows\system32\cmd.exe
Введіть x = 12
Введіть z = 12
Результат: y = 49.411765
Для продолжения нажмите любую клавишу . . . _
```



```
6  int main()
7  {
8      SetConsoleCP(1251);
9      SetConsoleOutputCP(1251);
10     double x, y, z;
11     printf("Введіть x = ");
12     scanf("%lf", &x);
13     printf("Введіть z = ");
14     scanf("%lf", &z);
15     y = (12 + x) / (5 + x) + x*x / (7 - z / 3);
16     printf("Результат: y = %f\n", y);
17     return 0;
18 }
```

164 %

Список ошибок

Все решение 0 Ошибки 0 Предупреждения 0 Сообщения Сборка и IntelliSense

| Код | Описание | Проект | Файл | Ст... | Состояние подавлены |
|-----|----------|--------|------|-------|---------------------|
|-----|----------|--------|------|-------|---------------------|

*Помилки
немає*

Мова **Cі** 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 |
| <code>cos(x)</code> | <code>cos(x)</code> | косинус x |
| <code>sin(x)</code> | <code>sin(x)</code> | синус x |
| <code>tg(x)</code> | <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 |

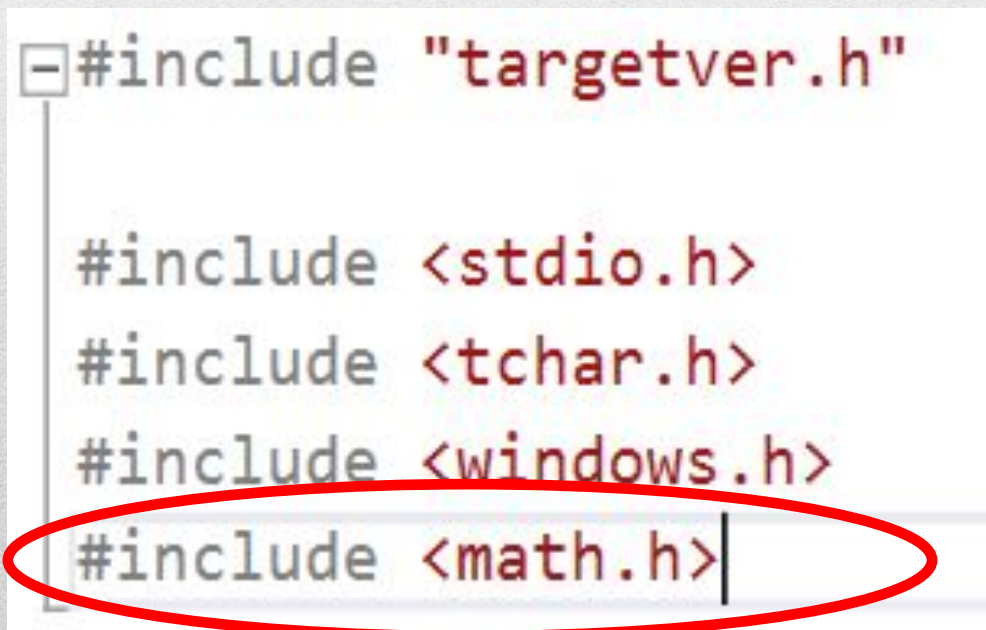
Мова **Ci** має математичну бібліотеку, яка містить такі функції:

| 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**.

Для цього у файлі **stdafx.h** треба дописати рядок:

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



```
#include "targetver.h"

#include <stdio.h>
#include <tchar.h>
#include <windows.h>
#include <math.h>
```


Тепер можна записувати вираз за допомогою функцій математичної бібліотеки:

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

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

```
int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    double x, y, z, a, b, c, s;
    printf("Введіть x:"); scanf("%lf", &x);
    printf("Введіть y:"); scanf("%lf", &y);
    printf("Введіть z:"); scanf("%lf", &z);
    a = pow(x, 2 * y) + exp(y - 1);
    b = 1 + x * fabs(y - tan(z));
    c = pow(x, 1.0 / 3) - log(z);
    s = a / b + c;
    printf("Result s = %lf", s);
    return 0;
}
```


Ця та інша презентації доступні на сайті:

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